On arithmetic and the discrete logarithm problem in class groups of curves

Der Fakultät für Mathematik und Informatik der Universität Leipzig eingereichte

Habilitationsschrift

zur Erlangung des akademischen Grades

doctor rerum naturalium habilitatus (Dr. rer. nat. habil.)

vorgelegt von

Dr. Claus Diem

geboren am 24.10.1972 in München

Leipzig, den 7.5.2008

Hiermit erkläre ich, die vorliegende Habilitationsschrift selbständig und ohne unzulässige fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angeführten Quellen und Hilfsmittel benutzt und sämtliche Teststellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialien oder erbrachten Dienstleistungen als solche gekennzeichnet.

On arithmetic and the discrete logarithm problem in class groups of curves

Claus Diem

Contents

Introduction

In this work computational problems related to divisors and divisor classes on curves are studied from a complexity theoretic point of view. Particular emphasis lies on the complexity of the discrete logarithm problem in degree 0 divisor class groups of curves over finite fields. Here we study the following general question: What interesting results on expected running times can one obtain if one imposes conditions on the *genera* and the *ground fields* but *no further conditions* on the curves and one bounds the expected running times in terms of either q^g, where q is the cardinality of the ground field and g the genus of the curve, or in terms of the cardinality of the degree 0 class group?

Note here that if E is an elliptic curve over a field K, the group of rational points $E(K)$ of E is canonically isomorphic to $\mathrm{Cl}^0(E)$, the degree 0 class group of E over K. We consider therefore in particular the discrete logarithm problem in the groups of rational points of elliptic curves over finite fields.

Additionally, the work contains a contribution on the foundations of algorithmic mathematics.

The main results

The main results are the following five new theorems on the discrete logarithm problem in degree 0 class groups of curves. The underlying complexity model is always a randomized random access machine model with logarithmic cost function. Of course, Theorem 4 also holds for a randomized Turing machine model.

Theorem 1 *Let some natural number $g \geq 2$ be fixed. Then the discrete logarithm problem in the degree 0 class groups of curves of genus g over finite fields can be solved in an expected time of*

$$\tilde{\mathcal{O}}(q^{2-\frac{2}{g}}) ,$$

where $\mathbb{F}_q$ is the ground field of the curve.

Theorem 2 *Let some natural number $g_0 \geq 2$ be fixed. Then the discrete logarithm problem in the degree 0 class groups of curves of genus $\geq g_0$ over finite fields can be solved in an expected time of*

$$\tilde{\mathcal{O}}((q^g)^{\frac{2}{g_0}(1-\frac{1}{g_0})}) ,$$

where $\mathbb{F}_q$ is the ground field and g the genus of the curve.

Theorem 3 *Let some natural number $g_0 \geq 2$ be fixed. Then the discrete logarithm problem in the degree 0 class groups of curves $\mathcal{C}/\mathbb{F}_q$ of genus $\geq g_0$ over finite fields can be solved in an expected time of*

$$\tilde{\mathcal{O}}((\#\,\mathrm{Cl}^0(\mathcal{C}))^{\frac{2}{g_0}(1-\frac{1}{g_0})}) .$$

In the following two theorems, q is always a prime power and n a natural number.

Theorem 4 *Let $\epsilon > 0$. Then the discrete logarithm problem in the groups of rational points of elliptic curves over finite fields $\mathbb{F}_{q^n}$ with $(2+\epsilon) \cdot n^2 \leq \log_2(q)$ can be solved in an expected time which is polynomially bounded in q.*

Theorem 5 *Let some natural number $n \geq 2$ be fixed. Then the discrete logarithm problem in the groups of rational points of elliptic curves over finite fields $\mathbb{F}_{q^n}$ can be solved in an expected time of*

$$\tilde{\mathcal{O}}(q^{2-\frac{2}{n}}) .$$

All these theorems can be found in Chapter 3 with the same numbering. More precisely, the locations of the theorems are respectively: Theorem 1: Section 3.3, Theorem 2 and Theorem 3: Section 3.4, Theorem 4 and Theorem 5: Section 3.5.

As a corollary to Theorem 4 one obtains easily:

Let again $\epsilon > 0$, and let $a > 2 + \epsilon$. Then the discrete logarithm problem in the groups of rational points of elliptic curves over finite fields $\mathbb{F}_{q^n}$ with $(2+\epsilon) \cdot n^2 \leq \log_2(q) \leq a \cdot n^2$, where q is a prime power and n a natural number, can be solved in an expected time of

$$e^{\mathcal{O}(1)\cdot(\log(q^n))^{2/3}} .$$

Indeed, we obtain an expected running time which is polynomially bounded in

$$q = 2^{\log_2(q)} = 2^{(\log_2(q))^{(1+1/2)\cdot 2/3}} \leq 2^{(\sqrt{a}\cdot n \log_2(q))^{2/3}} .$$

This result establishes for the first time that there exists a sequence of finite fields of increasing size such that the elliptic curve discrete logarithm over these fields can be solved in an an expected time which is *subexponential* in the input length.

As a special case of Theorem 3 we obtain:

One can solve the discrete logarithm problem in the degree 0 class groups of curves $\mathcal{C}/\mathbb{F}_q$ of genus at least 3 in an expected time of

$$\tilde{\mathcal{O}}((\#\operatorname{Cl}^0(\mathcal{C}))^{\frac{4}{9}}) .$$

In contrast, for any sequence of curves over finite fields such that the group order is divisible by a prime of size $\Theta(\#\operatorname{Cl}^0(\mathcal{C}))$, "generic methods" have an expected running time of $\Omega(\#\operatorname{Cl}^0(\mathcal{C})^{\frac{1}{2}})$.

Theorem 2 is a variant of Theorem 3, and clearly both Theorem 2 and Theorem 3 imply Theorem 1. Theorem 5 can be seen as an analog of Theorem 1 by substituting the genus g by the extension degree n. From a theoretical point of view, the analogy goes deeper: In both cases, the group under consideration is in a natural way isomorphic to the group of rational points in an abelian variety over $\mathbb{F}_q$. In the former case it is the Jacobian variety of the curve, and in the latter case it is the Weil restriction of the elliptic curve with respect to the extension $\mathbb{F}_{q^n}|\mathbb{F}_q$. The dimensions of these abelian varieties are g and n respectively. Even though Theorem 1 and Theorem 5 are analogous, we are not able to prove an analog of Theorems 2 or 3 for elliptic curves over extension fields.

Related results

Concering the general question on the discrete logarithm problem in degree 0 class groups of curves over finite fields posed at the beginning, besides the new results mentioned above, we are aware of the following results:

With the baby-step-giant-step algorithm and the results on arithmetic in class groups presented in Section 2.6, the discrete logarithm problem can be solved in a time of $\tilde{\mathcal{O}}(\#\operatorname{Cl}^0(\mathcal{C})^{\frac{1}{2}})$. By the bound $\#\operatorname{Cl}^0(\mathcal{C}) \leq (\sqrt{q}+1)^{2g}$ one can also obtain results in terms of q^g instead of $\#\operatorname{Cl}^0(\mathcal{C})$.

Besides these results, there is only one additional (proven) result we are aware of: It is the result on "high genus curves" by F. Heß which can be found in [Heß05]. Let us in order to formulate the result introduce the standard complexity function

$$L_N[\alpha, c] := e^{c\cdot\log(N)^{\alpha}\cdot(\log\log(N))^{1-\alpha}}$$

for parameters $\alpha \in (0,1)$ and $c > 0$. Then the result is:

Let us consider a class of curves over finite fields such that $\log(q) \in o(g \log(g))$*, where as usual* q *is the cardinality of the ground field and* g *is the genus. Then the discrete logarithm problem in the degree 0 class groups of such curves can be solved in an expected time of*

$$L_{q^g}[\frac{1}{2}, 32^{\frac{1}{2}} + \epsilon]$$

for any $\epsilon > 0$*.*

Similarly to the previous result, one can also obtain corresponding results in terms of $\#\,\mathrm{Cl}^0(\mathcal{C})$ via the bound $(\sqrt{q}-1)^{2g} \leq \#\,\mathrm{Cl}^0(\mathcal{C})$.

By reading the work [Heß05] in conjunction with the present work, the reader obtains an up-to-date account on the complexity of the discrete logarithm problem for classes of curves defined by imposing conditions on the genera and the ground fields.

We note that by considering certain "transfers" to "easier" groups one can obtain further classes of curves for which interesting statements on the complexity of the discrete logarithm problem hold. We mention here the supersingular elliptic curves; the discrete logarithm problem for these curves can be solved in an expected time of $L_q[\frac{1}{2}, 12^{\frac{1}{2}} + o(1)]$ via a transfer to the multiplicative group of an extension of degree at most 6 of the ground field.

In this work we do not consider any kind of "transfer". Neither do we apply advanced changes of the representation of the curves under consideration, contrary to the suggestion put forward in [Die06]. We note that one might argue that for Theorems 4 and 5 we pass from the $\mathbb{F}_{q^n}$-rational points of an elliptic curve E over $\mathbb{F}_{q^n}$ to the $\mathbb{F}_q$-rational points of the Weil restriction of E with respect to $\mathbb{F}_{q^n}|\mathbb{F}_q$. The important aspect is here nonetheless that *no computation* is performed in doing so. Indeed, the algorithm can be formulated without even mentioning the Weil restriction, and we do so.

Computational models and representations

In order to make a computation problem – like, for example, the discrete logarithm problem in degree 0 class groups of curves over finite fields – well defined, one has to state how the mathematical objects under consideration are represented for the computations. In Chapter 2 we discuss various possibilities to represent the objects under consideration: (non-singular, proper) curves, as well as divisors and divisor classes on curves. Our starting point is here the representation of a curve via a *plane model*, whereby we mean a possibly singular plane curve which is birational to the curve under consideration. We discuss various ways to represent divisors, with an emphasis on the ideal theoretic method inspired by the analogous task for number

fields. We do not consider methods which use expansions at singular points because these methods lead – in particular in small positive characteristic – to various complications which can easily be avoided with the ideal theoretic approach.

We discuss basic manipulations of divisors and divisor classes, and in particular the computation of the so-called "Riemann-Roch space" $L(D)$ of a divisor D. Also, we discuss how one can efficiently change between the various representations we describe. The results are based on a particular "field oriented" model of computation which we define in Chapter 1.

The main purpose of this chapter is to show that there are natural ways to represent curves, divisors and divisor classes such that basic computations can be performed in an number (or expected number) of field and bit operations which is polynomially bounded in certain input parameters. The results also translate to corresponding results over finite fields and in a bit-oriented model. The most important result for the discrete logarithm algorithms is that with a particular representation one can perform the computation in degree 0 class groups of curves over finite fields in the so-called *free representation* (where divisors are represented by formal sums of closed points) in an expected time which is polynomially bounded in the genus of the curve and the logarithm of the cardinality of the ground field (see Proposition 2.117).

Chapter 1 is more abstract: We discuss various problems which might be described as lying in the foundations of computational mathematics.

In complexity theoretic statements one often represents mathematical objects by other objects, often bit-strings. We are of the opinion that it is convenient to talk about representations not only in terms of representations by bit-strings but also by other kind of mathematical objects. Let us illustrate this with an example: Say we want to represent divisors on curves. We would like to express that we represent divisors by formal sums of closed points, but we do not want at that particular point in time consider the task to represent the closed points. A natural statement would then be the following sentence: "We represent divisors on curves by closed points." But now the question arises if one can give a precise mathematical meaning to this sentence. With other words: Can one define mathematical objects which capture the intuition of this sentence? We propose to capture a sentence like the one under consideration with an essentially surjective partial functor between two large groupoids.

In computational mathematics (especially in computational number theory), one often reads about "bit-operations", but very often at the same time no computational model is given. We propose a particular model which we call the *bit-RAM model* to capture the intuitive meaning of bit complexity.

Further, we introduce what we call the "generic field RAM" model. This is an abstract computational model which is designed to capture the idea of an algorithm which can be applied to instances of a computational problem over various fields. It captures the idea of "uniformity" in a stronger sense than any "algebraic" model in the literature we are aware of.

Related works

We mention now how this work relates to previous work by both other authors and the author, and we give some more information about the methods used to establish the results.

As already mentioned, all theorems mentioned above are new.

The algorithms for all the theorems rely on the so-called index calculus method which roughly consists of finding relations between prime divisors in a set of "medium size" and input elements and solving the discrete logarithm problem via linear algebra.

The theorems are as follows related to previous work by other authors:

The idea to apply the index calculus method to curves of a fixed genus goes back to P. Gaudry ([Gau00]). After a first improvement was suggested by R. Harley, it was realized by N. Thériault ([Thé03]) that by applying a large prime variation one can decrease the expected running time by a factor which is superpolynomial in the input length. The idea to use a double large prime variation goes back to K. Nagao and independently to P. Gaudry and E. Thomé. In both cases, the authors argued that on a heuristic basis it should be possible to obtain an expected running time of $\tilde{O}(q^{2-\frac{2}{g}})$ for hyperelliptic curves of fixed genus g in imaginary quadratic representation. Later a complete proof for hyperelliptic curves in imaginary quadratic representation and with cyclic degree 0 class group was given by Gaudry, Thomé and Thériault. A substantially easier proof for the same class of curves was then given by the author, and the work has meanwhile been published (see [GTTD07]). To obtain Theorem 1, we modify the construction of the graph of large prime relations, and we give an efficient procedure to compute a small system of group elements which with high probability is a generating system.

The proofs of Theorems 2 and 3 are then not so difficult. They rely on a quite weak statement on the probability that a uniformly distributed divisor splits into prime divisors whose degree satisfies a particular upper bound. The statement we need can be obtained from a certain much more refined statement in the work by F. Heß on the computation of discrete logarithms in degree 0 class groups of "high genus" which was already mentioned above ([Heß05]).

The algorithms for Theorem 4 and Theorem 5 rely on what we call a

"decomposition procedure" which in turn is based on solving systems of multivariate equations. The systems in turn are obtained with appropriate homogenizations of the so-called *summation polynomials* introduced by I. Semaev in [Sem98].

Shortly after Semaev made his preprint available, it was realized by Gaudry that one can use these summation polynomials to obtain interesting complexities for fixed extension degree. In the meantime Gaudry's work has been accepted for publication, and the present publicly available version of his work ([Gau04b]) also contains a "Theorem" in which the statement of Theorem 5 is claimed. We would however like to point out that Gaudry's work in various substantial ways falls short of establishing this result. More details on the shortcomings of Gaudry's work can be found in subsection 3.5.1.

Already at a presentation at the Elliptic Curve Cryptography Workshop (ECC) in Bochum in 2004, the author has argued that on a heuristic basis a statement similar to the corollary to Theorem 4 should hold. But it is only this work which contains a proof that there really does exist a sequence of increasing finite fields over which the elliptic curve discrete logarithm problem is solvable in subexponential time.

We consider most results in Chapter 2 to be "more or less known" to the experts. However, the fragmented literature on the questions studied in Chapter 2 made it hard for the author to judge if really all basic computations related of divisors which are necessary for the index calculus algorithms can be performed in a satisfying complexity for all curves. A related problem is that one should be careful in interpreting statements in the literature that certain computations can be performed in a particular time. Sometimes there are hidden restrictive assumptions, and in any case one should be careful to check if the representation used can efficiently be transformed into a representation one needs for other purposes.

As already remarked, we focus on the ideal theoretic approach to divisors. A key algorithm for this approach, namely an ideal-based algorithm for computation of Riemann-Roch spaces, has been published in [Heß01]. Further information on ideal arithmetic for number fields can be found in the book [Coh96], and often statements from [Coh96] can easily be transferred to function fields.

We would however like to stress that even though the basic techniques in this chapter where surely known before, some key statements in this chapter have not appeared in the literature before. Here we mention as examples the efficient transformation between the different representations we discuss and the factorization free computation of the maximal order.

It is our hope that this chapter will serve as a useful reference.

Chapter 1 consists mostly of definitions some of which are completely new while others are variants of definitions in the literature.

On the nature of this work

As already indicated in the beginning, we view this work as lying within the realm of complexity theory. Because of the objects under consideration and the methods employed, the work also falls within arithmetic algebraic geometry.

The discrete logarithm problems in the groups of rational points of elliptic curves over finite fields and more generally in degree 0 class groups of curves over finite fields have been suggested as primitives for public key cryptographic systems. The reader might therefore ask himself: What is the relationship between this work and cryptology? Is there any cryptologic application of the results obtained in this work?

Our answer is that this depends on the meaning of the word "cryptology" one has in mind. Surely the area of cryptology should include the design and analysis of cryptographic systems – whereby we mean practical cryptographic systems. Some researchers also consider the part of complexity theory dealing with theoretical problems which are motived by questions related to secrecy as being part of cryptology.

To us it seems however to be more appealing to reserve the work "cryptology" for the study of questions which are indeed related to potential practical applications. With this meaning of the word in mind, we clearly want to state:

This work does not fall into the realm of cryptology.

Moreover, the work is related to cryptologic questions only tangentially, if at all. It would clearly be a misuse of this work to evaluate the security of (practical) cryptographic systems on the basis of the asymptotic complexity theoretic statements in this work or the absence thereof.

We would like to take the opportunity and reflect a bit on the nature of works in complexity theory. Given the language of complexity theory – in which sentences like "Given X one can compute Y in time T" are made – one might have the feeling that complexity theory is part of applied mathematics. We would however like to point out that this would be a misconception. Indeed, in complexity theory a statement like the one above does not at all mean that something can really be computed in the sense that a physical machine outputs a result. Rather, the statement is an abbreviation for a longer statement in which the existence of certain mathematical objects with certain well-defined properties is asserted (for example the existence of a Turing machine with certain properties).

In short, we view this work as lying within complexity theory and for a large part also within arithmetic algebraic geometry, which both lie within pure mathematics.

We mention that this work contains no computer experiments. This is deliberate. Indeed, computer experiments would not bring us any closer to the goal we have set ourself: To study the discrete logarithm problem in class groups of curves from a complexity theoretic point of view.

Acknowledgments

I thank J. Adler, A. Bluher, A. Enge, G. Frey, S. Galbraith, P. Gaudry, F. Heß, M.-D. Huang, K. Khuri-Makdisi, V. Miller, A. Odlyzko, J. Scholten, E. Thomé, N. Thériault, E. Volcheck, and M. Wendt for discussions related to this work.

Chapter 1

Computational problems, computational models, and complexity

1.1 Introduction

In order to make a computational problem, like for example the discrete logarithm problem in degree 0 class groups of curves, well defined, one has to answer the following questions.

- *On what computational model is the problem based?*
- *What complexity measure / cost function is used?*
- *How are the mathematical objects under consideration represented?*

It suggests itself to ask for a computational model and a complexity measure which captures the intuitive notion of *bit-complexity* and at the same time allows for efficient indirect addressing. Surely, if indirect addressing is desired, RAM models are an appropriate answer. A problem is however that RAM models are always based on a particular set of commands, and the selection of these commands is associated with a certain arbitrariness. In order to overcome this arbitrariness we define a particular Random Access Machine (RAM) model which we call *bit-oriented Random Access Machine (bit-RAM)*. Information on this can be found in Section 1.4.

The bit-oriented approach is however not always the most appropriate one to model computational problems. For example, given a field k, one might enquire about the complexity to multiply two matrices with entries in k, measured in field operations in k. In this case, it does not seem to

be appropriate to demand that the field elements be represented by bit-strings (which is only possible when the field is countable anyway). Rather, one wishes to take what is called a "macroscopic standpoint starting from an idealized computer" in [BCS91]. One then enters the field of *algebraic complexity*. Models based on this approach are discussed in Section 1.6. Moreover, often computational problems can be formulated for arbitrary fields (for example the multiplication of two matrices). In this case, it is appropriate to ask for a uniform algorithm for all inputs over over all fields. We propose such a uniform computational model which we call a *generic field RAM*. Most of the results of Chapter 2 on basic computations related to curves and their divisors rely on variants of this model.

Additionally to presenting the bit-RAM and the generic field RAM model and randomized variants of these and deriving some basic properties of these models, we propose in this chapter a body of definitions related to computational problems.

1.2 Computational problems

We now propose some definitions which make it possible to talk about computational problems from the point of view of "mathematical content", that is, without having to put too much emphasis on the question of how the objects under consideration are represented for computational purposes, while on the other hand not being too imprecise. In particular, we propose a definition of the phrase "computational problem" which up to now has only been used in an intuitive way.

1.2.1 Representations

To motivate the following definitions, let us fix a (non-separably closed) field k and consider the problem to represent finite separable field extensions in terms of field elements in k and natural numbers. Every such extension is primitive, thus it is isomorphic to an extension of the form $(k[t]/(f(t)))|k$ with $f(t) \in k[t]$ separable and irreducible. Because of this, it is reasonable to say that "every finite separable extension of k can be represented by a polynomial in $k[t]$". In this statement, one might (but need not) add the information that the polynomials in question are separable and irreducible. This situation can also be described from a functorial point of view:

Example 1.1 We consider the set S of all separable irreducible polynomials in $k[t]$ and let $\boldsymbol{\mathcal{S}}$ be the category whose objects are the elements from S and which does not have any morphisms except the identities. Now the assignment $f(t) \mapsto k[t]/(f(t))$ is a functor from $\boldsymbol{\mathcal{S}}$ to the category of finite

separable extensions of k. The fact that every extension is isomorphic to an extension of the form $k[t]/(f(t))$ can be expressed by saying that this functor is *essentially surjective*.

Definition 1.2 Let $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{D}}$ be categories. Then a *partial functor* from $\boldsymbol{\mathcal{C}}$ to $\boldsymbol{\mathcal{D}}$ is a pair $(\boldsymbol{\mathcal{S}}, \mathcal{F})$, where $\boldsymbol{\mathcal{S}}$ is a subcategory of $\boldsymbol{\mathcal{C}}$ and $\mathcal{F} : \boldsymbol{\mathcal{S}} \longrightarrow \boldsymbol{\mathcal{D}}$ is a functor. We denote the pair $(\boldsymbol{\mathcal{S}}, \mathcal{F})$ by $\mathcal{F}$. The category $\boldsymbol{\mathcal{S}}$ is then called the *domain* of $\mathcal{F}$.

Recall that a *large groupoid* is a category whose only morphisms are isomorphisms. Note that the class of objects of such a category might or might not be a set. Recall also that a *full subcategory* $\boldsymbol{\mathcal{S}}$ of a category $\boldsymbol{\mathcal{C}}$ is a subcategory of $\boldsymbol{\mathcal{C}}$ such that for all $a, b \in \boldsymbol{\mathcal{S}}$, $\mathrm{Mor}_{\boldsymbol{\mathcal{S}}}(a, b) = \mathrm{Mor}_{\boldsymbol{\mathcal{C}}}(a, b)$.

Definition 1.3 Let $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{X}}$ be a large groupoids. Then a *partial representation* of $\boldsymbol{\mathcal{C}}$ by $\boldsymbol{\mathcal{X}}$ is a partial functor from $\boldsymbol{\mathcal{X}}$ to $\boldsymbol{\mathcal{C}}$ whose domain is a full subcategory of $\boldsymbol{\mathcal{C}}$. A *representation* of $\boldsymbol{\mathcal{C}}$ by $\boldsymbol{\mathcal{X}}$ is a partial representation which is essentially surjective.

Remark 1.4 All categories we consider in this chapter are large groupoids. The general philosophy is that representations and computational problems (see subsection 1.2.2) should in a certain sense be "well behaved" with respect to isomorphisms but not necessarily with respect to other morphisms. In applications of the terminology developed here the starting point are often general categories, and one applies the terminology to the associated large groupoids obtained by discarding all morphisms which are not isomorphisms. Another special case one often encounters is that one is given a set and one considers the associated groupoid whose objects are the elements of the set and which only has the identities as morphisms (see e.g. Example 1.1).

We also remark that one could just as well apply this above definition in a situation where only $\boldsymbol{\mathcal{X}}$ but not $\boldsymbol{\mathcal{C}}$ is a large groupoid. However, in subsection 1.2.2 we will need that $\boldsymbol{\mathcal{C}}$ is a large groupoid, and according to our general philosophy it does not cause any harm to impose this condition already now.

Definition 1.5 Let us fix a (partial) representation $\mathcal{F}$ of $\boldsymbol{\mathcal{C}}$ by $\boldsymbol{\mathcal{X}}$.

We say that $\boldsymbol{\mathcal{C}}$ *is via* $\mathcal{F}$ *(partially) represented by* $\boldsymbol{\mathcal{X}}$, and we then call $\boldsymbol{\mathcal{X}}$ the *category of representing objects* of $\boldsymbol{\mathcal{C}}$. We omit $\mathcal{F}$ if it is obvious from the context.

Now let furthermore a be an object of $\boldsymbol{\mathcal{C}}$ and x an object of $\boldsymbol{\mathcal{X}}$. If then $a \approx \mathcal{F}(x)$ we call x a *representing object* or a *representative* of a.

If the large groupoid $\boldsymbol{\mathcal{X}}$ is obtained from a set X by defining only the identities to be morphisms, we say that $\boldsymbol{\mathcal{C}}$ is (partially) represented by X. The same applies if the large groupoid $\boldsymbol{\mathcal{C}}$ is obtained from a set by defining only the identities to be morphisms.

Remark 1.6 Note that we can only "recover" an object of $\boldsymbol{\mathcal{C}}$ *up to isomorphism* from a representing object.

Terminology 1.7 Following the usual (intuitive) terminology in computational mathematics, if a is a representing object of x, we also say that "a represents x". We use similar expressions also to indicate that a large groupoid $\boldsymbol{\mathcal{C}}$ is represented by a large groupoid $\boldsymbol{\mathcal{X}}$.

For example, let k be a fixed field. Then instead of saying that we represent the *large groupoid* of finite separable field extensions of k by the *large groupoid* of separable irreducible polynomials, we might say that we represent finite separable field extensions of k by separable irreducible polynomials.

Let us now consider finite extensions of $\mathbb{F}_p(t)$ for some prime p. Not all extensions are separable, and in fact not all extensions are primitive. This means that the large groupoid of finite field extensions of $\mathbb{F}_p(t)$ is partially represented by the set of univariate polynomials over $\mathbb{F}_p(t)$.

Remark 1.8 Let both $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{C}}$ be obtained from sets X and C by defining only the identities to be morphisms. In this case, a partial representation of $\boldsymbol{\mathcal{C}}$ by $\boldsymbol{\mathcal{X}}$ is nothing but a partial map from X to C. A representation of $\boldsymbol{\mathcal{C}}$ by $\boldsymbol{\mathcal{X}}$ is a surjective partial map from X to C. (Note that partial representations correspond to partial maps but representations do *not* correspond to maps $X \longrightarrow C$ but as stated to surjective partial maps.)

Remark 1.9 Let $\boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{X}}$ be large groupoids. Let $(\boldsymbol{\mathcal{T}}, \mathcal{G})$ be a representation of $\boldsymbol{\mathcal{D}}$ by $\boldsymbol{\mathcal{C}}$, and let $(\boldsymbol{\mathcal{S}}, \mathcal{F})$ be a partial representation of $\boldsymbol{\mathcal{C}}$ by $\boldsymbol{\mathcal{X}}$. Then $\mathcal{G} \circ \mathcal{F} : \boldsymbol{\mathcal{S}} \longrightarrow \boldsymbol{\mathcal{D}}$ is a partial representation of $\boldsymbol{\mathcal{D}}$ by $\boldsymbol{\mathcal{X}}$. Moreover, if $\mathcal{F}$ is a representation, so is $(\boldsymbol{\mathcal{S}}, \mathcal{G} \circ \mathcal{F})$.

Let us again consider finite separable field extensions over a fixed field k. If we fix such an extension $\lambda|k$ and a primitive element $a \in \lambda$, we can represent each element of λ by its coordinate vector with respect to the "polynomial basis" $1, a, \ldots, a^{[\lambda:k]-1}$ defined by a. Now, as above, we want to vary the extension as well, and we want to use an irreducible separable polynomial $f(t) \in k[t]$ with $\lambda \approx k[t]/(f(t))$ to represent λ. All in all the task is now to "extend" the representation of separable extension fields by also representing an element in such a field.

This motives the following definition.

Definition 1.10 Let $\boldsymbol{\mathcal{C}} \longrightarrow \boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{X}} \longrightarrow \boldsymbol{\mathcal{Y}}$ be two functors between large groupoids. Then a *relative partial representation* of $\boldsymbol{\mathcal{C}} \longrightarrow \boldsymbol{\mathcal{D}}$ by $\boldsymbol{\mathcal{X}} \longrightarrow \boldsymbol{\mathcal{Y}}$ consists of a partial representation $(\boldsymbol{\mathcal{S}}, \mathcal{F})$ of $\boldsymbol{\mathcal{C}}$ by $\boldsymbol{\mathcal{X}}$ and a partial representation $(\boldsymbol{\mathcal{T}}, \mathcal{G})$ of $\boldsymbol{\mathcal{D}}$ by $\boldsymbol{\mathcal{Y}}$ such that the image of S in $\boldsymbol{\mathcal{Y}}$ is contained in $\boldsymbol{\mathcal{T}}$ and such that the diagram

$$\begin{array}{ccc} \boldsymbol{\mathcal{C}} & \longrightarrow & \boldsymbol{\mathcal{D}} \\ {\scriptstyle\mathcal{F}}\uparrow & & \uparrow{\scriptstyle\mathcal{G}} \\ \boldsymbol{\mathcal{S}} & \longrightarrow & \boldsymbol{\mathcal{T}} \\ \downarrow & & \downarrow \\ \boldsymbol{\mathcal{X}} & \longrightarrow & \boldsymbol{\mathcal{Y}} \end{array}$$

is commutative. A *relative representation* of $\boldsymbol{\mathcal{C}} \longrightarrow \boldsymbol{\mathcal{D}}$ by $\boldsymbol{\mathcal{X}} \longrightarrow \boldsymbol{\mathcal{Y}}$ is a relative partial representation such that both $\mathcal{F}$ and $\mathcal{G}$ are essentially surjective.

Example 1.11 We return to the example of finite separable field extensions over a fixed field k. Let now $\boldsymbol{\mathcal{D}}$ be the large groupoid of such extensions, and let $\boldsymbol{\mathcal{C}}$ be the large groupoid whose objects are tuples $(\lambda|k, a)$, where $\lambda|k$ is a finite separable field extension and $a \in \lambda$, and whose (iso-)morphisms are defined as follows: A morphism from $(\lambda|k, a)$ to $(\lambda'|k, a')$ is an isomorphism of field extensions $\lambda \longrightarrow \lambda'$ which maps a to a'. Now the large groupoid $\boldsymbol{\mathcal{C}}$ is represented by the set of all polynomials in $k[t]$. The large groupoid $\boldsymbol{\mathcal{D}}$ is represented by the set of tuples $(f(t), \underline{a})$, where $f(t) \in k[t]$ and $\underline{a} \in k^{[\lambda:k]}$. (Here $\underline{a}$ represents the element in $k[t]/(f(t))$ given by $\sum_{i=1}^{t} a_i t^{i-1}$.) Obviously we obtain a relative representation of $\boldsymbol{\mathcal{C}} \longrightarrow \boldsymbol{\mathcal{D}}$.

Definition 1.12 Let $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{X}}$ be a large groupoids and $(\boldsymbol{\mathcal{S}}, \mathcal{F})$ a partial representation of $\boldsymbol{\mathcal{C}}$ by $\boldsymbol{\mathcal{X}}$. Then the *category of represented objects* of $\boldsymbol{\mathcal{C}}$ by $\boldsymbol{\mathcal{X}}$ is is defined as follows: The objects are tuples (a, x), where a is an object of $\boldsymbol{\mathcal{C}}$ and x is an object of $\boldsymbol{\mathcal{S}}$ such that $a \approx \mathcal{F}(x)$. We define the morphisms by $\mathrm{Mor}((a, x), (b, y)) := \mathrm{Mor}(x, y)$.

Note that by definition we have a fully faithful functor from the domain of $\mathcal{F}$ to the category of represented objects of $\boldsymbol{\mathcal{C}}$ by $\boldsymbol{\mathcal{X}}$. Moreover, if $\mathcal{F}$ is a representation, we have an equivalence of categories between the domain of $\mathcal{F}$ and $\boldsymbol{\mathcal{C}}$. Because of this, one might question the relevance of this definition. However, it is of great help for the intuition to think about objects of $\boldsymbol{\mathcal{C}}$ with some "extra data for the representation" rather than merely the barren objects used for the representation themselves.

1.2.2 A formalism for computational problems

The following two definitions relate our abstract definitions on representability to the classical theory of computability. Here, by "classical theory" we mean the theory which is based on Turing machines or equivalent models (from the point of computability). In Section 1.6 analogous definitions in an "algebraic setting" are given.

Definition 1.13 Let $\boldsymbol{\mathcal{C}}$ still be a large groupoid. A *bit-representation* of $\boldsymbol{\mathcal{C}}$ is a representation of $\boldsymbol{\mathcal{C}}$ by $\mathbb{N}_0$.[1]

Remark 1.14 We often use the injection $\mathbb{N}_0 \longrightarrow \{0,1\}^*$ given by the 2-adic expansion. (That is, $n = \sum_{i=0}^{k-1} n_i 2^i$ with $n_i \in \{0,1\}$ for all $i = 1, \ldots, k-2$ and $n_{k-1} = 1$ is mapped to the string $n_0 n_1 \cdots n_{k-1}$.) This also explains the name "bit-representation".

Definition 1.15 Let $S \subseteq \mathbb{N}_0$, and let $c : S \longrightarrow \mathbb{N}_0$. Then we say that c is *computable* if there exists a Turing machine (or a machine based on an equivalent model from the point of view of computability) M which for every $x \in S$ computes $c(x)$. We then also say that M *computes* c.

Remark 1.16 Note that we do not impose any condition for $s \in \mathbb{N}_0 - S$, and in particular we do not require that the machine halts for $s \in \mathbb{N}_0 - S$.

For the following definition note that for every category $\boldsymbol{\mathcal{C}}$ we have the *large category of full subcategories* of $\boldsymbol{\mathcal{C}}$. The objects of this category are the full subcategories of $\boldsymbol{\mathcal{C}}$ and the morphisms between two such subcategories $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{S}}'$ are the functors from $\boldsymbol{\mathcal{S}}$ to $\boldsymbol{\mathcal{S}}'$. With the natural transformations this category is in fact a 2-category.

Definition 1.17 Let $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{C}}'$ be large groupoids, and let $\mathcal{F}$ be a bit-representation of $\boldsymbol{\mathcal{C}}$ with domain $S \subseteq \mathbb{N}_0$ and $\mathcal{F}'$ be a bit-representation of $\boldsymbol{\mathcal{C}}'$.

Then a *computational problem* from $\boldsymbol{\mathcal{C}}$ to $\boldsymbol{\mathcal{C}}'$ (with respect to the fixed representations) consists of:

- a functor $\mathcal{P}$ from $\boldsymbol{\mathcal{C}}$ to the category of full subcategories of $\boldsymbol{\mathcal{C}}'$ (for $\alpha : a \longrightarrow a'$ we write $\mathcal{P}_\alpha$ for the associated functor from $\mathcal{P}(a)$ to $\mathcal{P}(a')$)
- for every morphism $\alpha : a \longrightarrow a'$ of $\boldsymbol{\mathcal{C}}$ and every $b \in \mathcal{P}(a)$ a morphism $m_{\alpha,b} : b \longrightarrow \mathcal{P}_\alpha(b)$

such that

[1] We set $\mathbb{N}_0 := \mathbb{Z}_{\geq 0}$.

- for all $a \in \boldsymbol{\mathcal{C}}$, $\mathcal{P}(a)$ is non-empty
- all diagrams of the form

$$\begin{array}{ccc} b & \xrightarrow{\beta} & b' \\ \downarrow{\scriptstyle m_{\alpha,b}} & & \downarrow{\scriptstyle m_{\alpha,b'}} \\ \mathcal{P}_\alpha(b) & \xrightarrow{\mathcal{P}_\alpha(\beta)} & \mathcal{P}_\alpha(b') \end{array}$$

with $\alpha : a \longrightarrow a'$, $b, b' \in P(a)$ and $\beta : b \longrightarrow b'$ commute.

We denote a computational problem as above by $\mathcal{P}$. Let now such a computational problem be given. Then we call the objects from $\boldsymbol{\mathcal{C}}$ the *instances* of the problem and the elements from S *legitimate inputs*. We say that the computational problem (*problem* for short) is *computable* if there exists a computable function $c : S \longrightarrow \mathbb{N}_0$ such that for all $x \in S$, $\mathcal{F}'(c(x))$ is isomorphic to an object in $\mathcal{P}(\mathcal{F}(x))$. Moreover, if we have a (Turing or other) machine M which computes c, we also say that M *computes* $\mathcal{P}$. (Note that we only require termination for legitimate inputs by Definition 1.15.)

Remark 1.18 Note again that $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{C}}'$ are large groupoids. This implies that all morphisms in $\mathcal{P}(a)$ and all morphisms $m_{\alpha,b}$ are isomorphisms.

It follows immediately that the morphisms

$$(\mathcal{P}_{\alpha'} \circ \mathcal{P}_\alpha)(b) \xrightarrow{(m_{\alpha',\mathcal{P}_\alpha(b)})^{-1}} \mathcal{P}_\alpha(b) \xrightarrow{(m_{\alpha,b})^{-1}} b \xrightarrow{m_{\alpha'\alpha,b}} \mathcal{P}_{\alpha'\alpha}(b)$$

define an isomorphism of functors between $\mathcal{P}_{\alpha'} \circ \mathcal{P}_\alpha$ and $\mathcal{P}_{\alpha' \circ \alpha}$.

Note also that the definition of $m_{\alpha,\cdot} : \mathcal{P}(a) \longrightarrow \mathcal{P}(a')$ is analogous to the definition of an isomorphism of functors (it is an isomorphism of functors from $\mathrm{id}_{\mathcal{P}(\alpha)}$ to $\mathcal{P}_\alpha$ if and only if $\mathcal{P}(a) = \mathcal{P}(a')$).

We have the following three classes of examples:

Example 1.19 Let $\boldsymbol{\mathcal{C}}'$ be obtained from a set C' by defining only the identities to be morphisms, a computational problem from $\boldsymbol{\mathcal{C}}$ to $\boldsymbol{\mathcal{C}}'$ is nothing but an assignment P from the objects of $\boldsymbol{\mathcal{C}}$ to $\mathbb{P}(C')$ with $P(a) = P(a')$ for $a \approx a'$. (Here for a set X $\mathbb{P}(X)$ is the power set of X.) In particular, if $\boldsymbol{\mathcal{C}}$ is also obtained from a set C by defining only the identities to be morphisms, a computational problem from $\boldsymbol{\mathcal{C}}$ to $\boldsymbol{\mathcal{C}}'$ is the same as a map $C \longrightarrow \mathbb{P}(C)$.

Example 1.20 Let $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{C}}'$ be large groupoids with fixed representations. Then every functor $P : \boldsymbol{\mathcal{C}} \longrightarrow \boldsymbol{\mathcal{C}}'$ induces a computational problem from $\boldsymbol{\mathcal{C}}$ to $\boldsymbol{\mathcal{C}}'$ as follows:

The assignment on the objects is given by assigning to every object $a \in \boldsymbol{\mathcal{C}}$ the class consisting of $P(a)$. For every morphism $\alpha : a \longrightarrow a'$ of $\boldsymbol{\mathcal{C}}$ the functor P_α is given by $P_\alpha(P(a)) := P(a')$ on the objects and by $P_\alpha(\beta) := P(\alpha) \circ \beta \circ P(\alpha)^{-1}$ for every isomorphism β of $P(a)$. Furthermore, we set $m_{\alpha,P(a)} := P(\alpha)$.

Example 1.21 Let again $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{C}}'$ be large groupoids with fixed representations, and let $p : \boldsymbol{\mathcal{C}}' \longrightarrow \boldsymbol{\mathcal{C}}$ be a functor such that for every (iso-)morphism $\alpha : a \longrightarrow a'$ and every b in the fiber over a, there exists an object b' in the fiber over a' and an (iso-)morphism $\beta : b \longrightarrow b'$ with $\alpha \circ p = p \circ \beta$. (Note that $\boldsymbol{\mathcal{C}}$ with p is a special case of a category fibered in groupoids.)

Then we obtain as follows a computational problem $\mathcal{P}$ from $\boldsymbol{\mathcal{C}}$ to $\boldsymbol{\mathcal{C}}'$: On objects the functor $\mathcal{P}$ is defined by assigning to $a \in \boldsymbol{\mathcal{C}}$ its fiber in $\boldsymbol{\mathcal{C}}'$. Now let $\alpha : a \longrightarrow a'$ be a morphism in $\boldsymbol{\mathcal{C}}$. Then for each object b of $\boldsymbol{\mathcal{C}}'$ over a we choose some object $\mathcal{P}_\alpha(b)$ over a' in $\boldsymbol{\mathcal{C}}'$ together with some isomorphism $m_{\alpha,b} : b \longrightarrow \mathcal{P}_\alpha(b)$ over α.[2] Here, if $a' = a$ and $\alpha = \mathrm{id}_a$, we choose $\mathcal{P}_\alpha(b) = b$ and $m_{\alpha,b} = \mathrm{id}_b$. Finally, for $\beta : b \longrightarrow b'$ in $\mathcal{P}(a)$, we set $\mathcal{P}_\alpha(\beta) := m_{\alpha,b'} \circ \beta \circ m_{\alpha,b}^{-1}$. This choice of b' and β defines the functor $\mathcal{P}_\alpha$ and the assignments $m_{\alpha,.}$.

Arguably the most important example of a computational problem related to this work is the general discrete logarithm problem for class groups of curves, which can be defined as follows:

Example 1.22 We consider the large groupoid of all triples $(\mathcal{C}/\mathbb{F}_q, a, b)$ where $\mathcal{C}$ is a curve over $\mathbb{F}_q$, $a, b \in \mathrm{Cl}^0(\mathcal{C})$ with $b \in \langle a \rangle$, where the (iso-)morphisms $(\mathcal{C}, a, b) \longrightarrow (\mathcal{C}', a', b')$ are isomorphisms $\varphi : \mathcal{C} \longrightarrow \mathcal{C}'$ with $\varphi^*(a') = a, \varphi^*(b') = b$. The methods of the next chapter give various bit-representations of this large groupoid; let us choose one. We now consider the set of non-negative integers $\mathbb{N}_0$ which we turn into a category by defining only the identities to be morphisms, and we represented it by itself via the identity. The computational problem is now the assignment $(\mathcal{C}/\mathbb{F}_q, a, b) \mapsto x$, where x is the smallest non-negative integer with $x \cdot a = b$.

Example 1.23 As an example of a computational problem which is not induced by a functor as in Example 1.20 consider the following problem: We represent $\mathbb{Q}$ by $\mathbb{Z} \times \mathbb{N}$ via $(n, d) \mapsto \frac{n}{d}$, and we represent $\mathbb{Z} \times \mathbb{N}$ (essentially) by concatenation of the bit-strings of the two numbers. Let us fix a dense representation of vectors in $\mathbb{Q}^n$ for arbitrary $n \in \mathbb{N}$ ((essentially) given by concatenation of the representations of the elements), and let us represent univariate polynomials over $\mathbb{Q}$ by the corresponding coefficient vectors. Now

[2] Here we apply the axiom of choice. In order not to run into logical problems, one should first choose a universe and restrict oneself to sets in this universe.

consider the problem to determine if a given univariate polynomial over $\mathbb{Q}$ has a root in $\mathbb{Q}$ and if this is the case to compute one.

Formally, this computational problem is defined as follows: $\boldsymbol{\mathcal{C}}$ is the large groupoid obtained from the set $\mathbb{Q}[t]$ and $\boldsymbol{\mathcal{C}'}$ is the large groupoid obtained from the set $\mathbb{Q} \dot\cup \{\text{“No”}\}$. The computational problem is given by the map $\mathbb{Q}[t] \longrightarrow \mathbb{P}(\mathbb{Q} \dot\cup \{\text{“No”}\})$ mapping a polynomial to {”No”} if has no roots in $\mathbb{Q}$ and to the set of roots otherwise.

Definition 1.24 Let a functor $\boldsymbol{\mathcal{C}} \longrightarrow \boldsymbol{\mathcal{D}}$ be given. Then a *(partial) relative bit-representation* of $\boldsymbol{\mathcal{C}} \longrightarrow \boldsymbol{\mathcal{D}}$ is a (partial) relative representation of $\boldsymbol{\mathcal{C}} \longrightarrow \boldsymbol{\mathcal{D}}$ by the projection to the first coordinate $p_1 : \mathbb{N}_0^2 \longrightarrow \mathbb{N}_0$.

In the following, we represent $(n_1, n_2) \in \mathbb{N}_0^2$ essentially by concatenation of the two bit-strings of n_1 and n_2. More precisely, let $n_{1,0} \cdots n_{1,k-1}$ be the bit-string for n_1 and $n_{2,0} \cdots n_{2,\ell-1}$ the bit-string for n_2. Then we represent (n_1, n_2) by the string $n_{1,0}0n_{1,1}0 \cdots 0n_{1,k-1}1n_{2,0}0 \cdots 0n_{2,\ell-1}$. (We always write a bit followed by a 0 until we reach the end of n_1. Here we write a 1, then we alternatingly write a bit of n_2 and 0.)

In the context of Definition 1.24 this means that an object x of $\boldsymbol{\mathcal{C}}$ is represented by a bit-representation of its image in $\boldsymbol{\mathcal{D}}$ concatenated with a further bit-string.

Definition 1.25 Let $\boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{C}'}$ and $\boldsymbol{\mathcal{D}}$ be a large groupoids, and let $p : \boldsymbol{\mathcal{C}} \longrightarrow \boldsymbol{\mathcal{D}}$ and $p' : \boldsymbol{\mathcal{C}'} \longrightarrow \boldsymbol{\mathcal{D}}$ be functors. Let relative bit-representations of $\boldsymbol{\mathcal{C}} \longrightarrow \boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{C}'} \longrightarrow \boldsymbol{\mathcal{D}}$ be given which induce the same bit-representation on $\boldsymbol{\mathcal{D}}$. Let $(S, \mathcal{F})$ be the representation of $\boldsymbol{\mathcal{C}}$ by $\mathbb{N}_0^2$, $(S', \mathcal{F}')$ the representation of $\boldsymbol{\mathcal{C}'}$ by $\mathbb{N}_0^2$ and $(T, \mathcal{G})$ the representation of $\boldsymbol{\mathcal{D}}$ by $\mathbb{N}_0$. Then a *relative computational problem* from $\boldsymbol{\mathcal{C}}$ to $\boldsymbol{\mathcal{C}'}$ over $\boldsymbol{\mathcal{D}}$ (with respect to $\mathcal{F}$, $\mathcal{F}'$ and $\mathcal{G}$) is a computational problem $\mathcal{P}$ from $\boldsymbol{\mathcal{C}}$ to $\boldsymbol{\mathcal{C}'}$ which satisfies that $p'(\mathcal{P}(x)) = \{p(x)\}$ for all $x \in \boldsymbol{\mathcal{C}}$.

We say that such a problem $\mathcal{P}$ is *computable* if there exists a computable function $c : S \longrightarrow \mathbb{N}$ with $\mathcal{F}'(x, c(x)) \in \mathcal{P}(\mathcal{F}(x))$ for all $x \in S$. A (Turing or other) machine which computes c is again said to *compute* the relative computational problem.

Remark 1.26 Let $\boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{C}'}, \boldsymbol{\mathcal{D}}, p$ and p' be as above. Then similarly as for computational problems, every functor $P : \boldsymbol{\mathcal{C}} \longrightarrow \boldsymbol{\mathcal{C}'}$ with $p \circ P = p'$ induces a relative computational problem.

Remark 1.27 The computational problem defined in Example 1.21 defines a relative computational problem of $\boldsymbol{\mathcal{C}} \longrightarrow \boldsymbol{\mathcal{C}'}$ over $\boldsymbol{\mathcal{C}}$.

Example 1.28 We continue with Example 1.11. As in this example, let $\boldsymbol{\mathcal{D}}$ be the large groupoid of finite separable field extensions of a fixed field

k. Similarly to but slightly different from Example 1.11, let $\boldsymbol{\mathcal{C}}$ be the large groupoid of tuples $(\lambda|k, (a, b))$, where $\lambda|k$ is a finite separable field extension and $(a, b) \in \lambda^2$ and $\lambda|k$. Moreover, let $\boldsymbol{\mathcal{C}'}$ be the large groupoid of tuples $(\lambda|k, c)$ where $c \in \lambda$ and $\lambda|k$ is a finite separable field extension (both categories with the obvious (iso-)morphisms).

Now let k be a countable field, and let us fix a bit-representation of k such that the arithmetic operations in k are computable. This bit-representation induces a bit-representation of the polynomials over k and vectors in k^n for $n \in \mathbb{N}$. Following Example 1.11, we obtain relative bit-representations of $\boldsymbol{\mathcal{C}} \longrightarrow \boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{C}'} \longrightarrow \boldsymbol{\mathcal{D}}$.

Now we can describe the multiplication in finite separable field extensions of k via the functor $(\lambda|k, a, b) \mapsto (\lambda, a \cdot b)$. Clearly, this functor defines a relative computational problem which is computable.

1.3 The $\mathcal{O}$-notation

The $\mathcal{O}$-notation is of course classical; we present it here with an approach which is closely related to the definitions of the previous section.

Let $\boldsymbol{\mathcal{C}}$ be a large groupoid. In the following, we denote large groupoids associated to sets by defining only the identities to be morphism by boldface letters.

Let $f : \boldsymbol{\mathcal{C}} \longrightarrow \mathbf{R}_{\geq \mathbf{0}}$ be a functor. Then we define the class $\mathcal{O}(f)$ by

$$\mathcal{O}(f) := \big\{ g : \boldsymbol{\mathcal{C}} \longrightarrow \mathbf{R} \text{ a functor } \mid \exists c > 0 : |g(A)| \leq c \cdot f(A) \text{ for all objects } A \text{ of } \boldsymbol{\mathcal{C}} \big\} .$$

We define

$$\tilde{\mathcal{O}}(f) := (\log_2(\max\{f, 2\}))^{\mathcal{O}(1)} \cdot \mathcal{O}(f) .$$

Here – and also in the following – we use the obvious generalizations of the usual definitions for sets of functions from some set C to $\mathbb{R}$ to classes of functors from $\boldsymbol{\mathcal{C}}$ to $\mathbf{R}$.

Similar definitions and considerations also apply to the Ω-notation (denoting lower bounds) and the Θ-notation (denoting both upper and lower bounds). We note that we will not use the usual "Landau-style" notation "$f = \mathcal{O}(g)$" for $g \in \mathcal{O}(f)$.

For a functor $f : \boldsymbol{\mathcal{C}} \longrightarrow \mathbf{R}_{\geq \mathbf{0}}$ we define

$$\mathcal{P}oly(f) := \big\{ g : \boldsymbol{\mathcal{C}} \longrightarrow \mathbf{R} \text{ a functor } \mid \exists p(X) \in \mathbf{R}[X] \text{ with non-negative coefficients: } |g(A)| \leq p(f(A)) \text{ for all objects } A \text{ of } \boldsymbol{\mathcal{C}} \big\} ,$$

and if $g \in \mathcal{P}oly(f)$, we say that g is *polynomially bounded in* f. More generally, for functors $f_1, \ldots, f_k : \boldsymbol{\mathcal{C}} \longrightarrow \mathbf{R}_{\geq \mathbf{0}}$ we define

$$\begin{aligned}\mathcal{P}oly(f_1, \ldots, f_k) := \big\{ g : \boldsymbol{\mathcal{C}} \longrightarrow \mathbf{R} \text{ a functor } \mid \exists p(X) \in \mathbf{R}[X_1, \ldots, X_n] \\ \text{with non-negative coefficients:} \\ |g(A)| \leq p(f_1(A), \ldots, f_n(A)) \\ \text{for all objects } A \text{ of } \boldsymbol{\mathcal{C}} \big\} ,\end{aligned}$$

and if $g \in \mathcal{P}oly(f_1, \ldots, f_k)$, we say that g is *polynomially bounded in* $f_1, \ldots, f_k$.

Moreover, we fix the following generalization of the previous $\mathcal{O}$-notation (which we do however not use in the sequel).

Let $m_1, \ldots, m_k : \boldsymbol{\mathcal{C}} \longrightarrow \mathbf{R}$ be functors. Now let $(\boldsymbol{\mathcal{S}}, f)$ be partial functor from $\boldsymbol{\mathcal{C}}$ to $\mathbf{R}_{\geq \mathbf{0}}$, where $\boldsymbol{\mathcal{S}}$ is a full subcategory of $\boldsymbol{\mathcal{C}}$ such that there exist $C_1, \ldots, C_k > 0$ with $A \in \boldsymbol{\mathcal{S}}$ for all objects A of $\boldsymbol{\mathcal{C}}$ with $m_1(A) \geq C_1, \ldots, m_k(A) \geq C_k$.

Then we define

$$\begin{aligned}\mathcal{O}_{m_1,\ldots,m_k}(f) := \big\{ g : \boldsymbol{\mathcal{T}} \longrightarrow \mathbf{R} \text{ a functor } \mid \boldsymbol{\mathcal{T}} \text{ a large subcategory of } \boldsymbol{\mathcal{S}} \\ \text{such that there exist } c > 0, C_1, \ldots, C_k > 0 \\ \text{with } A \in \boldsymbol{\mathcal{T}} \text{ and } |g(A)| \leq c \cdot f(A) \\ \text{for all objects } A \text{ of } \boldsymbol{\mathcal{C}} \text{ with} \\ m_1(A) \geq C_1, \ldots, m_k(A) \geq C_k \big\} .\end{aligned}$$

For a partial functor g from $\boldsymbol{\mathcal{C}}$ to $\mathbf{R}$ we express $g \in \mathcal{O}_{m_1,\ldots,m_k}(f)$ by saying "$g \in \mathcal{O}(f)$ for $m_1, \ldots, m_k \longrightarrow \infty$". Similarly to above we set

$$\tilde{\mathcal{O}}_{m_1,\ldots,m_k}(f) := (\log_2(\max\{f, 2\}))^{\mathcal{O}_{m_1,\ldots,m_k}(1)} \cdot \mathcal{O}_{m_1,\ldots,m_k}(f) .$$

Remark 1.29 With these definitions one has particular the following intuitive compatability: Let $\boldsymbol{\mathcal{C}}$ be a large groupoid, and let $f : \boldsymbol{\mathcal{C}} \longrightarrow \mathbf{R}_{\geq \mathbf{0}}$ and $g : \boldsymbol{\mathcal{C}} \longrightarrow \mathbf{R}$ be functors. Now let $\boldsymbol{\mathcal{X}}$ be another large groupoid, and let $(\boldsymbol{\mathcal{S}}, \mathcal{F})$ be a representation of $\boldsymbol{\mathcal{C}}$ by $\boldsymbol{\mathcal{X}}$. Then $g \in \mathcal{O}(f)$ if and only if $g \circ \mathcal{F} \in \mathcal{O}(f \circ \mathcal{F})$. A similar statement holds if $m_1, \ldots, m_k$ are considered too, and a similar statement holds for $\mathcal{P}oly(f_1, \ldots, f_k)$.

In applications, one does not distinguish between the functors f, g and values of these functors. (E.g., one considers the genus of curves, which one denotes by g or the degree of plane models which one denotes by d and so on.) Because of the compatibility this does not cause any problems, even if one completely suppresses the domain of the functors. (Note that this is similar to the situation one encounters with probability spaces and random variables, where the probability spaces are often completely suppressed.)

Remark 1.30 In [vzGG03] the $\mathcal{O}$-notation is defined as follows: A partial function f from $\mathbb{N}$ to $\mathbb{R}$ is called *eventually positive* if there exists some

$N > 0$ such that for $n \geq N$, $f(n)$ is defined and (strictly) positive. Then for such a function f the set $\mathcal{O}(f)$ is the set of all partial functions from $\mathbb{N}$ to $\mathbb{R}$ which are eventually positive and such that there exist $N, c \in \mathbb{N}$ with $g(n) \leq c \cdot f(n)$ for all $n \geq N$.

We would like to comment on this definition and the relationship with our definition above.

First, it seems to be convenient to us to change this definition as follows: Given some f which is eventually positive, one defines $\mathcal{O}(f)$ as the set of all partial functions from $\mathbb{N}$ to $\mathbb{R}$ which are defined almost everywhere such that there exists $N, c \in \mathbb{N}$ with $|g(n)| \leq c \cdot f(n)$ for all $n \geq N$.

Now a generalization of this definition from $\mathbb{N}$ to large groupoids would be: Let $\boldsymbol{\mathcal{C}}$ be a large groupoid and let f be a functor which is defined on a full subcategory $\boldsymbol{\mathcal{S}}$ of $\boldsymbol{\mathcal{C}}$ such that there are up to isomorphism only finitely many objects of $\boldsymbol{\mathcal{C}}$ which are not contained in $\boldsymbol{\mathcal{S}}$, and let f be a functor from $\boldsymbol{\mathcal{S}}$ to $\mathbf{R}_{>\mathbf{0}}$. Then one might define

$$\begin{aligned}\mathcal{O}(f) := \{g : \boldsymbol{\mathcal{T}} \longrightarrow \mathbb{R} \mid\ & \boldsymbol{\mathcal{T}} \text{ a full subcategory of } \boldsymbol{\mathcal{S}} \\ & \text{such that there are up to isomorphism} \\ & \text{only finitely many objects of } \boldsymbol{\mathcal{C}} \\ & \text{which are not contained in } \boldsymbol{\mathcal{T}} \\ & \text{and such that } \exists c > 0 : |g(A)| \leq f(A) \\ & \text{for all objects } A \text{ of } \boldsymbol{\mathcal{T}}\}\ .\end{aligned}$$

However, this definition is problematic for the following reason: As mentioned in the previous remark, in applications of the $\mathcal{O}$-notation one often does not fully define which category one considers. Therefore in particular it is not clear how the isomorphism classes are defined, and it is then not clear what is meant by the condition that f and g in the definition are not defined on finitely many isomorphism classes. Note that in particular, with this definition one does not have the compatibility we mentioned in the previous remark.

We would like to comment on a particular use of the $\mathcal{O}$-notation in the literature which we consider to be a misuse. Often the $\mathcal{O}$-notation is used as follows: Let $\boldsymbol{\mathcal{C}}$ be a large groupoid and let f_1 and f_2 be two functions $\boldsymbol{\mathcal{C}} \longrightarrow \mathbf{R}_{\geq \mathbf{0}}$.[3] Then by "$g = \mathcal{O}(f_1 f_2)$" authors often mean that there exists some $c > 0$ such that $g(n) \leq c \cdot f_1(n) \cdot f_2(n)$ *for* $f_1(n)$ *and* $f_2(n)$ *large enough.* This is however not well-defined since the product $f_1 f_2$ is also merely a functor on $\boldsymbol{\mathcal{C}}$. In fact, this use of the $\mathcal{O}$-notation means in particular that the statement "$g = \mathcal{O}(f_1 f_2)$" has a different meaning than the statement "for $f := f_1 f_2$ we have $g = \mathcal{O}(f)$", which is an unfortunate situation.

[3] Of course, the large groupoid is not defined, but this is not what we would like to focus on here.

For example, often in works on the manipulation of square matrices with polynomial entries one finds statements that certain computations can be performed with $\mathcal{O}(n^\alpha \cdot d^\beta)$ field operations, where n is the size of the matrix and d the maximum of the degrees of the entries and α, β are constants. However, for $d = 0$ there are then often *infinitely many matrices* for which the bound does not hold, and therefore the statement is incorrect, even if one uses the definition of the $\mathcal{O}$-notation which is inspired by the definition in [vzGG03].

1.4 Bit-complexity

So far we have not addressed the first two questions posed at the beginning of this chapter: We still have to fix a particular computational model and complexity measure / cost function. In this section, we propose a particular computational model to capture the intuitive meaning of "bit-complexity".

On the basis of this model, we can then in particular talk about the complexity of a computational problem as defined in Section 1.2.

1.4.1 Some comments on bit-complexity

In the works on computational number theory, often no or very limited information is given on the underlying computational model and an intuitive approach is taken with respect to computational complexity: In a certain sense bit-operations needed for arithmetic operations are counted, but usually operations on the storage of the machine are assumed to be for free or essentially for free. It arises the question which of the various computational models in the literature capture this intuitive notion of "bit-complexity" appropriately.

At the first sight, one might think of multiple string Turing machines because this model is truly bit-oriented. However, on the other hand, one wants to be able to have quick access to the storage, and in particular indirect addressing should be possible, a feature which is not present in Turing machines.

These important features are present in the various Random Access Machine (RAM) models, and if nothing else is stated explicitly in a certain work on algorithms, it seems to be fair to interpret the claims with respect to an appropriate RAM. Let us recall the general idea of a Random Access Machine on an intuitive level:

The machine has registers $r_i\,(i \in \mathbb{N}_0)$, each of which can store integers or non-negative integers of arbitrary length (depending on the model). Furthermore, the machine has a program which is described with a rather

limited programming language. The register r_0 serves as an *accumulator* and contains the results of various instructions. An important feature of the programming language is that it allows indirect addressing: Let $r(i)$ be the content of register i at a particular time during the execution of the program. Then for every $i \in \mathbb{N}_0$, one can load $r(r(i))$, the content of register $r_{r(i)}$, into the accumulator.

It arises however a problem because there is a certain level of arbitrariness with respect to the possible instructions and the cost measure of the RAM.

In order to cope with this problem of arbitrariness we propose as a computational model a Random Access Machines whose arithmetic commands are obtained from Turing machines. This means that from a "macroscopic perspective" such a machine operates as the RAMs in the literature but the arithmetic operations are performed by Turing machines. An informal description of such machines, which we call *bit-oriented Random Access Machines (bit-RAMs)*, is as follows: The machine has registers r_i ($i \in \mathbb{N}_0$) and a program. The instructions in the program are of two types. The first type of instructions is based on the commands LOAD, STORE, GOTO, IF ... GOTO and END which have the same meaning as usual in the literature. Now additionally, for every multiple string Turing machine T with multiple input strings and one output string, we introduce a command c_T. The execution of this command is modeled by the Turing machine T. The running time in bit operations of the bit-RAM upon a certain input is then the sum of the running times for the executions of the Turing machine-related commands plus some cost for storage management.

1.4.2 Bit oriented Random Access Machines

We first fix a definition of the Turing machines which give rise to the arithmetic commands of the bit oriented RAM model. This definition is closely related to the definition of a multiple string Turing machine with with input and output string in [Pap94].

We fix a 4-element set Σ with elements $0, 1, \sqcup, \triangleright$, called the *alphabet* (the elements of Σ are called *symbols*). The element $\sqcup$ is called *blank symbol*, and the element $\triangleright$ is called *first symbol*. Furthermore, we fix a 3-element set with elements $\leftarrow, -, \rightarrow$, called the set of *directions*.

Definition 1.31 A *bit-oriented multiple string Turing machine with multiple input strings and one output string* consists of

- two natural numbers k, ℓ with $\ell < k$, the *number of strings* and the *number of input strings*

- a finite set Q, the *state space*
- an element $q_0 \in Q$, the *initial state* and an element $h \in Q$, the *halting state*
- a map $\delta : Q \times \Sigma^k \longrightarrow Q \times (\Sigma \times \{\leftarrow, -, \rightarrow\})^k$, the *transition function*

such that the following holds: Let $q \in Q$ and $(\sigma_1, \ldots, \sigma_k) \in \Sigma^k$, and let $\delta(q, \sigma_1, \ldots, \sigma_k) = (q', \sigma'_1, D_1, \ldots, \sigma'_k, D_k)$ with $q' \in Q, \sigma'_i \in \Sigma$ and $D_i \in \{\leftarrow, -, \rightarrow\}$. Then:

- For all $i = 1, \ldots, k$: $\sigma_i = \triangleright$ iff $\sigma'_i = \triangleright$, and if this is the case then $D_i = \rightarrow$.
- If $q = h$, then $q' = h$, $\sigma'_i = \sigma_i$ and $D_i = -$ for all i.
- $\sigma_k \in \{0, 1\}$ and $D_k \neq \leftarrow$.
- For all $i = 1, \ldots, \ell$, $\sigma'_i = \sigma_i$.

The four requirements can be interpreted as follows (see also the definitions of configuration and starting configuration below).

- Every string starts with a $\triangleright$ (which does not occur anywhere else in the string), and if the head hits this symbol, it recognizes that it is at the beginning of the string and moves right again.
- If the machine has reached the halting state, no more operations are performed.
- The k^{th} tape is write-only, and the k^{th} string has no blanks.
- Tapes $1, \ldots, \ell$ are read only.

A *configuration* is a tuple $(q, w_1, u_1, \ldots, w_k, u_k)$ where q is a state and $w_i, u_i \in \Sigma^*$ such that for all i, the first symbol of w_i is $\triangleright$, and this symbol occurs nowhere else in w_i and u_i. The set of configurations is called *configuration space*. The intuition of a configuration is that it indicates the current state q of the machine as well as the current string left to the reading head (including the symbol under the head) and the current string right to the reading head (without the symbol under the head).

The transition function in an obvious way leads to an operation on the configurations space. (If a reading head "falls off" the right side of a string, a $\sqcup$ is written, in particular strings cannot become shorter during the computation; cf. [Pap94] for details.)

Let for $\alpha \in \mathbb{N}_0$ $S(\alpha)$ be the corresponding bit-string (see also Remark 1.14).

Now let T be a bit-oriented multiple string Turing machine with ℓ input tapes, and let $\underline{\alpha} \in \mathbb{N}_0^\ell$. Then we define the *starting configuration* associated to $\underline{\alpha}$ as follows: The machine is in state q_0, for all $i = 1, \ldots, \ell$, string i is $\triangleright S(\alpha_i)$, and strings $\ell+1, \ldots, k$ are equal to $\triangleright$. Moreover, the heads are at the beginning of the strings.

If the machine halts, we associate to the k^{th} string the corresponding natural number (note that this string is of the form $\triangleright w$, where $w \in \{0,1\}^*$).

This association defines a partial function $f_T : \mathbb{N}_0^\ell \longrightarrow \mathbb{N}_0$, the function *computed by* T. As usual, we define the *running time* $\tau_T(\underline{\alpha})$ of T applied to $\underline{\alpha}$ to be ∞ if the machine does not halt (that is, $f_T(\underline{\alpha})$ is not defined), and otherwise to be the number of operations until it halts. If T applied on α halts, we define the *space requirements* or the *space complexity* $\text{Space}_T(\alpha)$ as the sum of the lengths of springs $\ell+1, \ldots, k$ at the time of halting. (Note that the strings never get shorter during the computation.)

We now give a formal definition of *bit-oriented Random Access Machines*.

Let $\mathbf{C}_{\text{basic}}$ be a set with 5 elements which we denote by LOAD, STORE, GOTO, IFGOTO, END; this set is called the set of *basic commands*.

Furthermore, let A be a set with 2 elements, which we denote by "=" and "↑".

Definition 1.32 A *basic instruction* is an element from the set $\mathbf{C}_{\text{bit}} \,\dot\cup\, \mathbf{C}_{\text{bit}} \times \mathbb{N}_0 \,\dot\cup\, \mathbf{C}_{\text{bit}} \times A \times \mathbb{N}_0$ which corresponds to a $*$ in the following table. Here and in the following we use the following notation: We write $X\ n$ (resp. $X\ m\,n$) for (X, n) (resp. (X, m, m)), and we call n (resp. (m, n) the *operand*. We denote the set of basic instructions by I_{basic}.

command	no operand	operand		
		$=n$	n	$\uparrow n$
LOAD	—	$*$	$*$	$*$
STORE	—	—	$*$	$*$
GOTO	—	—	$*$	—
IFGOTO	—	—	$*$	—
END	$*$	—	—	—

Example 1.33 LOAD 10 (that is, (LOAD, 10)) and STORE ↑ 5 (that is, (STORE, ↑, 5)), are instructions.

Let $\mathcal{T}$ be the set of all bit-oriented multiple string Turing machines with one output string. We fix a set $\mathbf{C}_{\text{arith}}$ which is in bijection to $\boldsymbol{\mathcal{T}}$ via a map $c : \mathcal{T} \longrightarrow \mathbf{C}_{\text{arith}}, T \mapsto c_T$. We call $\mathbf{C}_{\text{arith}}$ the set of *arithmetic commands*, and for $T \in \mathcal{T}$, we call c_T the *(arithmetic) command* for machine T.[4]

[4] We can of course set $\mathbf{C}_{\text{arith}} := \mathcal{T}$ and $c := \text{id}$, but it helps for the intuition to distinguish between a Turing machine and the related command.

Definition 1.34 The *instruction space* is $I_{\text{bit}} := I_{\text{basic}} \dot\cup \mathbf{C}_{\text{arith}}$. The elements of this space are called *instructions*, and the arithmetic commands are also called *arithmetic instructions*.

Definition 1.35 A *bit-oriented Random Access Machine (bit-RAM)* is a tuple $\Pi = (\pi_1, \ldots, \pi_r)$ of instructions.

We now define the semantics of the instructions, that is, we define how a bit-RAM operates.

We first define the *configuration space* as $S := \mathbb{N}_0^{(\mathbb{N}_0)} \times (\mathbb{N}_0 \dot\cup \{\infty\})$ (where $\mathbb{N}_0^{(\mathbb{N}_0)} := \{r \in \mathbb{N}_0^{\mathbb{N}_0} \mid \text{for nearly all } i, r(i) = 0\}$); the elements of this set are called *configurations*. If (r, k) is a configuration, r is called the *register configuration* and k is called the *counter*. Note that the configuration space does not depend on a particular bit-RAM.

Now we define the *configuration transition function* $\Delta : I_{\text{bit}} \times S \longrightarrow S$ as follows. Let (r, k) be a configuration, and let $\Delta(x, r, k) = (r', k')$.

Let x first be a basic instruction. Then $r'(i) = r(i)$ and $k' = k+1$ except for the cases indicated in the table.

command	no operand	operand		
		$= n$	n	$\uparrow n$
LOAD	—	$r'(0) = n$	$r'(0) = r(n)$	$r'(0) = r(r(n))$
STORE	—	—	$r'(n) = r(0)$	$r'(r(n)) = r(0)$
GOTO	—	—	$k' = n$	—
IFGOTO	—	—	$k' = n$ if $r(0) = 0$ and $k' = k + 1$ if $r(0) > 0$	—
END	$k' = 0$	—	—	—

Now let $x = c_T$ for some bit-oriented k-string Turing machine T with ℓ input strings and one output string. Then $r'(i) = r(i)$ for all $i \geq 1$. If T does not halt on $(r(1), \ldots, r(\ell))$, then $r'(0) = r(0)$ and $k' = \infty$. If it halts then $r'(0) = f_T(r(1), \ldots, r(\ell))$ and $k' = k + 1$.

We define the *operation* of Π on an *input* $\alpha \in \mathbb{N}_0$ as the following sequence $(r_t, k_t)_{t \in \mathbb{N}_0} \in S^{\mathbb{N}_0}$:

We define r_0 by $r_0(0) := \alpha$ and $r_0(i) := 0$ otherwise, and we set $k_0 := 1$. Then we define (r_t, k_t) for $t \geq 1$ by $(r_t, k_t) := \Delta(r_{t-1}, \pi_{k_{t-1}})$ if $k_{t-1} \in \{1, \ldots, r\}$ and by $(r_t, k_t) := (r_{t-1}, k_{t-1})$ otherwise.

If there exists some t with $k_t = 0$ or $k_t > r$ and $k_t \neq \infty$, we say that the machine *terminates*, and in this case, we define the *output* as $r_t(0)$ for such a t. (This definition is independent of t.) Note in particular that if there exists some t with $k_t = \infty$, then the machine does not terminate.

We define the *running time in RAM operations* or the *uniform running time* τ_u as ∞ if the machine does not terminate and otherwise as $\min\{t \in$

$\mathbb{N} \mid k_t \notin \{1, \ldots, r\}\}$. Note here that this measure does not say anything about the complexity of the operation because arbitrary complexity can be hidden in the application of one command c_T.

Finally, we define the *logarithmic running time* or the *running time in bit-operations* as follows: First we define a function $L := I_{\text{bit}} \times \mathbb{N}_0^{(\mathbb{N}_0)} \longrightarrow \mathbb{N}$, the *logarithmic cost measure*. Intuitively, this measure captures the complexities of the storage accesses and the operations of the Turing machines. More precisely, the definition is as follows. For some non-negative integer n, let $\ell(n)$, the *bit-length* of n, be defined as follows: $\ell(0) := 0$ and $\ell(n) := \lfloor \log_2(n) \rfloor + 1$ if $n \geq 1$. Then:

- – $L(\mathsf{LOAD}\ = n, r) := \ell(n)$,
  – $L(\mathsf{LOAD}\ n, r) := \ell(n) + \ell(r(n))$,
  – $L(\mathsf{LOAD}\ \uparrow n) := \ell(n) + \ell(r(n)) + \ell(r(r(n)))$.
  – $L(\mathsf{STORE}\ n, r) := \ell(r(0)) + \ell(n)$,
  – $L(\mathsf{STORE}\ \uparrow n, r) := \ell(r(0)) + \ell(r(n))$
- $L(\mathsf{GOTO}\ n, r) := 1$, $L(\mathsf{IFGOTO}\ n) := 1$, $L(\mathsf{END}\ ) := 1$
- $L(c_T, r) := \tau_T(r(1), \ldots, r(\ell))$, the running time of T applied to $(r(1), \ldots, r(\ell))$ if the Turing machine T has ℓ input strings.

Now, we are coming back to the operation of the RAM Π on the input α. We define the *logarithmic running time* or the *running time in bit-operations* of Π applied to α as ∞ if the machine does not terminate and otherwise as

$$\tau = \tau_\Pi(\alpha) := \sum_{t=0}^{\tau_u - 1} L(r_t, \pi_{k_t}) \,,$$

with the notations from above. In the following, whenever we speak about the running time or simply the time of the operation of a bit-RAM we mean the running time in bit-operations.

Moreover, if Π we define the *space requirements* or the *space complexity* $\text{Space}_\Pi(\alpha)$ of Π applied to α as follows: Let $(t_1, \ldots, t_r)$ with $t_1 < t_2 < \cdots < t_r$ be such that π_{t_i} is an instruction based on a Turing machine, and π_t is a basic instruction if $t \neq t_i$ for all i. Let $\pi_{t_i} = c_{T_i}$ for some Turing machine T_i with j_i input strings. Then

$$\text{Space}_\Pi(\alpha) := \sum_{i \in \mathbb{N}} \sup_{t \in \mathbb{N}_0} \ell(r_t(i)) + \sum_{i=1}^{r} \text{Space}_{T_i}(r(1), \ldots, r(j_i)) \,.^5$$

[5] Note that with this definition, the space complexity is at least as large as the input length. This means that this definition is not appropriate to study questions related to sub-linear space complexity classes.

1.4.3 Randomization

We persistently use the method of *randomization*. Informally this means that we introduce a new command RAND into the Random Access Machine which if executed returns a "random bit" which is stored in the accumulator r_0. Here the drawing is stochastically independent of all previous drawings. We call the resulting machine a *randomized bit-oriented Random Access Machine* (*randomized bit-RAM*).

We sketch the new ingredients for a possible rigorous approach to randomized bit-RAMs, starting with the bit-RAM model defined above.

First we enlarge the set of commands by a new element which we call RAND. We also enlarge the instruction space by RAND (that is, RAND has no operand).

Let Ω be a probability space with stochastically independent equidistributed probability variables $X_t : \Omega \longrightarrow \{0,1\}$ for $t \in \mathbb{N}$.

Let an input be fixed. Then similarly to the series $(r_t, k_t)_{t\in\mathbb{N}}$ defined above we now define a stochastic process $(R_t, K_t)_{t\in\mathbb{N}_0})$ with values in the configuration space S.

The process is determined by the randomized bit-RAM and the input similarly to above.

More precisely, let $\alpha \in \mathbb{N}$ be the input. Then we set $R_0(0) := \alpha, R_0(i) := 0$ and $K_0 := 1$ and define (R_t, K_t) for $t \geq 1$ as follows: If $K_{t-1} \notin \{1, \ldots, r\}$ then $(R_t, K_t) := (R_{t-1}, K_{t-1})$. So let $K_{t-1} \in \{1, \ldots, r\}$. If now $\pi_{K_{t-1}} \neq$ RAND then $(R_t, K_t) := \Delta(R_{t-1}, \pi_{K_{t-1}})$, just as above.

Now let $\pi_{K_{t-1}} =$ RAND. Then we set $K_t := K_{t-1}+1$ and $R_t(i) := R_{t-1}(i)$ for all $i \geq 1$, and we define $R_t(0) := X_t$.

We define the logarithmic cost for the command RAND as $L(\mathsf{RAND}, r, s) := 1$. The definitions of output, uniform running time and running time in bit operations are as above; these quantities are now also random variables on Ω. Note that the uniform running time and logarithmic running time are stopping times of the stochastic process $(R_t)_{t\in\mathbb{N}}$.

This completes the description of operation of a randomized bit-RAM. We fix the following two obvious remarks.

Remark 1.36 $(R_t, K_t)_{t\in\mathbb{N}_0}$ is a time-homogeneous Markov chain.

Remark 1.37 Let n be a natural number, let $r_1, \ldots, r_{\ell(n)}$ be independently uniformly distributed random variables with values in $\{0,1\}$, and let $N := \sum_{i=1}^{\ell(n)} r_i 2^{i-1}$. Then the probability that $N \leq n$ is $\geq \frac{1}{2}$. Moreover, conditionally to $N \leq n$, N is uniformly randomly distributed in $\{1, \ldots, n\}$. With these remarks one can easily obtain a randomized bit-RAM which upon input of some α outputs a uniformly randomly distributed random

variable with values in $\{1, \ldots, \alpha\}$ and which terminates in an expected time of $\mathcal{O}(\log(\alpha))$.

The following definition relates randomized bit-RAMs with computational problems as defined in Section 1.2.

Definition 1.38 Let $\boldsymbol{\mathcal{C}}$ be a large groupoid with bit-representation $(S, \mathcal{F})$, let $\boldsymbol{\mathcal{C}}'$ be a large groupoid with bit-representation $(S', \mathcal{F}')$, and let $\mathcal{P}$ be a computational problem from $\boldsymbol{\mathcal{C}}$ to $\boldsymbol{\mathcal{C}}'$. Now let Π be a randomized bit-RAM. Then we say that Π *computes* $\mathcal{P}$ if for every $x \in S$,

- Π applied to x terminates in finite expected time (that is, $\mathbb{E}(\tau_\Pi(x)) < \infty$)
- if with a probability > 0 Π applied to x terminates with some $y \in S'$, then we have $\mathcal{F}'(y) \in \mathcal{P}(\mathcal{F}(x))$.

An analogous definition holds for relative computational problems.

1.5 The use of the word "algorithm"

In Theorems 1 to 5 in the introduction to this work, we asserted the existence of "randomized algorithms" with certain properties. We should thus give a formal definition of "algorithm". As we stated above the theorems, "the underlying complexity model is always a randomized random access machine model with logarithmic cost function". This suggests that we define *a randomized algorithm* as nothing but a randomized bit-RAM and a deterministic algorithm as a bit-RAM.

Such a definition would however be at odds with the intuitive meaning of the world algorithm. Consider for example an "algorithm" in the intuitive meaning of the word which contains an instruction like "$c \longleftarrow ab$", where a and b are natural numbers. Then we would not obtain an "algorithm" in the rigorous sense because it is unclear *how* the two numbers should be multiplied.

It seems to us to be impossible to find a compromise between the usual informal use of the word algorithm and a formal definition. One possibility to resolve this problem would be to reserve the word "algorithm" only for informal descriptions and otherwise use words with formally defined meanings like "randomized bit-RAM".

We think however that it causes no harm if after these remarks we accept that there are two uses of the word algorithm: An informal use and a formal use. Whenever we make a claim that there exists an algorithm with certain properties, we have a formal use in mind. The specific computational model

is then clearly stated, and moreover, if we use randomization, we speak about a *randomized algorithm*.

If we use the word informally, we also have a specific computational model in mind, which might be randomized.[6] However, we always leave out certain details on the execution (as remarked above, even an innocently looking instruction like "$c \longleftarrow ab$" leaves out important information). We do however require that the various steps of the algorithm can be performed on the basis of the fixed computational model. We then often argue earlier or later that some steps in the algorithm can be performed in a certain time (resp. expected time) and maybe with certain space requirements, and this is all which is of interest to us. A priori, we also not demand that the algorithm terminates in a finite expected time.

In Chapter 3 we also talk about *procedures*. A procedure is similar to an algorithm, and in Chapter 3 the formal definition for *procedure* is also that of a randomized bit-RAM. However, there is a change of perspective: An algorithm takes an input and produces an output. A procedure on the other hand operates on the data structure. This is particularly important for the "relation generation procedure" for Theorem 1 in Section 3.3; in fact a "relation generation algorithm" would be extremely inefficient simply because then the tree of large prime relations would have to be read and stored again every time the "algorithm" is executed.

1.6 Algebraic complexity

In this section, we first briefly review some approaches to exact computations in rings in the literature, and then we describe a particular computational model which is based on bit-oriented Random Access Machines but allows exact computation over a fixed ring. We show how this model can also be applied for variable rings too, obtaining what we call *generic ring RAMs*. This model then allows a uniform formal description of algorithms over variable rings (in particular fields), and it will serve as a basis for the considerations in the next chapter.

1.6.1 Terminology

By a *ring* we always mean a ring with 1. Likewise, a homomorphism of rings is a homomorphism of rings with 1. Note that we do not demand that rings are always commutative.

[6] In contrast, in the literature, the word "algorithm" is often used without reference to a particular computational model.

Let now R be a commutative ring. By an *algebra* over R we mean a tuple (A, φ), where A is a ring and $\varphi : R \longrightarrow A$ is a homomorphism of rings whose image lies in the center of A. (As usual we omit φ if possible.) If (A, φ) and (B, ψ) are two R-algebras, then a *homomorphism of R-algebras* from A to B is a homomorphism of rings $\chi : A \longrightarrow B$ with $\chi \circ \varphi = \psi$.

1.6.2 Computation trees

One can use *computation trees* to study algorithmic problems in a fixed field (or in extension fields of a fixed field). This approach is for example taken in Bürgisser, Clausen, Shokrollahi: Algebraic Complexity Theory ([BCS91]). The model has however at least one shortcoming: It is not *uniform* for inputs of varying length. In fact, for different input lengths in principle completely different computations might be used. Another shortcoming is that the ground field is fixed (and in particular the characteristic is fixed). However if one wants to describe an algorithm which applies to varying fields, it is reasonable to demand that the formal description is also uniform over the fields.

We note that the phrase "algebraic complexity" is often used to denote the part of complexity theory which is based on straight-line programs and computation trees. We would however like to suggest to use this phrase for the part of complexity theory which is based on models which take a "macroscopic point of view" to computations in (possibly uncountable) rings. Straight-line programs and computation trees would then fall in the area of non-uniform algebraic complexity.

1.6.3 R-RAMs

A well-known model which features uniformity over a fixed ring R is described by Blum, Shub and Smale in [BSS89] (see also [BCSS98]). Roughly speaking this model can be seen as a Turing machine model over R.

This model is surely appropriate to study questions related to complexity classes such as if a particular problem can be computed in polynomial time. We are however seeking for a model which features (exact) arithmetic over a ring and otherwise is close to the RAM model; in particular we want to be able to (efficiently) use indirect addressing.

A model similar to the RAM model but for computations with real numbers is briefly described in [PS85]; it is called the *real RAM* model. This is a similar model than usual RAM model with the additional feature that (exact) storage and operation of real numbers is possible. This approach can easily be generalized to arbitrary rings R. We now give such a generalization; we call the resulting machines *R-RAMs* (R refers to the specific

ring).

From an intuitive point of view the starting point is a device similar to a bit-RAM, but now there are two kinds of registers (both indexed by natural numbers): The first kind of cells can store integers and the second kind can store elements of the ring R. Let us call the former registers r_i (as above) and the latter registers $s_i\,(i \in \mathbb{N}_0)$. Now additionally to r_0 the register s_0 serves as a second accumulator, and we have a second set of commands to operate on the registers s_i.

A formal description is as follows:

Let R be a ring. We take the definition of the Random Access Machine in subsection 1.4.2 as a starting point and only discuss the necessary modifications.

Additionally to the set $\mathbf{C}_{\text{bit}}$ we fix a second set of *commands* $\mathbf{C}_{\text{Ring}}$ with 7 elements which we denote by RINGLOAD, RINGSTORE, RINGADD, RINGSUB, RINGMULT, RINGDIV and RINGIFGOTO. We call these commands *ring oriented commands.*

A *ring oriented instruction* is an element from the set $\mathbf{C}_{\text{Ring}} \,\dot\cup\, \mathbf{C}_{\text{Ring}} \times R \,\dot\cup\, \mathbf{C}_{\text{Ring}} \times \{=\} \times \mathbb{N}_0 \,\dot\cup\, \mathbf{C}_{\text{Ring}} \times \{\uparrow\} \times \mathbb{N}_0$ which corresponds to a $*$ in the following table (where $a \in R$ and $n \in \mathbb{N}_0$). We denote the set of ring oriented instructions by $I_{R,\text{Ring}}$.

command	no operand	operand		
		$=a$	n	$\uparrow n$
RINGLOAD	—	$*$	$*$	$*$
RINGSTORE	—	—	$*$	$*$
RINGADD	$*$	—	—	—
RINGSUB	$*$	—	—	—
RINGMULT	$*$	—	—	—
RINGDIV	$*$	—	—	—
RINGIFGOTO	—	—	$*$	—

We define the *instruction space* as $I_R := I_{\text{bit}} \,\dot\cup\, I_{R,\text{Ring}}$, where I_{bit} is defined as for bit-RAMs.

Definition 1.39 An R-RAM is a tuple $\Pi = (\pi_1, \ldots, \pi_r)$ of instructions, that is, of elements from I_R.

We now define the *configuration space* as $S_R := \mathbb{N}_0^{(\mathbb{N}_0)} \times R^{(\mathbb{N}_0)} \times (\mathbb{N}_0 \,\dot\cup\, \{\infty\})$. If (r, s, k) is a *configuration*, that is, an element from S_R, we call r the *bit-register configuration*, s the *ring-register configuration* and k the *counter*.

We define the *configuration transition function* $\Delta : I_R \times S_R \longrightarrow S$ as follows: Let x be an instruction, let (x, r, s, k) be a configuration, and let

$T(x, r, s, k) = (r', s', k')$. Then if $x \in I_{\text{bit}}$, $s' = s$, and r' and k' are defined as in subsection 1.4.2. If x is a ring oriented instruction, then $r'(i) = r(i)$ for all $i \in \mathbb{N}_0$, $k' = k$, $s'(i) = s(i)$ for all $i \in \mathbb{N}$ and $s'(0)$ is defined as follows (with $a \in R, n \in \mathbb{N}_0$):

- – If $x = \mathsf{RINGLOAD}\ a : s'(0) = a$
  – If $x = \mathsf{RINGLOAD}\ n : s'(0) = s(n)$
  – If $x = \mathsf{RINGLOAD}\ \uparrow n : s'(0) = s(r(n))$
- – If $x = \mathsf{RINGSTORE}\ n : s'(n) = s(0)$
  – If $x = \mathsf{RINGSTORE}\ \uparrow n : s'(r(n)) = s(0)$
- If $x = \mathsf{RINGIFGOTO}\ n$: $k' = n$ if $s(0) = 0$ and $k' = k + 1$ otherwise
- If $x = \mathsf{RINGADD} : s'(0) = s(1) + s(2)$
- If $x = \mathsf{RINGSUB} : s'(0) = s(1) - s(2)$
- If $x = \mathsf{RINGMULT} : s'(0) = s(1) \cdot s(2)$
- If $x = \mathsf{RINGDIV} : s'(0) = s(1) \cdot s(2)^{-1}$ if $s'(2) \in R^*$ and $s'(0) = 0$ otherwise.

We define the set of *inputs* of R-RAMs as $\mathbb{N}_0 \times \dot{\bigcup}_{\ell \in \mathbb{N}_0} R^\ell$.

Now the operation of the RAM $\Pi = (\pi_1, \ldots, \pi_k)$ on the input $(\alpha, \underline{a}) = (\alpha; a_1, \ldots, a_\ell)$ is the following sequence $(r_t, s_t, k) \in S^{\mathbb{N}_0}$:

We define r_0 by $r_0(0) := \alpha$ and $r_0(i) := 0$ otherwise. We define s_0 by $s_0(i) := a_i$ for $i = 1, \ldots, \ell$ and $s_0(i) := 0$ for $i \geq \ell + 1$. We define $k_0 := 1$. Then for $t \geq 1$ we define (r_t, s_t, k_t) for $t \geq 1$ analogously to the definition in subsection 1.4.2: $(r_t, s_t, k_t) := \Delta(r_{t-1}, s_{t-1}, k_{t-1})$ if $k_{t-1} \in \{1, \ldots, r\}$ and by $(r_t, s_t, k_t) := (r_{k-1}, s_{t-1}, k_{t-1})$ otherwise.

The definitions of termination, output and running time in RAM operation are now just as in subsection 1.4.2.

Given an input such that Π terminates with a running time in RAM operations of τ_u, we define the *running time in ring operations with overhead* as $\tau_R := \#\{t \in \{0, \ldots, \tau_u - 1 \mid \text{ the command in } \pi_{k_t} \text{ is in } \mathbf{C}_{\text{Ring}}\}$. Moreover, we define the *running time in ring operations without overhead* as

$$\#\{t \in \{0, \ldots, \tau_u - 1 \mid \text{ the command in } \pi_{k_t} \text{ is in } \{\mathsf{RINGADD}, \mathsf{RINGSUB}, \mathsf{RINGMULT}, \mathsf{RINGDIV}\}\}.$$

We call the running time in ring operations *with* overhead also the *running time in ring operations*. Note however, usually in the literature the running time in ring operations without overhead is called running time in

ring operations. (This corresponds for example to the definitions for computation trees.) Sometimes, when we want to emphasize the underlying ring, we also speak about the *running time in ring operations in R*. If R is a field, we also speak about the *running time in field operations*.

Furthermore, we define the *logarithmic cost measure* $L : I_R \times \mathbb{N}_0^{(\mathbb{N}_0)} \times k^{(\mathbb{N}_0)} \longrightarrow \mathbb{N}$ similarly to the logarithmic cost measure in subsection 1.4.2: For instructions in I_{bit} the definitions are as in subsection 1.4.2, and for the instructions based on commands in $\mathbf{C}_{\text{Ring}}$, the definition is as follows: If X is a command, $n \in \mathbb{N}$ and $a \in k$, then

- $L(X = a, r, s) := 1$
- $L(X\, n, r, s) := \ell(n)$
- $L(X \uparrow n, r, s) := \ell(n) + \ell(r(n))$
- $L(X, r, s) := 1$

We define the *logarithmic running time* or the *running time in bit operations* as $\tau_{\text{bit}} := \sum_{t=0}^{\tau_u - 1} L(\pi_{k_t}, r_t, s_t)$. We call the sum $\tau_R + \tau_{\text{bit}}$ the *running time in ring and bit operations* or the *running time is ring operations in R and bit operations*.

Finally we define the *space requirements in ring elements* or the *space complexity in ring elements* as the number of registers which contain element $\neq 0$ at least once during the computation, that is, $\#\{i \in \mathbb{N}_0 \,|\, \exists t \in \mathbb{N}_0 : r_t(i) \neq 0\}$.

Remark 1.40 If R is a field with a total ordering $\leq$ (e.g., $R = \mathbb{R}$), one might introduce a further command which returns 1 or 0 according to the field element in s_1 being ≥ 0 or < 0.

Also, if R is a field of characteristic $p > 0$, one might introduce a command to compute p^{th}-th roots of unity. Indeed, we will introduce such a command below.

1.6.4 Generic field RAMs

Let now $\varphi : R \longrightarrow S$ be a ring homomorphism. Note that we have an obvious canonical map $I_R \longrightarrow I_S$. Via this map, we associate to every R-RAM Π an S-RAM $\varphi(\Pi)$.

Let now R be a commutative ring. If now A is an R-algebra with canonical morphism $\varphi : R \longrightarrow A$, we denote $\varphi(\Pi)$ also by $A(\Pi)$.

We call a tuple $(A, \alpha, \underline{a})$, where A is an R-algebra, $\alpha \in \mathbb{N}_0, \underline{a} \in R^n$ for some $n \geq 0$, an *input* over A. We define the *operation* of the R-RAM Π on such a tuple $(A, \alpha, \underline{a})$ as the operation of $A(\Pi)$ on $(\alpha, \underline{a})$.

With this definition, we can also define the uniform running time, the running time in ring operations (in A) (with or without overhead) and the running time in bit operation of Π applied to such a tuple $(A, \alpha, \underline{a})$.

Note the following informal description of the operation of Π on $(A, \alpha, \underline{a})$: At the beginning of the computation, the "algebraic registers" s_i are instantiated to contain elements from A and the ring-oriented commands are likewise instantiated to perform operations with such elements, and then the input is stored and the computation is performed.

If we want to stress that we allow inputs over arbitrary R-algebras rather then merely over R itself, we call an R-RAM also an *R-algebra* RAM. Note however that by definition an R-RAM just as an R-algebra RAM is merely a tuple of elements from I_R; it is the operation which is extended to R-algebras.

We call a $\mathbb{Z}$-algebra RAM a *ring RAM*. Note that this RAM operates on inputs over arbitrary rings. To emphasize this, we also speak of a *generic ring RAM*. If we now restrict the possible inputs to triples $(k, a, \underline{a})$, where k is a field, we obtain what we call a *(generic) field RAM*.

For Chapter 2 the following variant of the generic field RAM model is particularly relevant: We enrich the generic field RAM model with a command to compute p^{th} roots in characteristic $p > 0$. The additional command, called FIELDCHARROOT, takes no operand and operates as follows: If the characteristic of the field k is 0, then 0 is written in register s_0. If the characteristic of the field k is $p > 0$, then if $s(1)^{\frac{1}{p}} \in k$, this is written in s_0. Otherwise, again 0 is written. We charge one field and one bit operation for each application of the command. The resulting machines will serve as a formal basis for deterministic algorithms in the "algebraic setting".

We remark that it would be reasonable to enrich generic field RAMs also with a command which returns the characteristic of the field. The existence of such a command would however not change the complexities obtained in the next chapter.[7]

Analogously to randomized bit-RAMs we also define *randomized generic field RAMs*. For this we enrich the generic field RAM model with two further commands: RAND and FIELDFACTOR. (We could also include the command FIELDCHARROOT but this would not change our results.) The formal definition of the operation of the command RAND is analogous to the one in subsection 1.4.3.

The command FIELDFACTOR (which does not take an operand) factors a

[7]Sometimes – for example in the algorithm we give for Proposition 1.50 – we have to check if the characteristic is positive and below a certain bound, and if so, we have to determine it. However, such bounds are always so small that this computation can efficiently be performed by brute force.

univariate polynomial, and in doing so it mimics the behavior of randomized (Las Vegas) algorithms: With a probability of $\frac{1}{2}$ it fails and with a probability of $\frac{1}{2}$ it returns the list of normalized irreducible factors (including multiplicities).

It follows a more detailed but still informal description of the operation of this command. A formal description can be given analogously to the description of the command RAND given in subsection 1.4.3.

We first fix how univariate polynomials are stored. The idea is to use the *dense* representation. The concrete representation is of course somewhat arbitrary, but let us for concreteness say that a polynomial $\sum_{i=0}^{d} a_i t^i$ with $a_d \neq 0$ is stored as an array $(a_0, 0, a_1, \ldots, a_{d-1}, 0, a_d, 1)$ which starts at at a certain register s_i and ends in register s_{i+2d-1}.

Now the command operates as follows: It tries to interpret the contents of the registers starting at s_0 as a polynomial. If this is not the case, the command terminates with writing 0 in r_0. If it recognizes a polynomial, it operates as follows: With a probability of $\frac{1}{2}$ the command terminates with writing 0 in register r_0. (The idea is that the factorization fails.) With a probability of $\frac{1}{2}$ it terminates with the normalized irreducible factors (without multiplicities), starting with the first coefficient of the first element in s_0. Also, starting in r_0, the corresponding list of multiplicities is written. The ordering of the factors is chosen uniformly at random from all possible orderings. Moreover, the execution of the command is stochastically independent of all previous and later uses of randomization in the algorithm.

We are now coming to the complexity of the two commands RAND and FIELDFACTOR. To define the bit-complexity, we again extend the logarithmic cost measure L to include the two commands. Analogously to above, we define $L(\mathsf{RAND}, r, s) := 1$, and we define $L(\mathsf{FIELDFACTOR}, r, s) := 1$. Then we define the running time in bit-operations analogously to above.

Concerning the running running time in field operations (with or without overhead), we proceed as follows: We do not take into account an occurrence of the command RAND. Moreover, whenever the command FIELDFACTOR is applied we charge 1 field operation if the content of the registers starting with s_0 cannot be interpreted as a univariate polynomial, and otherwise, if the polynomial has degree d, we charge d^2 field operations. Note that this means that with the command FIELDFACTOR a polynomial of degree d can be factored with an expected number of $\mathcal{O}(d^2)$ field operations. We note that we choose to charge d^2 field operations because a polynomial of degree d over a fixed finite field can be factored with an expected number of $o(d^2)$ field operations with the Kaltofen-Shoup algorithm; see [KS98]. See also subsection 1.6.6 for further discussion.

1.6.5 Algebraic computational problems

We come to the changes in our definitions for computational problems which have to be performed for the change from the bit-oriented to the "algebraic" computational model.

If one fixes a ground field, the modifications are indeed quite minor, but we would like to vary the field as well.

Let $\boldsymbol{Fields}$ be the large groupoid obtained from the category of fields by discarding all morphisms which are not isomorphisms. We now only consider large groupoids over $\boldsymbol{Fields}$, that is, we consider tuples $(\mathcal{C}, \mathcal{C} \longrightarrow \boldsymbol{Fields})$, where $\mathcal{C}$ is a large groupoid and $\mathcal{C} \longrightarrow \boldsymbol{Fields}$ is a functor. Also the functors we consider are over $\boldsymbol{Fields}$, that is, they are compatible with the functors to fields.

Apart from this "relative approach" we use the terminology on (partial) representations and relative (partial) representations from subsection 1.2.

Remark 1.41 A situation one encounters often is as follows: The fibers of $\mathcal{X}$ over $\boldsymbol{Fields}$ are sets, and moreover: For $x \in \mathcal{X}$ lying over a field k and an isomorphism $k \longrightarrow k'$ there exists exactly one $x' \in \mathcal{X}$ such that there exists some isomorphism $x \longrightarrow x'$ over the isomorphism $k \longrightarrow k'$.

Example 1.42 A large groupoid which is often used to represent other large groupoids is the large groupoid of vectors of arbitrary lengths over arbitrary fields, defined as follows: The objects are tuples $(k, \underline{a})$, where k is a field and $\underline{a} \in k^n$ for some $n \in \mathbb{N}_0$. The morphisms are defined as follows: Let k and k' be fields, and let $\underline{a} \in k^n, \underline{a}' \in (k')^m$. If $m \neq n$, there are no morphisms. If $n = m$, we define $\mathrm{Mor}((k, \underline{a}), (k', \underline{a}')) := \{\varphi : \mathrm{Iso}(k, k') \mid \varphi(a_i) = a'_i \text{ for all } i = 1, \ldots, n\}$.

Now we have to substitute the bit representations by representations which are appropriate for a generic field RAM.

For this we define a large groupoid $\boldsymbol{RAMinputs}$: The objects are the inputs over arbitrary fields already considered in the definition of generic field RAMs, that is, they are triples $(k, \alpha, \underline{a})$, where $\alpha \in \mathbb{N}_0$ and $\underline{a} \in k^n$ for some $n \in \mathbb{N}_0$. The morphisms are defined as follows: Let $(k, \alpha, \underline{a})$, $(k', \alpha, \underline{a})$ be two objects with $\underline{a} \in k^n$ and $\underline{a}' \in k^m$. Then if $\alpha \neq \alpha'$, there are no morphisms from $(k, \alpha, \underline{a})$ to $(k', \alpha, \underline{a}'))$. For $\alpha = \alpha'$, we define $\mathrm{Mor}((k, \alpha, \underline{a}), (k', \alpha', \underline{a})) := \mathrm{Mor}((k, \underline{a}), (k', \underline{a}'))$.

Definition 1.43 A *generic field RAM representation* of a large groupoid $\mathcal{C}$ is a representation of $\mathcal{C}$ by $\boldsymbol{RAMinputs}$. Similarly, a *relative generic field RAM representation* of a functor $\mathcal{C} \longrightarrow \mathcal{C}'$ between large groupoids is a representation of $\mathcal{C} \longrightarrow \mathcal{C}'$ by $\boldsymbol{RAMinputs} \longrightarrow \boldsymbol{RAMinputs}$.

Note again that we consider throughout large groupoids and functors over $\boldsymbol{Fields}$; this applies in particular the definition just made.

A *field oriented computational problem* is defined as a computational problem in the bit-oriented models (see Definition 1.17) with the differences that we substitute $\mathbb{N}_0$ by $\boldsymbol{RAMinputs}$ (and all categories and functors are over $\boldsymbol{Fields}$). Analogously *relative field oriented computational problems* are defined as in Definition 1.25 with these modifications.

Let a computational problem or a relative computational problem be given, and let Π be a generic field RAM enriched with the command FIELDCHARROOT or a randomized generic field RAM. Then we use the phrase "Π computes the problem" analogously to Definitions 1.17, 1.25 and 1.38.

1.6.6 Algebraic complexity and bit-complexity over finite fields

If one considers computational problems over finite fields, any complexity theoretic statement on the basis of a generic field RAM immediately leads to a corresponding statement on the basis of a bit-RAM, as described now:

Let $\boldsymbol{Fields_{fin}}$ be the large groupoid of finite fields. We fix a bit-representation of $\boldsymbol{Fields_{fin}}$ along the following lines: We first represent each finite field by an irreducible polynomial over the prime field, and we represent the irreducible polynomial by its coefficient vector. Then we represent the coefficient vector by a natural number. (There is of course some ambiguity in this description but clearly, there are "reasonable" possibilities to fill in the gaps.)

Now we fix a relative representation of the large groupoid of tuples (k, a) where k is a finite field and $a \in k$ over $\boldsymbol{Fields_{fin}}$ by representing the element $a \in k$ by its coefficient vector with respect to a polynomial basis; cf. the discussion above Definition 1.10. We extend this representation in an obvious way to represent the large groupoid of tuples $(k, \underline{a})$, where $\underline{a} \in k^n$ for some $n \in \mathbb{N}$ and then also to $\boldsymbol{RAMinputs}$.

It is obvious that with this representation the field operations of a generic field RAM restricted to instances over finite fields $\mathbb{F}_q$ can be performed with $\mathcal{O}(\log^2(q))$ bit-operations. In fact, this running time can be lowered to $\mathcal{O}(\log(q) \cdot (\log\log(q))^2 \cdot \log\log\log(q))$ by using FFT for multiplication and additionally a fast Euclidian algorithm for division; see [vzGG03, Corollary 11.10].

With this result, one can associate to any generic field RAM a bit-RAM which performs the same computations over finite fields. Roughly, the bit-RAM is defined by using the even numbered registers for the previous bit-registers r_i and the odd-numbered registers for the previous "algebraic"

registers. Along these lines one obtains the following proposition.

Proposition 1.44 *Let Π be a generic field RAM. Then there exists a bit-RAM Π_{bit} such that the following holds: Let $(\mathbb{F}_q, \alpha, \underline{a})$ be an input over a finite field, and let y represent this input. Then Π applied to $(\mathbb{F}_q, \alpha, \underline{a})$ terminates if and only Π_{bit} applied to y terminates.*

Let us assume that Π terminates, and let τ_f be the running time in field operations with overhead and τ_{bit} the running time in bit operations. Then Π_{bit} terminates in $\mathcal{O}(\tau_f \cdot \log(q) \cdot (\log\log(q))^2 \cdot \log\log\log(q) + \tau_{\text{bit}})$ bit operations.

Remark 1.45 Let $q = p^n$ for a prime p. Then in the finite field $\mathbb{F}_{p^n}$, we have $a^{-p} = a^{p^{n-1}}$. Therefore the computation of p^{th} roots in $\mathbb{F}_{p^n}$ can be performed with $\mathcal{O}(\log(q))$ field multiplications in $\mathbb{F}_{p^n}$. On a bit-RAM this is possible with $\mathcal{O}(\log(q)^2 \cdot \log\log(q) \cdot \log\log\log(q))$ bit operations.

Therefore, for every generic field RAM enriched with the command FIELDCHARROOT Π, there exists a bit-RAM Π_{bit} as in the proposition except that if Π_{bit} terminates, then it terminates in $\mathcal{O}(\tau_f \cdot \log(q)^2 \cdot \log\log(q) \cdot \log\log\log(q) + \tau_{\text{bit}})$ bit operations.

With the Kaltofen-Shoup algorithm ([KS98]) and FFT-based multiplication one needs $\mathcal{O}((d\log(q))^2)$ bit-operations to factorize a polynomial of degree d over a finite field $\mathbb{F}_q$. (The given bound is not best-possible.) As we have defined the running time in field operations of one application of the command FIELDFACTOR to a polynomial of degree d to be d^2 field operations, this gives rise to the following proposition.

Proposition 1.46 *Let Π be a generic field RAM with randomization. Then there exists a bit-RAM Π_{bit} such that the following holds:*

Let $(\mathbb{F}_q, \alpha, \underline{a})$ be an input over a finite field, and let y represent this input. Then Π applied to $(\mathbb{F}_q, \alpha, \underline{a})$ terminates in a finite expected number of bit operations if and only Π_{bit} applied to y terminates in a finite expected number of bit operations.

Let us assume that Π terminates in a finite expected number of bit operations, and let $\mathbb{E}(\tau_f)$ be the expected running time in field operations with overhead and $\mathbb{E}(\tau_{\text{bit}})$ the expected number of bit operations. Then Π_{bit} terminates in an expected number of $\mathcal{O}(\mathbb{E}(\tau_f) \cdot \log^2(q) + \mathbb{E}(\tau_{\text{bit}}))$ bit operations.

1.7 Algebraic complexity and finite algebras

This section focuses on basic computational problems related to finite commutative algebras over fields. As an application we consider basic questions

related to field oriented computational problems and finite field extensions. The model of computation is always given by generic field RAMs enriched with the command FIELDCHARROOT or by randomized generic field RAMs (which means in particular that the command FIELDFACTOR is available).

It is a basic observation that given any finite algebra A, represented by a multiplication table over k, one can perform (deterministically) the arithmetic operations in A with $\mathcal{O}(\dim_k(A))$ field operations (in k). We start off by formalizing this result with the terminology of the previous sections.

We then recall results from the literature which say that if k is perfect, one can determine the primary decomposition as well as the prime ideals of a commutative k-algebra A with an expected number of field operations which is polynomially bounded in $\dim_k(A)$. As an application we show that given a finite field extension $\lambda|k$, again represented by a multiplication table, and a polynomial $f(t) \in \lambda[t]$, one can factorize this polynomial over λ with an expected number of field operations which is polynomially bounded in $[\lambda : k]$ and $\deg(f)$.

As an application of these results we study the following problem: Let a field oriented computational problem $\mathcal{P} : \boldsymbol{\mathcal{C}} \longrightarrow \boldsymbol{\mathcal{C}}'$ be given. Then one can consider the problem which is intuitively described as follows: Given a finite field extension $\lambda|k$, represented by a multiplication table, and some object x of $\boldsymbol{\mathcal{C}}$ lying over λ, compute $\mathcal{P}(x)$ with a generic field RAM instantiated with k (and not λ)! It is easy to turn every deterministic algorithm to compute $\mathcal{P}$ into an algorithm to compute this "extended problem". With the result on factorization of polynomials the same holds for randomized algorithms if one restricts oneself to computational problems over perfect fields.

1.7.1 Computing in finite algebras

In this subsection, we discuss the easy statement that given any finite k-algebra A, represented by a multiplication table, one can perform the arithmetic operations in A with $\mathcal{O}(\dim_k(A))$ field operations in k. In doing so we make the effort to in detail describe how one should define the large groupoids, functors etc. for an application of the notions of the previous sections. Just as in subsection 1.6.6, all large groupoids and functors are over the large groupoid $\boldsymbol{\mathcal{F}ields}$ of fields.

We start off with the large groupoid $\mathcal{ALG}$ whose objects and (iso-)morphisms are defined as follows: The objects are finite algebras over fields. The (iso-)morphisms are pairs of isomorphisms which fit into the following

diagram.

$$\begin{array}{ccc} A & \longrightarrow & A' \\ \uparrow & & \uparrow \\ k & \longrightarrow & k' \end{array}$$

Note that $\mathcal{A}LG$ is a large groupoid over the large groupoid $\boldsymbol{\mathcal{F}ields}$ of fields.

For some field k and $n \in \mathbb{N}$ we define $k^{n\times n\times n} := k^{\{1,\dots,n\}^3}$, and we call the elements of this set *multiplication tables*.

If $M = (m_{i,j,k})_{i,j,k} \in k^{n\times n\times n}$ is such a multiplication table, we define an associated operation $\cdot_M$ on k^n as follows: The operation is the unique k-bilinear operation with $\underline{e}_i \cdot_M \underline{e}_j = (m_{i,j,1}, \dots, m_{i,j,n})$, where $\underline{e}_i$ is the i-th standard vector.

Note that if $B(k,n)$ is the set of k-bilinear operations on k^n, we have a bijection $k^{n\times n\times n} \longrightarrow B(k,n)\,, M \mapsto \cdot_M$.

Now let $\boldsymbol{\mathcal{M}}$ be the large groupoid of multiplication tables over arbitrary fields. The definition is completely analogous to the definition in Example 1.42: The objects are tuples (k, M), where k is a field and M a multiplication table with entries in k. Let (k, M) and (k', M') be two objects. Then if the multiplication tables have different size, there are no morphisms between the objects. If on the other hand both have size n, the set of (iso-)morphisms from (k, M) and (k', M') is $\{\varphi \in \mathrm{Iso}(k, k') \mid m'_{i,j,k} = \varphi(m_{i,j,k}) \text{ for all } i, j, k = 1, \dots, n\}$.

We now want to define an $\boldsymbol{\mathcal{M}}$-representation of $\mathcal{A}LG$ over k which makes the statement "we represent finite k-algebras by multiplication tables" precise.

For this, we first define the full subcategory $\boldsymbol{\mathcal{S}}$ of $\boldsymbol{\mathcal{M}}$ which consists of all multiplication tables which define algebras. Then the representation is given by the functor $\mathcal{F} : \boldsymbol{\mathcal{S}} \longrightarrow \mathcal{A}LG$ which is given as follows: If $M \in k^{n\times n\times n}$, then $\mathcal{F}((k, M)) = (k^n, \cdot_M)$. If $\varphi : (k, M) \longrightarrow (k', M')$ is a morphism with $M \in k^{n\times n\times n}$ and $M' \in (k')^{n\times n\times n}$, then in particular $\varphi : k \longrightarrow k'$. This induces a map $k^n \longrightarrow (k')^n$. Now by definition of the morphisms in $\boldsymbol{\mathcal{M}}$ this map is a morphism of fields and we have the commutative diagram

$$\begin{array}{ccc} (k^n, \cdot_M) & \longrightarrow & ((k')^n, \cdot_{M'}) \\ \uparrow & & \uparrow \\ k & \xrightarrow{\varphi} & k' \,. \end{array}$$

We define $\mathcal{F}(\varphi)$ to be the morphism in the upper row. Obviously we have defined an essentially surjective functor, that is, a representation of $\boldsymbol{\mathcal{A}LG}$ by $\boldsymbol{\mathcal{M}}$.

This representation induces a relative representation of the large groupoid of tuples (A, a) with $a \in A$ (and the obvious isomorphisms) over $\boldsymbol{\mathcal{A}LG}$.

We now fix a representation of $\boldsymbol{\mathcal{M}}$ by $\boldsymbol{\mathcal{RAM}inputs}$. (As often, there is some ambiguity here but we assume that the representation is in an obvious sense "reasonable".) We obtain also a relative representation of the large groupoid of tuples (A, a) by $\boldsymbol{\mathcal{RAM}inputs}$. (With the obvious isomorphisms.) (Again there is some ambiguity here.) Analogously, we obtain a relative representation of the large groupoid of tuples $((k, A), (a, b))$ with k a field, A a finite separable k-algebra and $(a, b) \in A^2$ by $\boldsymbol{\mathcal{RAM}inputs}$.

We now have the following obvious proposition.

Proposition 1.47 *There exists a generic field RAM* Π *which computes* $\mathcal{P}$ *such that the following holds: If* r *represents* $((k, A), (a, b))$*, then the number of field operations performed by* Π *if applied to* r *is in* $\mathcal{O}(\dim_k(A)^3)$*, and the number of bit operations is polynomially bounded in* $\dim_k(A)$*.*

Remark 1.48 More intuitively, the previous proposition can be stated as follows:

Given a field k *and a finite* k*-algebra* A*, represented by a multiplication table, as well as* $a, b \in A$*, one can compute* $a \cdot b \in A$ *in a number of field operations in* k *which is in* $\mathcal{O}(\dim_k(A)^3)$ *and a number of bit operations which is polynomially bounded in* $\dim_k(A)$*.*

In the following we mostly use an intuitive formulation along the lines of this formulation.

1.7.2 Factoring finite commutative algebras and polynomials

We consider the problem to determine the decomposition into local rings as well as the prime ideals of a finite commutative algebra over a field. Throughout we assume that the algebra is given by a multiplication table with respect to a k-basis. A nice overview over algorithms for this problem is given in [KM04a, Section 7].

We do not anymore make the effort to formalize the description with the notions of the previous sections.

The following result is proven in [KM04a, Section 7].

Proposition 1.49 *Given a finite commutative* k*-algebra* A*, one can with a randomized algorithm determine the canonical decomposition* $A = A_1 \oplus \cdots \oplus A_\ell$ *of* A *into local rings in an expected number of field and bit operations which is polynomially bounded in* $\dim_k(A)$*. (Here the* A_i *are represented via coordinate vectors of* k*-bases with respect to the chosen basis of* A*.)*

Note again that we assume that the algebra A is given by a multiplication table with respect to a k-basis. Note also that the A_i are not subalgebras

of A in our terminology as the unity of A_i does not coincide with the unity of A.

The statement in the following proposition is proven in [KM04a] under the additional assumption that the algebra A is local. Because of this, we state it here with a proof.

Proposition 1.50 *Given a finite commutative k-algebra A over a perfect field k, one can with a deterministic algorithm compute the nilradical of A in a number of field operations which is polynomially bounded in* $\dim_k(A)$. *(Again the nilradical is represented via coordinate vectors of a k-basis.)*

Let $d := \dim_k(A)$ and $p := \mathrm{char}(k)$. Let $\mathcal{N}$ be the nilradical. Our proof of this proposition is based on the following two lemmata.

Lemma 1.51 *Let either the characteristic of k be 0 or larger than d. Then some $a \in A$ is nilpotent if and only if* $\mathrm{Tr}(ab) = 0$ *for all $b \in A$. With other words, $\mathcal{N}$ is the kernel of the k-linear map*

$$A \longrightarrow \mathrm{Hom}_k(A, k)\,,\ a \mapsto (b \mapsto \mathrm{Tr}(ab))\,.$$

Proof. Let $a \in \mathcal{N}$. Then $a^i = 0$ for some $i \in \mathbb{N}$. Now let $b \in A$. Then $(ab)^i = 0$ too. Therefore the minimal polynomial of ab is $t^i \in k[t]$ for some $i \leq d$. In particular the characteristic polynomial of a is t^d. Therefore, the trace of ab is 0.

For the converse let us first assume that A is a local algebra. Now let $a \notin \mathcal{N}$. Then a is a unit and there exists some $b \in A$ with $ab = 1$. Therefore $\mathrm{Tr}(ab) = d \neq 0$.

Let now A be arbitrary, and let $A = A_1 \oplus \cdots \oplus A_t$ be the decomposition into local algebras, and let $\mathfrak{m}_i$ be the maximal ideal of A_i. Then $\mathcal{N} = \bigcap_{i=1}^{\ell} \mathfrak{m}_i$. Let $a \notin \mathcal{N}$, $a = (a_1, \ldots, a_\ell)$. Then there exists some $j = 1, \ldots, t$ such that $a_j \notin \mathfrak{m}_j$. Therefore there exists a $b \in A$ with $ab = (\delta_{i,j})_{i,j}$. Now $\mathrm{Tr}_{A/k}(ab) = \mathrm{Tr}_{A_j/k}(1) \neq 0$. □

The beginning of the proof also implies:

Lemma 1.52 $\mathcal{N} = \{a \in A \mid a^d = 0\}$

The algorithm for Proposition 1.50 is now as follows:

First we check by "brute force" if either ($p = 0$ or $p > d$) or ($p > 0$ and $p \leq d$). In the first case, we compute the nilradical following Lemma 1.51.

So let us now assume that $p \geq 0$ and $p \leq d$. Let $j \in \mathbb{N}$ be such that $p^j \geq d$. Let $b_1, \ldots, b_d$ be the fixed basis of A. For $a \in A$, let $\underline{a} \in k^d$ be the associated coordinate vector. Let $M \in k^{d \times d}$ be the matrix which defines the k-linear map given by $b_i \mapsto b_i^p$.

Let $\sigma : k^d \longrightarrow k^d, (a_1, \ldots, a_d) \mapsto (a_1^p, \ldots, a_d^p)$. Let $\Lambda_M : k^d \longrightarrow k^d, \underline{a} \mapsto M\underline{a}$. Note that for $a \in A$, $M\sigma\underline{a} = (\Lambda_M \circ \sigma)(\underline{a})$ is the coordinate vector of a^p. More generally, for $i \in \mathbb{N}$ $(\Lambda_M \circ \sigma)^i(\underline{a})$ is the coordinate vector of a^{p^i}.

By the previous lemma, an element $a \in A$ is in $\mathcal{N}$ if and only if $a^{p^j} = 0$, and this is the case if and only if $(\Lambda_M \circ \sigma)^j(\underline{a}) = 0$.

Let for $N \in k^{d\times d}$ $\sigma(N)$ be the matrix obtained by raising all entries of N to the p^{th} power. Then $\sigma \circ \Lambda_M = \Lambda_{\sigma(M)} \circ \sigma$, and therefore, $(\Lambda_M \circ \sigma)^j = \Lambda_{\sigma^{j-1}(M)\sigma^{j-2}(M)\cdots\sigma(M)M} \circ \sigma^j$.

Now let $\underline{g}_1, \ldots, \underline{g}_t$ be a basis of $\ker(\Lambda_{\sigma^{j-1}(M)\sigma^{j-2}(M)\cdots\sigma(M)M})$. Then as k is perfect, $\sigma^{-j}(\underline{g}_1), \ldots, \sigma^{-j}(\underline{g}_t)$ are the coordinate vectors of a basis of $\mathcal{N}$.

The computation is now obvious. □

Remark 1.53 Note that by the previous proposition, one can determine the maximal ideal of a local finite commutative k-algebra A in a number of field operations which is polynomially bounded in $\dim_k(A)$. This applies of course in particular to the components A_i in Proposition 1.49.

Remark 1.54 Let $\mathfrak{m}$ be the maximal ideal of A, let $\lambda := A/\mathfrak{m}$, and let $a_1, \ldots, a_{\dim_k(\mathfrak{m})}$ be a k-basis of $\mathfrak{m}$. Let $a_1, \ldots, a_{\dim_k(A)}$ be an extension to a k-basis of A. Then $a_{\dim_k(\mathfrak{m})+1}, \ldots, a_{\dim_k(A)}$ induces a k-basis of λ.

From the previous proposition it follows that one can compute a k-basis of λ and the corresponding multiplication matrix of λ in a number of field operations which is polynomially bounded in $\dim_k(A)$.

The previous results imply:

Proposition 1.55 *Given a finite field extension $\lambda|k$ over a perfect field k, represented by a multiplication table over k, as well as a polynomial $f(t) \in \lambda[t]$ of the form $f(t) = g(t)^e$ for a monic irreducible polynomial $g(t) \in \lambda[t]$ and $e \in \mathbb{N}$, one can with a deterministic algorithm compute $g(t)$ and e in a number of field operations which is polynomially bounded in $[\lambda : k]$ and* $\deg(f)$.

In particular, if $\mathrm{char}(k) = p > 0$*, then one can compute p^{th} roots in finite extension fields $\lambda|k$ in a number of field operations which is polynomially bounded in $[\lambda : k]$ and* $\deg(f)$.

Proof. Let $A := \lambda[t]/(f(t))$. Then A is a local commutative k-algebra with maximal ideal $\mathfrak{m} = (g(t))$.

As a λ-vector space A has the basis given by the residue classes $[1]_{(f)}, \ldots, [t^{\deg(f)-1}]_{(f)}$. We fix the ordering $[1]_{(f)} < \cdots < [t^{\deg(f)-1}]_{(f)}$ on the basis elements. Then $[g(t)]_{(f)}$, the residue class of $g(t)$, is the unique element of $\mathfrak{m}$ whose coefficient vector with respect to $[1]_{(f)}, \ldots, [t^{\deg(f)-1}]_{(f)}$ has the least initial term and whose leading coefficient is 1.

So to determine $g(t)$, we proceed as follows: We compute a k-basis of $\mathfrak{m}$ (see Proposition 1.50). This basis also gives a generating system of $\mathfrak{m}$ over λ. From this generating system we determine $g(t)$ with a Gaussian elimination. □

Proposition 1.56 *Given a finite field extension $\lambda|k$ over a perfect field k, represented by a multiplication table over k, as well as a polynomial $f(t) \in \lambda[t]$, one can with a randomized algorithm determine the factorization of $f(t)$ (over λ) in an expected number of field and bit operations in k which is polynomially bounded in $[\lambda : k]$ and $\deg(f)$.*

Proof. Let $f(t) = c \cdot f_1^{e_1} \cdots f_\ell^{e_\ell}$ with $c \in \lambda$, distinct monic and irreducible polynomials $f_i \in \lambda[t]$ and $e_i \in \mathbb{N}$. We consider the finite k-algebra $\lambda[t]/(f(t))$. We have the decomposition $\lambda[t]/(f(t)) \simeq \lambda[t]/((f_1)^{e_1}) \oplus \cdots \oplus \lambda[t]/((f_k)^{e_\ell})$. Let φ_i be the projection to the i^{th} factor. Then by the decomposition we have $[t]_{(f)} = \varphi_1([t]) + \cdots + \varphi_\ell([t])$. Moreover, the characteristic polynomial of $\varphi_i([t])$ is $f_i^{e_i}$.

We first compute the decomposition, say $\lambda[t]/(f(t)) = A_1 \oplus \cdots \oplus A_\ell$. We compute the multiplication tables of the A_i (with respect to the computed bases) and the decomposition of $[t]_{(f)}$, say $[t]_{(f)} = t_1 + \cdots + t_\ell$. For the first task see Proposition 1.49, the second task is an easy linear algebra computation.

Now we compute the characteristic polynomials of the t_i's in the factors A_i. This gives the polynomials $f_i^{e_i}$. Now the f_i's and the e_i's can be computed as claimed by the previous proposition. □

1.7.3 Computational problems and finite field extensions

Let $\boldsymbol{\mathcal{C}}$ be a large groupoid, as usual over the large groupoid of fields, and let $p : \boldsymbol{\mathcal{C}} \longrightarrow \boldsymbol{\mathcal{F}ields}$ be the structural functor. Now we consider the following large groupoid $\boldsymbol{\mathcal{C}}_{\text{ext}}$:

The objects are tuples $(x, \lambda|k)$, where $\lambda|k$ is a finite field extension and x is an object of $\boldsymbol{\mathcal{C}}$ over λ. Let $(x, \lambda|k)$ and $(x', \lambda'|k')$ be two such objects. Then $\text{Mor}((x, \lambda|k), (x', \lambda'|k)) := \{\varphi \in \text{Mor}(x, x') \mid p(\varphi) \in \text{Iso}(k, k')\}$.

Let $\boldsymbol{\mathcal{F}E}$ be the large groupoid of finite field extensions. Note that this is a full subcategory of $\boldsymbol{\mathcal{A}LG}$. Just as $\boldsymbol{\mathcal{A}LG}$ we represent $\boldsymbol{\mathcal{F}E}$ by $\boldsymbol{\mathcal{M}}$, the large groupoid of multiplication tables. Note that we have already fixed a representation of $\boldsymbol{\mathcal{M}}$ by $\boldsymbol{\mathcal{R}AMinputs}$, thus we have a representation of $\boldsymbol{\mathcal{F}E}$ by $\boldsymbol{\mathcal{R}AMinputs}$ as well.

We have the functor $\boldsymbol{\mathcal{C}}_{\text{ext}} \longrightarrow \boldsymbol{\mathcal{F}E}$ defined by $(a, \lambda|k) \mapsto \lambda|k$ (and the obvious maps for morphisms). By composing this with the functor $\boldsymbol{\mathcal{F}E} \longrightarrow \boldsymbol{\mathcal{F}ields}, \lambda|k \mapsto k$, $\boldsymbol{\mathcal{C}}_{\text{ext}}$ becomes a large groupoid over the large groupoid of fields.

For the following propositions we consider the special case of $\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{RAMinputs}}$. We choose a relative field RAM representation of $\boldsymbol{\mathcal{RAMinputs}}_{\text{ext}} \longrightarrow \boldsymbol{\mathcal{FE}}$, that is, a representation by $\boldsymbol{\mathcal{RAMinputs}} \longrightarrow \boldsymbol{\mathcal{RAMinputs}}$ (as always over $\boldsymbol{\mathcal{Fields}}$), where $\boldsymbol{\mathcal{RAMinputs}} \longrightarrow \boldsymbol{\mathcal{FE}}$ is the representation already considered. (The field extensions are represented by multiplication matrices, and the elements of a fixed field extension are represented by their coordinate vectors in $k^{[\lambda:k]}$; we assume that this representation is "reasonable".)

The following proposition is obvious.

Proposition 1.57 *Let a generic field RAM* Π *be given. There exists a RAM* Π_{ext} *such that the following holds:*

For every field extension $\lambda|k$*, input* x *over* λ *and some input* y *over* k *representing* $(x, \lambda|k) \in \boldsymbol{\mathcal{RAMinputs}}_{\text{ext}}$*,* Π *terminates on* x *if and only if* Π_{ext} *terminates on* y*.*

For instances for which Π *terminates, the running time in field operations of* Π_{ext} *is in* $\mathcal{O}([\lambda : k]^3 \cdot$ *(the number of field operations used by* Π *on* $x)$*). An analogous statement holds for the running time in bit operations.*

The following two propositions follow from Propositions 1.55 and 1.56 respectively.

Proposition 1.58 *Let a generic field RAM enriched with* FIELDCHARROOT Π *be given. There exists a RAM* Π_{ext} *such that the following holds:*

For every field extension $\lambda|k$ *over a perfect field* k*, input* x *over* λ *and some input* y *over* k *representing* $(x, \lambda|k) \in \boldsymbol{\mathcal{RAMinputs}}_{\text{ext}}$*,* Π *terminates on* x *if and only if* Π_{ext} *terminates on* y*.*

For instances over perfect fields for which Π *terminates, the running time in field operations of* Π_{ext} *is in* $\mathcal{P}oly([\lambda : k] \cdot$ *(the number of field operations used by* Π *on* $x)$*). An analogous statement holds for the running time in bit operations.*

Proposition 1.59 *Now let a generic field RAM* Π *enriched by randomization be given. There exists a randomized RAM* Π_{ext} *such that the following holds:*

For every field extension $\lambda|k$*, input* x *over* λ *and some input* y *over* k *representing* $(x, \lambda|k) \in \boldsymbol{\mathcal{RAMinputs}}_{\text{ext}}$*,* Π *terminates on* x *in a finite expected number of field operations if and only if* Π_{ext} *terminates on* y *in a finite expected number of field operations.*

For instances for which Π *terminates in a finite expected number of field operations, the expected running time of* Π_{ext} *in field operations is in* $\mathcal{P}oly([\lambda : k] \cdot$ *(the expected number of field operations used by* Π *on* $x)$*). An analogous statement holds for the running time in bit operations.*

If now $\boldsymbol{\mathcal{C}}$ is any large groupoid represented by $\boldsymbol{\mathcal{RAM}inputs}$ (over $\boldsymbol{\mathcal{F}ields}$), then $\boldsymbol{\mathcal{C}}_{\text{ext}} \longrightarrow \boldsymbol{\mathcal{FE}}$ is relatively represented by $\boldsymbol{\mathcal{RAM}inputs}_{\text{ext}} \longrightarrow \boldsymbol{\mathcal{FE}}$ (again over $\boldsymbol{\mathcal{F}ields}$) which in turn is relatively represented by $\boldsymbol{\mathcal{RAM}inputs} \longrightarrow \boldsymbol{\mathcal{RAM}inputs}$. In this way, any computational problem $\mathcal{P} : \boldsymbol{\mathcal{C}} \longrightarrow \boldsymbol{\mathcal{C}}'$ leads to a computational problem $\mathcal{P}_{\text{ext}} : \boldsymbol{\mathcal{C}}_{\text{ext}} \longrightarrow \boldsymbol{\mathcal{C}}'_{\text{ext}}$. Now the three propositions above can be used to study the complexity of $\mathcal{P}_{\text{ext}}$ in relation to the complexity of the complexity of the original problem $\mathcal{P}$.

Similar remarks also hold for relative computational problems.

Chapter 2

Representations and basic computations

2.1 Introduction

In this chapter, we discuss various methods to represent the basic objects in the computations: curves as well as points, divisors and divisor classes on curves. Moreover, we state some results concerning computations with divisors, most importantly concerning the arithmetic in the divisor group and the computation of Riemann-Roch spaces, and applications on the arithmetic in the divisor class group. We restrict ourselves throughout to curves over *perfect fields*.

Throughout this most of this chapter, we use generic field RAMs enriched with the command FIELDCHARROOT (to determine p^{th} roots in characteristic $p > 0$) or randomized generic field RAMs (that is, generic field RAMs enriched with the commands FIELDFACTOR (to factorize polynomials) and RAND (for randomization) as computational models for the algorithms; see Section 1.6. If we use randomized generic field RAMs to derive a certain computational result, we state that we use a randomized algorithm. (If we do not use randomization, sometimes we stress this by saying that we have a deterministic algorithm, but not always.)

All running times (resp. expected running times) in field and bit operations we derive in this chapter are polynomially bounded in certain data associated with the input. For computations over finite fields $\mathbb{F}_q$, one can obtain running times (resp. expected running times) in bit-operations (on a bit-RAM) by multiplying the given running times (resp. expected running times) with a polynomial factor in $\log(q)$ (see subsection 1.6.6 for details). Note moreover that if we give a deterministic algorithm (that is, if we have a (deterministic) generic field RAM), and we restrict ourselves to instances over finite fields, then we have a deterministic algorithm in the bit-oriented

setting (that is, a (deterministic) bit-RAM) too.

At the end of the chapter we also discuss some computations over finite fields and in the randomized bit-RAM model.

2.2 Terminology and notation

We start off by fixing some terminology and notation which we use throughout this work. We also comment on some well known facts from algebraic geometry.

Generalities

We usually denote fields by k, λ and K. The algebraic closure of a field k is denoted by $\overline{k}$. The characteristic is always denoted by p.

If A is a graded commutative ring and M a graded A-module, we denote the degree-i part of M by M_i.

If A is a ring, I an ideal in A and $a \in A$, we denote the residue class of a in A/I by $[a]_I$. If $A = \mathbb{Z}$ and $I = (n)$, we also write $[a]_n$.

Closed subschemes

Let A be a commutative ring and I an ideal in A. Then we denote the closed *subscheme* defined by I (that is, the closed subscheme defined by the closed immersion $\mathrm{Spec}(A/I) \longrightarrow \mathrm{Spec}(A)$ induced by $A \longrightarrow A/I$) by $V(I)$. If $I = (f_1, \ldots, f_s)$, we set $V(f_1, \ldots, f_s) := V(I)$.

Let now A be a graded commutative ring and I a homogeneous ideal of A. Then similarly to the previous notation we denote the closed subscheme defined by I (the closed subscheme defined by the closed immersion $\mathrm{Proj}(A/I) \longrightarrow \mathrm{Proj}(A)$) by $V(I)$. If $I = (F_1, \ldots, F_s)$, we again set $V(F_1, \ldots, F_s) := V(I)$.

Base change

Let S be a scheme, let X be an S-scheme and let $T \longrightarrow S$ be a morphism. Then as usual we denote the T-scheme $X \times_S T$ by X_T. Note that the definition is in fact relative not only to T but also to the morphism from T to S, whereas the notation does not reflect this. This ambiguity does however not cause any problems in this work. Moreover, also as usual, if $T = \mathrm{Spec}(A)$, we set $X_A := X_T$.

Let again X be an S-scheme, let k be a field, and a morphism $\mathrm{Spec}(k) \longrightarrow S$ be given. Let us assume that X_k is integral. Then we set $k(X) := \kappa(X_k)$, the function field of X_k.

Meromorphic sections and divisors

For background on the following definitions and remarks on meromorphic sections and divisors we refer to [Gro67, §20, §21].

Let X be an integral scheme, and let $\mathcal{L}$ be an invertible sheaf on X. Let $\mathcal{M}_X$ be the sheaf of meromorphic (that is, rational) functions on X (for every non-empty open subset U of X, $\Gamma(U, \mathcal{M}_X) = \kappa(X)$). Further, let $\mathcal{M}(\mathcal{L}) := \mathcal{L} \otimes_{\mathcal{O}_X} \mathcal{M}_X$ be the sheaf of meromorphic sections of $\mathcal{L}$, and let $M(X, \mathcal{L}) := \Gamma(X, \mathcal{M}(\mathcal{L}))$ be the space of meromorphic sections of $\mathcal{L}$ on X. Now let $s \in M(X, \mathcal{L}) - \{0\}$ by a non-trivial meromorphic section of $\mathcal{L}$ on X. Then $(\mathcal{L}, s)$ defines a Cartier divisor (the divisor of zeroes of s) which we denote by $\mathrm{div}_{\mathcal{L}}(s)$ or by $\mathrm{div}(s)$ if $\mathcal{L}$ is obvious from the context. For $\mathcal{L} = \mathcal{O}_X$, the divisor of zeroes of s is the principal divisor of s which we denote by (s).

If D is a Cartier divisor on X, we denote the associated invertible subsheaf of the sheaf $\mathcal{M}_X$ by $\mathcal{O}(D)$. Note that by definition for every open subset U of X, $\Gamma(U, \mathcal{O}(D))$ is contained in $\Gamma(U, \mathcal{M}_X) = \kappa(X)$, and therefore $M(X, \mathcal{O}(D))$ is canonically isomorphic to $\kappa(X)$; we identify them. Note now that $\mathrm{div}_{\mathcal{O}(D)}(1) = D$ and more generally for $u \in \kappa(X)^*$, $\mathrm{div}_{\mathcal{O}(D)}(u) = (u) + D$. In particular,

$$\begin{aligned} \Gamma(\mathcal{C}, \mathcal{O}(D)) &= \{u \in \kappa(X)^* \mid \mathrm{div}_{\mathcal{O}(D)}(u) \geq 0\} \cup \{0\} \\ &= \{u \in \kappa(X)^* \mid (u) \geq -D\} \cup \{0\} \ . \end{aligned}$$

Another interesting fact is that every non-trivial meromorphic section s of an invertible sheaf $\mathcal{L}$, defines an isomorphism $\mathcal{L} \xrightarrow{\sim} \mathcal{O}(\mathrm{div}_{\mathcal{L}}(s))$.

If X is locally noetherian, we identify the effective Cartier divisors on $(\mathbb{P}^1_k)^n$ and the locally principal closed subschemes of $(\mathbb{P}^1_k)^n$ (cf. [Gro67, (21.2.12)]).

Affine and projective space

We set $\mathbb{A}^1 := \mathrm{Spec}(\mathbb{Z}[\mathrm{x}])$ and $\mathbb{P}^1 := \mathrm{Proj}(\mathbb{Z}[\mathrm{X}, \mathrm{Y}])$. We identify $\mathbb{Z}[\mathrm{x}]$ with $\mathbb{Z}[\frac{\mathrm{X}}{\mathrm{Y}}] \subset \kappa(\mathbb{P}^1)$ via $\mathrm{x} \longleftrightarrow \frac{\mathrm{X}}{\mathrm{Y}}$, and with this identification, we identify $\mathbb{A}^1$ with $\mathbb{P}^1 - V(\mathrm{Y})$.

We set $\mathbb{P}^2 := \mathrm{Proj}(\mathbb{Z}[X, Y, Z])$ (note the different font for the variables!). Furthermore we set $x := \frac{X}{Z}$ and $y := \frac{Y}{Z}$. Note that with this definition $\mathrm{Proj}(\mathbb{P}^2_A, \mathcal{O}(i))$ is canonically isomorphic to $A[X, Y, Z]_i$. In particular, we have the "homogeneous coordinate system" $X, Y, Z \in \Gamma(\mathbb{P}^n_A, \mathcal{O}(1))$ on $\mathbb{P}^n_A$.

Projective spaces of dimension > 2 are seldomly considered in this work, and we do not use coordinate systems of these spaces. However in Section 3.5 we consider products of $\mathbb{P}^1$'s, and then we identify $(\mathbb{P}^1)^n$ componentwise with $\prod_{i=1}^n \mathrm{Proj}(\mathbb{Z}[\mathrm{X}_i, \mathrm{Y}_i])$. We then set $\mathrm{x}_i := \frac{\mathrm{X}_i}{\mathrm{Y}_i}$ and define $\mathbb{A}^n := \mathrm{Spec}(\mathbb{Z}[\mathrm{x}_1, \ldots, \mathrm{x}_n])$.

Curves and their plane models

By a *curve* over a field k we mean a smooth and proper geometrically irreducible 1-dimensional k-scheme. We denote the genus of a curve $\mathcal{C}$ by $g = g(\mathcal{C})$.

Let k be a field, and let $\mathcal{C}$ be a curve over k.

We have $\mathbb{P}^1(\overline{k}) = \mathbb{A}^1(\overline{k}) \dot\cup \{\infty\}$ for a unique point $\infty \in \mathbb{P}^1(k) - \mathbb{A}^1(k)$ which we call the *point at infinity*. Note that every rational function on $\mathcal{C}$, that is, every element in $k(\mathcal{C})$, extends uniquely to a morphism from $\mathcal{C}$ to $\mathbb{P}^1_k$. Indeed, the set of morphisms from $\mathcal{C}$ to $\mathbb{P}^1_k$ without the constant morphism to ∞ are in bijection with $k(\mathcal{C})$. Note in particular that x then becomes the identity on $\mathbb{P}^1_k$.

For a (Cartier or Weil) divisor D on $\mathcal{C}$, we set $L(D) := \Gamma(\mathcal{C}, \mathcal{O}(D))$, and following [Heß01] we call this vector space the *Riemann-Roch space* of D.

We often make use of *plane models* of curves. By a *plane model* of $\mathcal{C}$ we mean a proper 1-dimensional subscheme of $\mathbb{P}^2_k$ which is over k birational to the curve. We denote a plane model of a curve $\mathcal{C}$ by $\mathcal{C}_{pm}$, and we usually fix a birational map $\pi : \mathcal{C} \longrightarrow \mathcal{C}_{pm}$. Moreover, we assume that $\mathcal{C}_{pm} \neq V(Z)$. We then denote the non-singular part of $\mathcal{C}_{pm}$ by $\mathcal{C}_{ns}$. We denote the degree of a plane model by d, that is $d = \deg(\mathcal{C}_{pm})$. As every invertible sheaf on $\mathbb{P}^n_k$ is isomorphic to $\mathcal{O}(d)$ for some $d \in \mathbb{Z}$, every plane model is given by an irreducible polynomial $F(X, Y, Z) \in k[X, Y, Z]$ (unique up to constants) which satisfies $\deg(F) = d$ (see [Har77, Corollary 6.17]).

So let $\pi : \mathcal{C} \longrightarrow \mathcal{C}_{pm} \subseteq \mathbb{P}^2_k$ be a birational map from a curve to one of its plane models. We set $\mathcal{O}_{\mathcal{C}}(i) := \pi^*(\mathcal{O}(i))$ for $i \in \mathbb{Z}$. Let $W \in M(\mathbb{P}^2_k, \mathcal{O}(i)) = k(X, Y, Z)_i$ be a meromorphic section degree i whose pole divisor does not contain $\mathcal{C}_{pm}$. Then $\pi^*(W) \in M(\mathcal{C}, \mathcal{O}_{\mathcal{C}}(i))$ is defined, and we set $W_{|\mathcal{C}} := \pi^*(W)$. Let now $W \in M(\mathbb{P}^2_k, \mathcal{O}(i))$ with $W \neq 0$ such that the support of $V(W)$ does not contain $\mathcal{C}_{pm}$. Then $W_{|\mathcal{C}} \neq 0$, and we have $\pi^*(\mathrm{div}(W)) = \mathrm{div}(W_{|\mathcal{C}})$. Note that these definitions and remarks in particular apply to rational functions on $\mathbb{P}^2_k$ (which we denote by small letters).

The divisor class group

Let $\mathcal{C}$ still be a curve over k. Let $\mathrm{P}(\mathcal{C})$ be the group of principal divisors on $\mathcal{C}$. Then the *(divisor) class group* of $\mathcal{C}$ is the group $\mathrm{Cl}(\mathcal{C}) := \mathrm{Div}(\mathcal{C})/\mathrm{P}(\mathcal{C})$. We denote the divisor class of a divisor D on $\mathcal{C}$ by $[D]$. The *degree* of a class $[D]$ is $\deg([D]) := \deg(D)$. Moreover, the *degree 0 (divisor) class group* of $\mathcal{C}$ is $\mathrm{Cl}^0(\mathcal{C}) := \mathrm{Div}^0(\mathcal{C})/\mathrm{P}(\mathcal{C})$.

Note that the definition of the divisor class group relies on the fact that principal divisors have degree 0.

Random variables

We frequently use random variables. We adopt the following convention: A random variable with values in a set X is called a *random element* of X. Likewise, for example, we talk about random vectors in a vector space, random vector subspaces of a fixed vector space, and so on. Note that this terminology is for example also used in the theory of random graphs; a "random graph" is in fact by definition a random variable with values in the set of graphs on a fixed set of vertices.

We would like to stress that every time we use this terminology we do in fact consider random variables, not specific elements which are chosen according to a certain distribution.

In fact, we reserve the terminology that a certain element is *chosen* with respect to a certain distribution to the description of algorithms (as a shorthand for applications of the command RAND), whereas otherwise, in particular in the analyses of algorithms, we use random variables.

Moreover, we denote random elements in a set X, that is random variables with values in X, just as the elements in X themselves. Indeed, the usual notation from probability theory, where one denotes random variables by capital letters and elements from the set by small letters is not applicable in our context because we consider random variables with values in various different sets whose elements are denoted in all kind of different ways.

2.3 Representing finite separable field extensions

As a preliminary consideration for computations with curves, we consider in this section the question how to represent finite separable field extensions and elements in these extensions.

In this section we furthermore – and for the last time – make the effort to in detail describe how one should define the categories, functors etc. for an application of the notions of the previous chapter. In the rest of this work, we use a less rigorous descriptions, as usual in computational mathematics. It is a rather straightforward but sometimes tedious task to turn formulate the statements rigorously using the general terminology on representations developed in the previous chapter.

Every finite separable extension of fields $\lambda|k$ is primitive, that is, there exists some $a \in \lambda$ with $\lambda = k[a]$. We address the question how to compute such a primitive element.

The computation is of course trivial if λ is represented over k as a primitive extension, that is, if it is represented by an irreducible polynomial $f \in k[t]$ with $\lambda \simeq k[t]/(f(t))$.

We now assume that λ is represented over k by a multiplication table. We first discuss the necessary setup to formulate the computational problem rigorously using the terminology of the previous chapter. Here we apply the definitions of Section 1.2 with the modifications of Section 1.6.

We start off with the large groupoid $\boldsymbol{SFE}$ of finite separable field extensions defined analogously to $\boldsymbol{ALG}$ and $\boldsymbol{FE}$. We represent this large groupoid by the large groupoid $\boldsymbol{\mathcal{M}}$ of multiplication tables (which we represent by $\boldsymbol{RAMinputs}$).

We define the large groupoid $\mathcal{C}'$ whose objects are tuples $(\lambda|k, a)$ of finite separable field extensions $\lambda|k$ and elements $a \in \lambda$ with the obvious morphisms. We represent such tuples by multiplication tables and coordinate vectors, such that we obtain a relative generic field RAM representation of $\mathcal{C}' \longrightarrow \mathcal{FSE}$.

We now consider the relative field oriented computational problem (over $\mathcal{FSE}$) given by $\mathcal{FSE} \longrightarrow \mathcal{C}', \lambda|k \mapsto \{(\lambda|k, a) \mid a \text{ is a primitive element of } \lambda|k\}$ together with: For every isomorphism $\alpha : \lambda \longrightarrow \lambda'$ of separable field extensions over k the morphism $P_\alpha : (\lambda|k, a) \mapsto (\lambda'|k, \alpha(a))$ and for every such α and a the induced morphism $(\lambda|k, a) \mapsto (\lambda'|k, \alpha(a))$ (called $m_{\alpha,a}$). (Note that this problem is a special case of an analog of Example 1.26 for generic field RAMs.)

We have the following satisfying proposition.

Proposition 2.1 *Given a finite field extension $\lambda|k$, represented by a multiplication table, one can with a randomized algorithm compute a primitive element of $\lambda|k$ (and its minimal polynomial) in an expected number of field and bit operations which is polynomially bounded in the extension degree $[\lambda : k]$.*

This proposition is essentially proven in [KM04a, Section 6]. We note however that the proof in [KM04a] does not immediately give the desired result because it is (essentially) based on k-RAMs whereas we want to derive the result on the basis of a generic field RAM. (As stated in the introduction to this chapter, we also allow the computation of p^{th} roots of unity in positive characteristic p, but we do not need this here.) Because of this and for the sake of completeness, we give the proof of the proposition below.

Remark 2.2 As always, the proposition implies a corresponding statement over finite fields and on a bit-RAM: Given an extension $\lambda|k$ of finite fields, represented by a multiplication table, one can with a randomized algorithm compute a primitive element of $\lambda|k$ (and its minimal polynomial) in an expected time which is polynomially bounded in the extension degree $[\lambda : k]$ and $\log(q)$.

Over finite fields and for the bit-RAM model, there is however a much easier proof than the one we give below: One uniformly randomly chooses an element of $\lambda|k$ and computes its minimal polynomial. As shown in [KM04a, Section 6], the probability that the element is primitive is $\geq \frac{1}{2}$.

Remark 2.3 One often encounters representations of finite separable field extensions in the following form: The extension is given by a tower of primitive extensions $\lambda = \lambda_r \supsetneq \lambda_{r-1} \supsetneq \cdots \supsetneq \lambda_0 = k$, where each λ_i is given in the form $\lambda_i \simeq \lambda_{i-1}[t_i]/(f_i(t_i))$ for $i = 1, \ldots, r$ and an explicit polynomial $f_i \in \lambda_{i-1}$. Then to represent the elements of λ, one recursively represents elements of λ_i by the polynomial basis given by t_i over λ_{i-1}.

Given such a representation, it is an easy task to compute a multiplication table of $\lambda|k$, and then one can apply Proposition 2.1 to find a primitive element.

Proof of Proposition 2.1

Before we come to the proof we note again that we want to derive the desired result on the basis of a generic field RAM. Note that the computational model requires that we can only access field elements which are algebraic combinations of the images of the integers and the input elements.

The algorithm is based on the following easy lemma from [BG04] (see also [KM04a, Lemma 3.8]), which we state here without proof.

Lemma 2.4 *Let W be an n-dimensional vector space over k with basis $w_1, \ldots, w_n$. Let $H_1, \ldots, H_m \subsetneq W$ be proper vector subspaces of W, and let $S \subseteq k$ be a finite set. Now let $c_1, \ldots, c_n$ be independently and uniformly distributed random elements of S. Then the probability that $c_1 w_1 + \cdots c_n w_n \in H_1 \cup \cdots \cup H_m$ is at most $\frac{m}{\#S}$.*

Let $n := [\lambda : k]$. Now an element $a \in \lambda$ is not primitive over k if and only if for some $i = 1, \ldots, n$, $\sigma_1(a) = \sigma_i(a)$, where $\sigma_i : \lambda \hookrightarrow \overline{k}$ for $i = 1, \ldots, n$ are the different embeddings of $\lambda|k$ into an algebraic closure $\overline{k}$ of k.

The idea is now to apply this lemma with $H_i := \{a \in \lambda \,|\, \sigma_1(a) = \sigma_{i+1}(a)\}$ $(i = 1, \ldots, m := n-1)$ and $\#S = 2m = 2([\lambda : k] - 1)$, except if the all input elements are algebraic over the prime field and the field generated by the input elements is too small for this.

The following lemma guarantees that we can generate sufficiently fast sufficiently many field elements from the input elements.

Lemma 2.5 *Let a vector $\underline{v} \in k^r$ and a natural number s be given. Let k_0 be the prime field of k and $R := k_0[v_1, \ldots, v_r]$. Then one can construct a vector consisting of $\min\{s, \#R\}$ different elements from R with $\mathcal{O}(s)$ field and bit operations (independently of n).*

We give an algorithm with the claimed properties below at the end of this subsection.

Now we show how we can apply the previous two lemmata to obtain a proof of Proposition 2.1.

Let $M = (m_{i,j,k})_{i,j,k}$ be the multiplication table. Analogously to the notation in the previous lemma, let k_0 be the prime field of k, and let $R := k_0[\{m_{i,j,k}\}_{i,j,k}]$. Let $m := n - 1$. Let $b_1, \ldots, b_n$ be the fixed basis of $\lambda|k$ with respect to which the multiplication table is given.

We are now coming to the algorithm.

We start off with computing a vector $\underline{a}$ over $R \subseteq k$ consisting of $\min\{2m, \#R\}$ elements of k. According to the previous lemma, this can be done with $\mathcal{O}(n) = \mathcal{O}([\lambda : k])$ field and bit operations.

Now we make a case distinction according to whether the length of the vector is $2m$ or less.

Assume first that the vector has length $2m$. We choose $i_1, \ldots, i_n \in \{1, \ldots, 2m\}$ uniformly at random and test whether the element given by $(a_{i_1}, \ldots, a_{i_n})$ defines a primitive element. This can be done in a number of field and bit operations which is polynomially bounded in $n = [\lambda : k]$. By Lemma 2.4 with $S = \{a_1, \ldots, a_{2m}\}$, the probability that the element is primitive is $\geq \frac{1}{2}$.

We now consider the case that the vector has smaller length. Then R is a finite field. We now first give some mathematical background and then return to the algorithm.

Let q be the power of p such that $R = \mathbb{F}_q$. Let λ_0 be the vector space which is generated by the basis elements $b_1, \ldots, b_n$ over R. By definition of R, the multiplication of λ restricts to λ_0; with the induced multiplication λ_0 is a finite extension field of R of degree n. Now λ is generated by λ_0 and k over R, and $[\lambda_0 : r] = [\lambda : k] = n$. This equality implies that λ_0 and k are linearly disjoint over R, that is, $\lambda = \lambda_0 \otimes_R k$. It follows in particular that the extension $\lambda|k$ is cyclic of order n. Let $\sigma_{\lambda_0|R}$ be the Frobenius automorphism of $\lambda_0|R$. Then the automorphism group of $\lambda|k$ is generated by $\sigma := \sigma_{\lambda_0|R} \otimes_R \mathrm{id}_k$, and σ is given on coordinate vectors by $(a_1, \ldots, a_n) \mapsto (a_1^q, \ldots, a_n^q)$.

Let now $d_i := [k[b_i] : k] = [R[b_i] : R]$. Then $\mathrm{lcm}(d_1, \ldots, d_n) = n$. There therefore exist $e_1, \ldots, e_n \in \mathbb{N}$ such that $e_i | d_i$, the e_i are coprime and $e_1 \cdots e_n = n$. Let now $c_i := N_{\mathbb{F}_q[b_i]|\mathbb{F}_{q^{e_i}}}(b_i) = b_i^{\frac{q^{d_i}-1}{q^{e_i}-1}}$. Then $R[c_i] = \mathbb{F}_{q^{e_i}}$. Therefore, the extension $k[c_i]|k$ has degree e_i too. By the properties of the f_i stated above, $c := c_1 \cdots c_n$ is a primitive element of $\lambda_0|R$ and of $\lambda|k$.

We now return to the algorithm. We first compute the minimal polynomials of the basis elements b_i. Then we in particular know the degrees d_i of the extensions $k[b_i]|k$. We determine suitable e_i and then the c_i. Finally, we

compute c. All these computations can obviously be performed as claimed.□

Proof of Lemma 2.5

The idea of the algorithm is: One tries to construct a vector containing s elements from k by first considering all elements in the prime field k_0, then from $k_0[v_1]$, then from $k_0[v_1, v_2]$ and so on, and furthermore, if the elements from $k_0[v_1, \ldots, v_{\ell-1}]$ have already been considered, one considers first the elements of $k_0[v_1, \ldots, v_\ell]$ of degree 1 in v_ℓ, then of degree 2 in v_ℓ and so on.

Note that if we have constructed a vector consisting of all elements of $k[v_1, \ldots, v_{\ell-1}]$ then in particular we know that $k[v_1, \ldots, v_{\ell-1}]$ is a finite field.

Algorithm for constructing a vector of different elements in a field

Input: A field k, $\underline{v} \in k^r$ and $s \in \mathbb{N}$.

1. Let $i \longleftarrow 0$
   Let $a_1 \longleftarrow 0_k \in k$
   Repeat
   Let $i \longleftarrow i + 1$
   Let $a_i \longleftarrow a_{i-1} + 1_k$
   If $i = s$, then output $\underline{a}$, STOP
   Until $a = 0_k$.
   Let $ch \longleftarrow i$
   (This is the characteristic of k.)
2. For $\ell = 1, \ldots, r$ do
   2.1. Let $v \longleftarrow 1_k$
   2.2. Repeat
   2.2.1. Let $v \longleftarrow v \cdot v_\ell$
   2.2.2. For $\ell = 1, \ldots, i$ do
   If $v = a_\ell$, then abandon the Repeat-loop
   2.2.3. Let $t \longleftarrow i$
   2.2.4. For $j = 1, \ldots, ch - 1$ do
   For $u = 1, \ldots, t$ do
   Let $i \longleftarrow i + 1$
   Let $a_i \longleftarrow v + a_{i-t}$
   If $i = s$, then output $\underline{a}$, STOP
3. Output $\underline{a}$.

We sketch the proof of the correctness of the algorithm.
Note first that i is always the length of the array already constructed.

In Step 1 an array of distinct elements of the prime field k_0 is constructed. Now either the algorithm stops with an array of s different field elements from k_0 or at the end of Step 1 the array contains all elements from k_0.

Let us now consider Step 2.

Now the following statements always hold:

- v is a power of v_ℓ.
- Before the execution of Step 2.1.: $k_0[v_1, \ldots, v_{\ell-1}]$ is a field and $\underline{a}$ contains all elements from this field exactly once.
- After Step 2.2.1.: Let $v = v_\ell^d$. (This also applies to the next items.) Then $\underline{a}$ contains all elements of $k_0[v_1, \ldots, v_\ell]$ of degree $< d$ in v_ℓ.
- Concerning Step 2.2.2.: This loop is abandoned if and only if v_ℓ is algebraic over $k_0[v_1, \ldots, v_{\ell-1}]$ and its minimal polynomial has degree d.
- After Step 2.2.4.: Then $a_1, \ldots, a_t$ contains all elements of $k_0[v_1, \ldots, v_\ell]$ of degree $< d$ in v_ℓ and furthermore either v_ℓ is transcendental over $k_0[v_1, \ldots, v_{\ell-1}]$ or it is algebraic and its minimal polynomial has degree $> d$.

The enlargement of the vector in Step 2.2.4. is based on the following fact: Let $a_1, \ldots, a_t$ contain all elements of $k_0[v_1, \ldots, v_\ell]$ of degree $< d$ in v_ℓ exactly once. Then the elements of $k_0[v_1, \ldots, v_\ell]$ of degree $\leq d$ in v_ℓ can uniquely be written as $cv^d + a_x$ with $c \in k_0$ and $x \in \{1, \ldots, t\}$. Moreover, of course, $(c+1)v^d + a_x = v^d + (cv^d + a_x)$.

It is obvious that the algorithm has the claimed complexity. □

2.4 Representing curves

We represent curves by plane models which are birational to the curve. The plane models themselves are represented by a defining homogeneous polynomial in $k[X, Y, Z]$.

The following proposition is [Heß05, Theorem 56].

Proposition 2.6 *Any curve over a finite field has a plane model of degree $\mathcal{O}(g)$ (uniformly over all finite fields).*

This proposition is complemented by the following proposition for curves over infinite fields. Together these two propositions indicate that it is reasonable to represent curves by plane models for computational purposes.

Proposition 2.7 *Every curve of genus ≥ 1 with a divisor of degree 1 over an infinite field has a plane model of degree at most $4g$.*

Proof. Every genus 1 curve with such a divisor is isomorphic to a plane cubic (if D is a divisor of degree 1, the linear system $|3D|$ is very ample of (projective) dimension 2 and degree 3 and thus defines an embedding into $\mathbb{P}^2_k$).

Let $\mathcal{C}$ be a curve of genus $g \geq 2$. Let first $\mathcal{C}$ be hyperelliptic. Then we have a covering of degree 2 $\mathcal{C} \longrightarrow \mathcal{D}$ with $g(\mathcal{D}) = 0$. Now as $\mathcal{C}$ has a divisor of degree 1, so has $\mathcal{D}$. Therefore, $\mathcal{D} \approx \mathbb{P}^1_k$. Now by Artin-Schreier and by Kummer theory, $k(\mathcal{C}) \simeq k(\mathrm{x})[y]/(f(\mathrm{x}, y))$, for a polynomial $f(\mathrm{x}, y) \in K[\mathrm{x}, y]$ which has degree 2 in x and degree $2g + 1$ or $2g + 2$ in y.

So let $\mathcal{C}$ be non-hyperelliptic, and let D be a divisor of degree 1 on $\mathcal{C}$. Note that for $n \geq 2g - 1$, nD is non-special, thus $\dim(L(nD)) = n + 1 - g$.

Now depending on the characteristic of k we proceed as follows:

If the characteristic divides $2g$, then we choose some $u \in L((2g+1)D) - L(2gD)$. Otherwise we choose some $u \in L(2gD) - L((2g-1)D)$. Now in any case the degree of the covering $u : \mathcal{C} \longrightarrow \mathbb{P}^1_k$, that is, the extension degree $[k(\mathcal{C}) : k(u)]$, is prime to the characteristic. In particular, the extension $k(\mathcal{C})|k(u)$ is separable.

Let $u, v_1, \ldots, v_{g+1}$ be a basis of $L((2g + 1)D)$. Note that $u, v_1, \ldots, v_{g+1}$ generate $k(\mathcal{C})$ over k because $(2g+1) \cdot D$ is very ample; see [Har77, Corollary 3.2]. With other words, we have $k(\mathcal{C}) = k(u)[v_1, \ldots, v_{g+2}]$.

Now by the theorem of the primitive element and its proof in [Lan93] ([Lan93, V, §4, Theorem 4.6]), there exist $a, a_1, \ldots, a_{g+2} \in k$ such that $k(\mathcal{C}) = k(u, v)$ with $v := au + a_1u_1 + \cdots + a_{g+2}u_{g+2}$.

This implies that $u, v, 1$ define a rational map to $\mathbb{P}^2_k$ (which of al always extends to a morphism) which is birational onto its image (that is, the image is a plane model of the curve).

Let $F \in K[X, Y, Z]$ be the polynomial defining the plane model. Note that F is unique up to multiplication by a scalar, and it is a polynomial of minimal degree with $F(u, v, 1) = 0$.

By the Riemann-Roch Theorem, for $i \geq 1$, $L(i \cdot (2g+1) \cdot D)$ has dimension $i \cdot (2g+1) + 1 - g$. On the other hand, the space of homogeneous polynomials in 3 variables of degree i has dimension $\binom{i+2}{2} = (i + 2)(i + 1)/2$. For $i \geq 4g$, we have $(i + 2)(i + 1)/2 > i \cdot (2g + 1) + 1 - g$, and there exists some non-zero polynomial $G \in K[X, Y, Z]$ with $G(u, v, 1) = 0$. This implies that $\deg(F) \leq 4g$. □

We will use Proposition 2.6 later in this work, in Sections 3.3 and 3.4. Proposition 2.7 will in fact never be used, and for the time being we impose no restriction on the degree d.

2.5 Representing points and divisors and related computational problems

In this section, we present various possibilities of representing closed points and divisors on curves over perfect fields, starting from a plane model. We also address related computational aspects like for example the arithmetic in the divisor group, the computation of principal divisors, and the computation of Riemann-Roch spaces.

2.5.1 Overview

In this subsection we first give a brief overview over different approaches to represent closed points and divisors over perfect fields appearing in the literature. We then briefly mention the different approaches to compute Riemann-Roch spaces which are described in the literature. We also make some historical remarks.

The following three different approaches to represent closed points on curves over perfect fields are classical and appear often in the literature.

The first (and arguably most intuitive) approach to represent closed points of a curve is based on coordinates (in finite extension fields of the ground field) of points of a plane model. Divisors can then be represented by giving the support and the coefficients. We describe this approach briefly in subsection 2.5.2. For a fixed curve with a fixed birational morphism to a plane model this description as such leads however often only to a partial representation of points and divisors on the curve. If the plane model has singularities with various branches one needs additional "local" information to represent points lying over these singular points. One possibility to represent points lying over singular points is to use truncated parameterizations of local branches. The description is particularly easy for ordinary singular points (that is, if the tangents of the local branches are all distinct). In characteristic 0 or in "large" positive characteristic, one can use truncated Newton-Puiseux expansions for arbitrary singularities. However, in "small" characteristic one cannot use Newton-Puiseux expansions anymore in general. A natural substitute are then the more complex *Hamburger-Noether expansions*: These are certain systems of polynomials together with one power series. As pointed out by A. Campillo and J. Farrán in [CF05], one might also use *symbolic Hamburger-Noether expansions*. Here the power series is substituted by a polynomial in two variables.

The second approach is based on *ideal theory* of function fields. Here one realizes the curve as a covering of the projective line and one defines two orders in the function field, a "finite" and an "infinite" order. Now, there is a bijection between the divisor group on the curve and the product of the

two ideal groups of the two orders. One can represent each divisor by the corresponding pair of ideals. The ideals in turn can then conveniently be represented by module bases (for example by so-called *Hermite normal form bases*). We call this representation *joint ideal representation*. Alternatively, one can represent each divisor by as a formal sum of prime divisors, where each prime divisor is represented as a prime ideal of one of the two orders. We call this representation the *free ideal representation*. We describe the ideal theoretic approach in detail in subsection 2.5.4. Note that the so-called "Mumford representation" of so-called semi-reduced divisors on hyperelliptic curves in "imaginary quadratic representation" is a special case of the joint ideal representation.

Thirdly, one can represent effective divisors by linear subspaces of Riemann-Roch spaces of sufficiently high degree or more generally of spaces of global sections of invertible sheaves of sufficiently high degree. We call the corresponding representation the *joint global representation*, and again we have a related free representation. This approach is addressed in subsection 2.5.5.

An important basic computational problem related to divisors is the computation of the Riemann-Roch space $L(D)$ of a divisor D. This is in particular so because together with algorithms for the computation of principal divisors and addition / subtraction of divisors, every algorithm to compute Riemann-Roch spaces leads to an algorithm for the computation in degree 0 class groups of curves.

The three approaches to represent divisors described above are closely related to three approaches for algorithms for these problems.

A classical approach for the effective determination of Riemann-Roch spaces relies on the theory of adjoints and the "Residue Theorem" by A. Brill and M. Noether from 1874 ([BN74]). In fact, the first effective method for the computation of Riemann-Roch spaces was given by M. Noether in 1884 ([Noe84]). Noether's approach was in particular suggested by Goppa ([Gop82]) in 1982 to construct algebraic-geometric codes. It was subsequently studied by various authors, including (in chronological order) D. Le Brigand and J. Risler ([LBR88]), M.-D. Huang and D. Ierardi ([HI94]), E. Volcheck ([Vol94], [Vol95]) and finally A. Campillo and A. Farrán ([CF05]).

The ideal theoretic approach leads to an easy algorithm with a satisfying complexity due to F. Heß ([Heß01]). This algorithm from about the year 1999 is inspired by the ideal-theoretic proof of the Riemann-Roch Theorem by K. Hensel and G. Landsberg from about a century earlier ([HL02]). However, in contrast to the work by Hensel and Landsberg, no expansions are used. We present this algorithm in subsubsection 2.5.4.7.

The third approach leads to elegant and very fast algorithms based on

linear algebra due to K. Khuri-Makdisi ([KM04b], [KM04a]). It should however be noted that if one starts with a curve represented by a plane model, to set up the representation via the third approach, one needs first to compute Riemann-Roch space of some divisors of sufficiently high degree.

One of the main goals of this section is to show that one can represent divisors in such a way that one can perform various computational problems related to divisors efficiently. In particular, one goal is to show that – with an appropriate representation – one can compute Riemann-Roch spaces of divisors in a number of field and bit operations which is polynomially bounded in d, the degree of the plane model, and the so-called height of the divisor. Using the theory of adjoints, partial results were established by M.-D. Huang and D. Ierardi ([HI94]) and E. Volcheck ([Vol94]): M.-D. Huang and D. Ierardi showed the result for curves given plane models with ordinary singularities which are defined over the ground field,[1] using a different representation (which can be easily obtained from the coordinate representation): They represent systems of points by their so-called Chow-form (which gives a "dual description").

Volcheck then studied curves with ordinary singularities of any kind. He gave an appropriate algorithm which does however require that the ground field is "not too small" in relation to the degree of the curve.

Using the ideal-based representation and the ideal theoretic approach to Riemann-Roch spaces, the desired result was established by Heß. Indeed, the author is of the opinion that it is much easier to show this result in full generality with the ideal theoretic method than with the expansion-based method via adjoints.

Based on these considerations, in the following we *only discuss expansion free representations and corresponding algorithms in detail. In doing so we put particular emphasis on the ideal theoretic approach.*

The reader is referred to [CF05] for an up-to-date account for computations with divisors using Hamburger-Noether expansions. Note, however, that no complexity estimates are given in this work.

2.5.2 The coordinate representation

As stated above, we consider curves over perfect fields, represented by plane models. Let $\mathcal{C}$ be such a curve with a fixed plane model $\mathcal{C}_{pm}$. We rep-

[1] The introduction as well as the results in [HI94] are formulated as if they did hold for arbitrary curves provided that the singularities are defined over the ground field. This is however not the case, and in fact it is implicitly assumed throughout the work that the singularities are ordinary. The first occurrence is in subsection 1.1., where the authors claim that a singularity of a plane curve can be resolved by another plane curve of at most twice the degree of the original curve. Another occurrence is in the proof of Lemma 2.1.

resent both the curve and the plane model by a homogeneous polynomial $F(X,Y,Z) \in k[X,Y,Z]$ defining the plane model. Let $\pi : \mathcal{C} \longrightarrow \mathcal{C}_{pm}$ be a fixed birational map. Let $g = g(\mathcal{C})$ be the genus of $\mathcal{C}$, $d = \deg(C_{pm}) = \deg(F)$ the degree of the plane model, and let p be the characteristic of k; cf. the notations in section 2.2

For any extension $\lambda|k$ of fields, the points in $\mathcal{C}_{pm}(\lambda)$ can easily be represented by their coordinates in λ. This also provides an easy representation for all points in $\mathcal{C}(\lambda)$ which lie over *non-singular* points of $\mathcal{C}_{pm}$.

Now every closed point P of $\mathcal{C}_{pm}$ (resp. every closed point of $\mathcal{C}$) can be represented by a finite field extension $\lambda|k$ and one point in $\mathcal{C}(\lambda)$ (resp. in $\mathcal{C}(\lambda)$) defining P. (We can for example set $\lambda := \kappa(P)$, such that $[\lambda : k] = \deg(P)$.) In order to bound the length of the representation, we choose some $c \geq 1$ and consider only such extensions $\lambda|k$ for which $[\lambda : k] < \deg(P)^c$ (uniformly over all curves $\mathcal{C}$ and all plane models). Following the definitions in Section 1.3, abbreviate this by saying that "$[\lambda : k]$ is polynomially bounded in $\deg(P)$".

The field extension $\lambda|k$ can then for example be represented by a multiplication table or by a finite separable polynomial over k. Note here again that a separable defining polynomial can be computed from a multiplication table with a randomized algorithm in an expected number of field and bit operations which is polynomially bounded in $[\lambda : k]$.

We call the partial representation of closed points of curves by coordinates in finite extension fields in plane models we just described the *coordinate representation*.

This partial representation immediately leads to a representation of divisors whose support is disjoint from the points lying over singular points of the plane model: Similarly to the sparse representation of vectors one stores the support and the coefficients. We call this partial representation of divisors *free coordinate representation*.

We note here that obviously in a similar manner every (partial) representation of points leads to a corresponding (partial) representation of divisors, which we then call a *free representation*.

Points of $\mathcal{C}$ lying over the singular points of $\mathcal{C}_{pm}$ are not so easy to deal with. As a generalization of the representation via coordinates, it is natural to consider some expansion based description for these points. This is quite easy if one is concerned with ordinary singularities (that is, the tangent lines of all local branches are different), but the general case (in particular in small positive characteristic) is quite technical. As already mentioned above, we refer to [CF05] for further information.

2.5.3 Reduced matrices and Hermite normal forms

In the next subsection we will discuss how one can use ideal theory in function fields to represent and compute with divisors of curves. In doing so we will extensively use *Hermite normal form bases (HNF-bases)* of $k[\mathrm{x}]$- and $k[\frac{1}{\mathrm{x}}]$-submodules of finite field extensions of $k(\mathrm{x})$.

Additionally, in Heß' algorithm to compute Riemann-Roch spaces matrices over $k[\mathrm{x}]$ we call *degree-reduced* are used.

In this subsection we first give an overview over degree-reduced matrices. Then we briefly discuss applications of Gröbner base theory to matrices over $k[\mathrm{x}]$. Finally, we turn to Hermite normal forms and HNF-bases. Here we restrict ourselves to definitions and statements with respect to $k[\mathrm{x}]$ since the definitions and statements with respect to $k[\frac{1}{\mathrm{x}}]$ can immediately be obtained by applying the isomorphism $k[\mathrm{x}] \simeq k[\frac{1}{\mathrm{x}}]$, $\mathrm{x} \longleftrightarrow \frac{1}{\mathrm{x}}$.

T. Mulders and A. Storjohann have obtained particularly efficient algorithms for various computational tasks we consider in this subsection, like the computation of determinants and Hermite normal forms of matrices over $k[\mathrm{x}]$ (see [MS03]). For readers who are interested in these issues we recommend to read this work in conjunction with this subsection. Note however that in [MS03] row-operations on matrices are considered whereas we consider column-operations. When citing [MS03] we implicitly transpose all definitions and statements.

2.5.3.1 Degree-reduced matrices

Definition 2.8 Let R be a domain, and let $A, B \in \mathrm{Quot}(R)^{r\times s}$. Then B is *R-right-equivalent* or *R-column equivalent* to A if there exists a unimodular matrix (that is, an invertible matrix) $U \in R^{s\times s}$ with $B = AU$. Similarly, B is *R-left-equivalent* or *R-row equivalent* to A if there exists a unimodular matrix $T \in R^{r\times r}$ with $B = TA$.

In this work, we use column vectors for elements in R^r. Consequently, we consider matrices up to right-equivalence.

Definition 2.9 Let $g_1, g_2 \in k[\mathrm{x}]$ with $g_1, g_2 \neq 0$. Then the *valuation-degree* of $\frac{g_1}{g_2} \in k(\mathrm{x})$ is $\mathrm{valdeg}(\frac{g_1}{g_2}) := \deg(g_1) - \deg(g_2)$. Moreover, we set $\mathrm{valdeg}(0) := -\infty$.

Remark 2.10 Note that $v_\infty(\frac{g_1}{g_2}) = -\mathrm{valdeg}(\frac{g_1}{g_2})$, where v_∞ is the degree valuation on $k(\mathrm{x})$ (corresponding to the point ∞ on $\mathbb{P}^1_k$).

We remark that we use the phrase "valuation-degree" instead of merely "degree" because for non-constant $\frac{g_1}{g_2}$ the degree of $\frac{g_1}{g_2}$ interpreted as an element in $k(\mathbb{P}^1)$, that is, as a covering $\mathbb{P}^1_k \longrightarrow \mathbb{P}^1_k$, is equal to the height of the divisor $(\frac{g_1}{g_2})$, and if $\gcd(g_1, g_2) = 1$, this is equal to $\max\{\deg(g_1), \deg(g_2)\}$.

Definition 2.11 Let $g_1, g_2 \in k[\mathrm{x}]$ with $g_1, g_2 \neq 0$, and let $d := \mathrm{valdeg}(\frac{g_1}{g_2})$. Then the *leading coefficient* of $\frac{g_1}{g_2}$ is the unique element $a \in k$ with $\mathrm{valdeg}(\frac{g_1}{g_2} - a\mathrm{x}^d) < d$.

Explicitly, let b be the leading coefficient of the polynomial g_1, and let c be the leading coefficient of the polynomial of g_2. Then $a = \frac{b}{c}$. Indeed, we have $\deg(g_1 - \frac{b}{c}\mathrm{x}^{\deg(g_1)-\deg(g_2)}g_2) < \deg(g_1)$, thus $\mathrm{valdeg}(\frac{g_1}{g_2} - \frac{b}{c}\mathrm{x}^{\mathrm{valdeg}(\frac{g_1}{g_2})}) < \mathrm{valdeg}(\frac{g_1}{g_2})$.

Definition 2.12 Let now $\underline{v} \in k(\mathrm{x})^r$ for some r. Then the *valuation-degree* of $\underline{v}$ is $\mathrm{valdeg}(\underline{v}) := \max_{i=1}^r \mathrm{valdeg}(v_i)$. If $\underline{v} \in k[\mathrm{x}]^r$, we speak also of the *degree* of $\underline{v}$, $\deg(\underline{v})$. Similarly, the *valuation-degree* of a matrix over $k(\mathrm{x})$ is the maximum of the valuation-degrees of its entries.

Definition 2.13 Let $\underline{v} \in k(\mathrm{x})^r$, $\underline{v} \neq \underline{0}$. Let $d = \mathrm{valdeg}(\underline{v})$ and $\underline{v} = \underline{a}\mathrm{x}^d + \underline{w}$, where $\mathrm{valdeg}(\underline{w}) < d$. Then $\underline{a}$ is called the *vector of leading coefficients* of $\underline{v}$, denoted $\mathrm{lc}(\underline{v})$. We complement this definition with $\mathrm{lc}(\underline{0}) := \underline{0}$.

Moreover, for $A = (\ \underline{a}_1\ \cdots\ \underline{a}_s\ ) \in k[\mathrm{x}]^{r\times s}$, we set $\mathrm{lc}(A) := (\ \mathrm{lc}(\underline{a}_1)\ \cdots\ \mathrm{lc}(\underline{a}_s)\ ) \in k^{r\times s}$.

Let in the following $A \in k(\mathrm{x})^{r\times s}$ be a matrix with columns $\underline{a}_1, \ldots, \underline{a}_s$.

Lemma 2.14 *If the columns of A are $k(\mathrm{x})$-linearly dependent, the columns of $\mathrm{lc}(A)$ are k-linearly dependent.*

Proof. The statement is trivial if A has a zero-column. So let us assume that this is not the case.

Let $\underline{b} \in k(\mathrm{x})^s, \underline{b} \neq \underline{0}$ be such that $A\underline{b} = \underline{0}$. Let $d := \max_{j=1}^s \mathrm{valdeg}(b_j\underline{a}_j)$, and let

$$c_j := \begin{cases} b_j & \text{if } \mathrm{valdeg}(b_j\underline{a}_j) = d \\ 0 & \text{if } \mathrm{valdeg}(b_j\underline{a}_j) < d\,. \end{cases}$$

Then $\underline{c} \neq \underline{0}$ and $\sum_{j=1}^s c_j\mathrm{lc}(\underline{a}_j) = \underline{0}$. □

The following lemma is easy.

Lemma 2.15 *The following conditions are equivalent:*

a) For all $\underline{b} \in k[\mathrm{x}]^s$, $\mathrm{valdeg}(A\,\underline{b}) = \max_{j=1}^s \mathrm{valdeg}(b_j\,\underline{a}_j)$.

b) The non-zero columns of $\mathrm{lc}(A)$ are linearly independent.

c) Let $\tilde{A} \in k(\mathrm{x})^{r\times t}$ be the matrix obtained from A by deleting the zero-columns. Then $\tilde{A}$ has full column rank and the maximum of the degrees of the determinants of the $t \times t$-minors of $\tilde{A}$ is equal to $\sum_{j=1}^s \mathrm{valdeg}(\underline{a}_j)$.

In [Heß01] and related works a matrix $A \in k(\mathrm{x})^{r\times s}$ of full column rank is called *reduced* if the conditions of the lemma are satisfied. As however the word "reduced" occurs frequently in the literature with different meanings, and we also use it differently below, we define:

Definition 2.16 The matrix A is *degree-reduced* if the conditions of the lemma are satisfied.

Remark 2.17 Let $\tilde{A}$ be the matrix which is obtained from A by deleting the zero-columns. Then A is degree-reduced if and only if $\tilde{A}$ is, and this means in particular that $\tilde{A}$ has full column rank.

Remark 2.18 Let $T \in k(\mathrm{x})^{r\times r}$. Then $T \in (k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})})^{r\times r}$ if and only if $\mathrm{valdeg}(T) \leq 0$ if and only if $\mathrm{valdeg}(T\underline{v}) \leq \mathrm{valdeg}(\underline{v})$ for all $\underline{v} \in k(\mathrm{x})^r$. Let now T be invertible in $(k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})})^{r\times r}$. Then $\mathrm{valdeg}(T\underline{v}) = \mathrm{valdeg}(\underline{v})$ for all $\underline{v} \in k(\mathrm{x})^r$.[2] By item a) of Lemma 2.15 this implies in particular that A is degree-reduced if and only if TA is degree-reduced.

Proposition 2.19

a) There exists a degree-reduced matrix $M \in k(\mathrm{x})^{r\times s}$ which is $k[\mathrm{x}]$-right-equivalent to A.

b) Let $M, M' \in k(\mathrm{x})^{r\times s}$ be degree-reduced matrices which are both $k[\mathrm{x}]$-right-equivalent to A such that the degrees of the columns are monotonically increasing from left to right. Then $\mathrm{valdeg}(\underline{m}_j) = \mathrm{valdeg}(\underline{m}'_j)$ for all $j = 1, \ldots, s$.

c) Given a matrix $A \in k[\mathrm{x}]^{r\times s}$, one can compute such a matrix M in a number of field and bit operations which is polynomially bounded in $\deg(A)$, r and s.

Proof. Let $g \in k(\mathrm{x}), g \neq 0$. Then obviously A is degree-reduced if and only if gA is. For the existence of M we can therefore restrict ourselves to matrices in $k[\mathrm{x}]^{r\times s}$.

Both the existence of M in a) as well as the computational result in c) follow from an obvious *reduction algorithm*:

Let us assume that $A \in k[\mathrm{x}]^{r\times s}$ is not degree-reduced. By a column swap we can obtain that there exists some $t \in \{1, \ldots, s\}$ such that columns $1, \ldots, t$ are zero and columns $t+1, \ldots, s$ are non-zero. Let $\underline{b} \in \ker(\mathrm{lc}(A))$ with $\underline{b} \neq \underline{0}$ with $b_1 = \cdots = b_t = 0$. Now let $k \in \{t+1, \ldots, s\}$ be such that

[2] One can show that conversely any matrix $T \in k(\mathrm{x})^{r\times r}$ with $\mathrm{valdeg}(T\underline{v}) = \mathrm{valdeg}(\underline{v})$ for all $\underline{v} \in k(\mathrm{x})^r$ is invertible in $(k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})})^{r\times r}$. Indeed, this follows easily with Hermite normal forms with respect to $(k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})})^{r\times r}$.

$\deg(b_k \underline{a}_k)$ is maximal. (This means that $b_k \neq 0$ and for all $j = 1, \dots s$ with $j \neq k$ and $b_j \neq 0$, $\deg(\underline{a}_j) \leq \deg(\underline{a}_k)$.) Then $a'_k := \sum_j b_j \mathrm{x}^{\deg(\underline{a}_k)-\deg(\underline{a}_j)} \underline{a}_j = b_k \underline{a}_k + \sum_{j \neq k} b_j \mathrm{x}^{\deg(\underline{a}_k)-\deg(\underline{a}_j)} \underline{a}_j$ has smaller degree than $\underline{a}_k$. Note here that the coefficients are all in $k[\mathrm{x}]$.

This means that if we substitute $\underline{a}_k$ by $\underline{a}'_k$, the sum of the degrees of the columns decreases. After finitely many such reduction steps, the procedure has to terminate, and the result is a degree-reduced matrix. More precisely, the number of reduction steps is bounded by $\deg(A) \cdot s$.

A possible algorithm consists of the following loop: Swap the zero-columns to the left and let $A = (O|\tilde{A})$ as above. Then compute $\ker(\tilde{A})$. If $\ker(\tilde{A}) = 0$, the algorithm terminates, otherwise a reduction step is performed as indicated.

We come to item b). Let $\mathfrak{M}$ be the $k[\mathrm{x}]$-module generated by the columns of A. Now let $d \in \mathbb{N}$, and let $\mathfrak{M}_d$ be the $k[\mathrm{x}]$-submodule of $\mathfrak{M}$ generated by the elements of $\mathfrak{M}$ of degree $\leq d$. As M (resp. M') is degree-reduced, the columns of degree $\leq d$ of M (resp. M') generate $\mathfrak{M}_d$. This implies that the columns of degree d form $k[\mathrm{x}]$-bases of $\mathfrak{M}_d/\mathfrak{M}_{d-1}$. This implies the claim.□

The following observations are crucial for the ideal-theoretic approach to Riemann-Roch spaces and thus for Heß' algorithm (see subsubsection 2.5.4.7).

Lemma 2.20 *Let $A \in k(\mathrm{x})^{r \times r}$ be of full rank. Then the following conditions are equivalent:*

a) A is degree-reduced.

b) $\mathrm{valdeg}(\det(A)) = \sum_{j=1}^{r} \mathrm{valdeg}(\underline{a}_j)$.

c) The matrix $T := A \operatorname{diag}(\mathrm{x}^{-\mathrm{valdeg}(\underline{a}_1)}, \dots, \mathrm{x}^{-\mathrm{valdeg}(\underline{a}_r)}) \in (k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})})^{r \times r}$ is unimodular.

d) There exists a unimodular matrix $T \in (k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})})^{r \times r}$ and $d_1 \geq \dots \geq d_r \in \mathbb{Z}$ such that $A = T \operatorname{diag}(\mathrm{x}^{-d_1}, \dots, \mathrm{x}^{-d_r})$.

Lemma 2.21 *Let $A \in k(\mathrm{x})^{r \times r}$ be of full rank. Then there exists uniquely determined integers $d_1 \geq \dots \geq d_r$ and unimodular matrices $T \in (k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})})^{r \times r}$, $U \in k[\mathrm{x}]^{r \times r}$ such that $AU = T \operatorname{diag}(\mathrm{x}^{-d_1}, \dots, \mathrm{x}^{-d_r})$.*

Proof. The existence is obvious. For the uniqueness, let $d_1 \geq \dots \geq d_r \in \mathbb{Z}$ and $d'_1 \geq \dots \geq d'_r \in \mathbb{Z}$, and let $T \in (k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})})^{r \times r}, U \in k[\mathrm{x}]$ with $\operatorname{diag}(\mathrm{x}^{-d_1}, \dots, \mathrm{x}^{-d_r})\, U = T \operatorname{diag}(\mathrm{x}^{-d'_1}, \dots, \mathrm{x}^{-d'_r})$. By the previous lemma, the right-hand side and thus also the left-hand side is degree-reduced. Now

the matrices $\mathrm{diag}(\mathrm{x}^{-d_1},\dots,\mathrm{x}^{-d_r})$ and $T\,\mathrm{diag}(\mathrm{x}^{-d'_1},\dots,\mathrm{x}^{-d'_r})$ are $k[\mathrm{x}]$-right-equivalent, and thus by Proposition 2.19 b) $d_i = d'_i$ for all $i = 1,\dots,r$. □

Proposition 2.22 *Let V be an r-dimensional $k(\mathrm{x})$-vector space ($r < \infty$). Let $\mathfrak{M}_0$ be a finitely generated $k[\mathrm{x}]$-submodule of V and $\mathfrak{M}_\infty$ a finitely generated $k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})}$-submodule of V, both of rank r. Then there exist uniquely determined integers $d_1 \geq \cdots \geq d_r$ and a basis $v_1,\dots,v_r$ of $\mathfrak{M}_0$ such that $\mathrm{x}^{d_1}v_1,\dots,\mathrm{x}^{d_r}v_r$ is a $k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})}$-basis of $\mathfrak{M}_\infty$. Then the following holds:*

- *For all $n \in \mathbb{Z}$, the set*

$$\bigcup_{i=1}^{r} \{\mathrm{x}^j v_i \mid 0 \leq j \leq d_i + n\}$$

  is a k-basis of $\mathfrak{M}_0 \cap \mathrm{x}^n \cdot \mathfrak{M}_\infty$.

- *Let for $n \in \mathbb{Z}$ $\iota(n) := \max\{j \in \{1,\dots,r\} \mid d_j + n \geq -1\}$. Then $\dim(\mathfrak{M}_0 \cap x^n \cdot \mathfrak{M}_\infty) - \dim(\mathfrak{M}_0 \cap x^{n-1} \cdot \mathfrak{M}_\infty) = \iota(n-1)$.*

Proof. Let $b_1,\dots,b_r$ be a $k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})}$-basis of $\mathfrak{M}_\infty$ and $\tilde{v}_1,\dots,\tilde{v}_r$ a $k[\mathrm{x}]$-basis of $\mathfrak{M}_0$. Let A be the coordinate matrix of $\tilde{v}_1,\dots,\tilde{v}_r$ with respect to $b_1,\dots,b_r$, and let $AU = M$ with $U \in k[\mathrm{x}]^{r\times r}$ unimodular and M degree-reduced. Let $M = TD$ with $D = \mathrm{diag}(\mathrm{x}^{-d_1},\dots,\mathrm{x}^{-d_r})$ and $T \in k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})}$ unimodular. Now let $v_1,\dots,v_r \in k(\mathrm{x})$ be such that the coordinate matrix with respect to $\tilde{v}_1,\dots,\tilde{v}_r$ is U. Then

- $\tilde{v}_1,\dots,\tilde{v}_r$ from a $k[\mathrm{x}]$-basis of $\mathfrak{M}_0$.
- The coordinate matrix of $\tilde{v}_1,\dots,\tilde{v}_r$ with respect to $b_1,\dots,b_r$ is M.
- The coordinate matrix of $\mathrm{x}^{d_1}\tilde{v}_1,\dots,\mathrm{x}^{d_r}\tilde{v}_r$ with respect to $b_1,\dots,b_r$ is T. In particular, $\mathrm{x}^{d_1}\tilde{v}_1,\dots,\mathrm{x}^{d_r}\tilde{v}_r$ form a $k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})}$-basis of $\mathfrak{M}_\infty$.

This proves the existence of the integers $d_1 \geq \cdots \geq d_r$ and the basis $v_1,\dots,v_r$. The uniqueness of the integers follows from the previous lemma.

We come to the first item in the proposition. Clearly $\mathrm{x}^{d_i+n}v_1,\dots,\mathrm{x}^{d_r+n}v_r$ forms a $k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})}$-basis of $\mathrm{x}^n \cdot \mathfrak{M}_\infty$.

Now let $a_1,\dots,a_r \in k[x]$. Then $\sum_{i=1}^r a_i v_i = \sum_{i=1}^r a_i x^{-d_i-n} x^{d_i+n} v_i \in \mathrm{x}^n \cdot \mathfrak{M}_\infty$ if and only if $a_i x^{-d_i-n} \in k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})}$, and this is the case if and only if $\mathrm{valdeg}(a_i x^{-d_i-n}) \leq 0$, that is, $\mathrm{valdeg}(a_i) \leq d_i + n$.

Now, we are coming to the second item in the proposition. By the first item $\dim_k(\mathfrak{M}_0 \cap x^n \cdot \mathfrak{M}_\infty) = \sum_{j=1}^{\iota(n)}(d_j + n + 1) = \sum_{j=1}^{\iota(n-1)}(d_j + n + 1)$. This implies that

$$\dim_k(\mathfrak{M}_0 \cap x^n \cdot \mathfrak{M}_\infty) - \dim_k(\mathfrak{M}_0 \cap x^{n-1} \cdot \mathfrak{M}_\infty) = \iota(n-1)\ .$$

Note that this formula also establishes the uniqueness of the integers $d_1 \geq \cdots \geq d_r$. □

Remark 2.23 For matrices A of full column rank, all statements in this subsubsection up to Lemma 2.21 also hold if $k(x)$ is replaced by $k((x^{-1}))$ and $k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})}$ is replaced by $k[[x^{-1}]]$.

The statements of Proposition 2.22 then also hold provided that $\mathfrak{M}_0$ is generated as a $k[\mathrm{x}]$-module by a $k((\mathrm{x}^{-1}))$-basis of V.

Let still V be a finite dimensional $k((\mathrm{x}^{-1}))$-vector space. Then a $k[\mathrm{x}]$-submodule of V which is generated by a $k((\mathrm{x}^{-1}))$-linearly independent set can be regarded as a *non-archimedean lattice* with respect to the non-archimedean absolute value $|| \cdot || = e^{-v_\infty} = e^{\deg}$. Moreover, by Remark 2.18 (and footnote 2) the isometric transformations of V are exactly the endomorphisms given by univariate matrices over $k[[x^{-1}]]$ (with respect to some basis). Section 4 of [Heß01] is written from the point of view of non-archimedean lattices. (But maybe the role of the role of the absolute value $||\cdot||$ is not so clear because Heß works with the degree instead of the absolute value $|| \cdot ||$.)

2.5.3.2 Monomial orders

In this subsubsection we first discuss some general statements concerning monomial orders on $k[\mathrm{x}]^r$ and right-equivalence of matrices in $k[\mathrm{x}]^{r\times s}$. Then we relate these statements with degree-reduced matrices and matrices in Hermite normal form, which we discuss in the next subsubsection. The exposition crucially relies on the theory of Gröbner bases.

We follow the terminology in Chapter 15 of [Eis95]. In particlar, a *monomial* is an element of $k[\mathrm{x}]^r$ of the form $\mathrm{x}^\alpha \underline{e}_i$ for $\alpha \geq 0$ (where $\underline{e}_i$ is the i-th standard basis vector), and a *term* is the product of scalar and a monomial.

We require that a *reduced Gröbner* basis has monic leading terms.[3]

Let us fix a (strict) monomial order $>$ on $k[\mathrm{x}]^r$. All the following definitions and statements are with respect to this fixed monomial order.

Let now $A \in k[\mathrm{x}]^{r\times s}$. Inspired by the usual definitions in the theory of Gröbner bases we define:

Definition 2.24 The matrix A is *top-reduced* if the initial terms of the non-trivial columns of A occur in distinct rows. The matrix A is *reduced* if the non-zero columns of A form a reduced Gröbner basis.

[3]The book [Eis95] is a bit vague on the issue if a reduced Gröbner basis should have monic initial terms. The condition seems not be be required at the end of Section 15.2, but it is part of the definition in Exercise 15.14.

Remark 2.25 The following conditions are equivalent:

- A is top-reduced.
- No non-zero column is top-reducible with respect to the system of the remaining columns (that is, for no $j, \ell = 1, \ldots, s$ with $j \neq \ell$ and $\underline{a}_\ell \neq 0$, $\mathrm{in}_>(\underline{a}_j) \mid \mathrm{in}_>(\underline{a}_\ell)$).
- The non-zero columns of A form a minimal Gröbner basis.[4]

Proposition 2.26 *Let $A \in k[\mathrm{x}]^{r\times s}$. Then there exists a unique matrix $M \in k[\mathrm{x}]^{r\times s}$ which is right-equivalent to A and has the following properties:*

- *M is reduced*
- *the zero-columns are to the left, and the initial terms of the other columns are increasing from left to right, that is, $\mathrm{in}_>(\underline{m}_1) \leq \mathrm{in}_>(\underline{m}_2) \leq \cdots \leq \mathrm{in}_>(\underline{m}_s)$.*

Moreover, the matrix M can be computed from A.

Definition 2.27 We call the matrix M in the proposition the *normal form* of A with respect to $>$, and M itself is called *in normal form* with respect to $>$.

Proof of Proposition 2.26. The existence and uniquenes of the matrix M follows from the general statements that reduced Gröbner bases exist and are unique. As usual, M can be computed with a Buchberger-style reduction algorithm. We review this procedure now in our context.

We first show how one can obtain a top-reduced matrix: Let $A \in k[\mathrm{x}]^{r\times s}$. Assume that there are $j, \ell \in \{1, \ldots, s\}$ with $j \neq \ell$ such that $\underline{a}_j, \underline{a}_\ell \neq 0$ and the initial terms of $\underline{a}_j$ and $\underline{a}_\ell$ occur in the same row, say row i. Let us further assume that the initial term of $\underline{a}_j$ is larger than or equal to the one of $\underline{a}_\ell$. Then one substitutes $\underline{a}_j$ by the S-polynomial $S(\underline{a}_j, \underline{a}_\ell) = a_{i,\ell} \cdot \underline{a}_j - a_{i,j} \cdot \mathrm{x}^{\deg(\underline{a}_{i,j}) - \deg(\underline{a}_{i,\ell})} \cdot \underline{a}_\ell$. Upon iterating this procedure one finally obtains a top-reduced matrix.

Let now $A \in k[\mathrm{x}]^{r\times s}$ be top-reduced. Note again that the columns of A now form a Gröbner basis. One now performs a column swap such that $\underline{a}_1 \leq \underline{a}_2 \leq \cdots \leq \underline{a}_s$. Then one divides each non-zero column by the coefficient of its initial term, such that the columns are monic.

One can now obtain a right-equivalent reduced matrix with the desired additional properties by the following loop:

[4] Note that it is a special property of $k[\mathrm{x}]^r$ that top-reduced systems form a Gröbner basis. The statement is far from true in polynomial rings over k in several variables.

For $j = 2$ to r do
    substitute $\underline{a}_j$ by the (unique) reduction of $\underline{a}_j$ by $\underline{a}_1, \ldots, \underline{a}_{j-1}$.

More precicely, let us consider the j^{th} iteration, so that the matrix $(\underline{a}_1 \cdots \underline{a}_{j-1})$ is already reduced. Then the reduction of $\underline{a}_j$ with respect to $\underline{a}_1, \ldots, \underline{a}_{j-1}$ can be computed as follows:

For $\ell = 1, \ldots, j-1$ such that $\underline{a}_\ell \neq \underline{0}$ we consider the leading term of $\underline{a}_\ell$. Say for some ℓ this leading term is $a\mathrm{x}^\alpha \underline{e}_i$ with $a \in k - \{0\}$ and $\alpha \leq \deg(a_{i,j})$. Then we substitute $\underline{a}_j$ by the reduction of $\underline{a}_j$ by $\underline{a}_\ell$, which is $\underline{a}_j - \mathrm{x}^{\deg(a_{i,j})-\alpha}\underline{a}_\ell$. We stop if for no $\ell = 1, \ldots, j-1$ the condition is satisfied. □

As already stated in the proposition, the normal form of a matrix over $k[\mathrm{x}]$ can be computed. However, the complexity of the reduction algorithm just described depends crucially on the order.

Let us explicitly consider the following two monomial orders for $k[\mathrm{x}]^r$:

$$\mathrm{x}^\alpha \underline{e}_i >_d \mathrm{x}^\beta \underline{e}_j$$

if and only if

$$\alpha > \beta \quad \text{or} \quad \alpha = \beta \text{ and } j > i \,,$$

and

$$\mathrm{x}^\alpha \underline{e}_i >_e \mathrm{x}^\beta \underline{e}_j$$

if and only if

$$j > i \quad \text{or } j = i \text{ and } \alpha > \beta \,.$$

Note that $>_d$ refines the order by *d*egree, and $>_e$ refines the *e*limination order $\underline{e}_n > \underline{e}_{n-1} > \cdots > \underline{e}_1$.

We first discuss order $>_d$. The order $>_e$ is strongly related to Hermite normal forms which are discussed in the next subsubsection.

Note first the following characterization of the initial term: Let $\underline{a} \in k[\mathrm{x}]^r$, $\underline{a} \neq \underline{0}$. Let $a\mathrm{x}^\alpha \underline{e}_i$ be the initial term of $\underline{a}$ (with $a \in k$) with respect to $>_d$. Then for all $j \leq i$, $\deg(a_j) \leq \alpha$ and for all $j > i$, $\deg(a_j) > \alpha$. In particular, $\alpha = \deg(\underline{a})$.

Note now that a matrix of full column rank with is top-reduced with respect to $>_d$ is in particular degree-reduced (in fact, up to a column swap, the matrix of leading coefficients is in column echelon form).

Note further that in the reduction algorithm in the proof of Proposition 2.26 the degrees of the intermediate matrices never exceed the degree of the original matrix. This implies immediately that with the reduction algorithm one can compute a right-equivalent top-reduced matrix in a number of field and bit operations which is polynomially bounded by r, s and $\deg(A)$.

It is a bit more complicated that with this bound on field and bit operations one can also compute a normal form. This is however shown in [MS03, Section 7]. Let us indicate how the final reduction is performed: We want to compute the reduction of $\underline{a}_j$ by $\underline{a}_1, \ldots, \underline{a}_{j-1}$ as at the end of the proof of Proposition 2.26. For this we proceed with reductions by $\underline{a}_j$ by the individual columns $\underline{a}_1, \ldots, \underline{a}_{j-1}$ as indicated in the proof of Proposition 2.26, however with a condition on the index ℓ: Let for $\ell = 1, \ldots, j-1$ such that $\underline{a}_\ell \neq \underline{0}$ i_ℓ be index of the row in which the leading term of $\underline{a}_\ell$ is situated. Then we choose $\ell \in \{1, \ldots, j-1\}$ such that $\deg(a_{i_\ell,j}) - \deg(\underline{a}_\ell) = \deg(a_{i_\ell,j}) - \deg(a_{i,\ell})$ is maximal, and we reduce $\underline{a}_j$ by $\underline{a}_\ell$. With this choice one obtains (cf. [MS03]):

Proposition 2.28 *One can compute the normal form of a matrix $A \in k[\mathrm{x}]^{r\times s}$ with respect to $>_d$ in a number of field and bit operations which is polynomially bounded in r, s and $\deg(A)$.*

We therefore in particular obtain a second efficient reduction algorithm for the computation of a degree-reduced right-equivalent matrix.

Remark 2.29 Top-reduced matrices with respect to $>_d$ are called matrices in *weak Popov form* in [MS03]. The reduction algorithm for the computation of a weak Popov form in [MS03] is the same as the reduction algorithm to obtain a top-reduced matrix presented above. Additionally, a matrix in *Popov form* is exactly a matrix in normal form with respect to $>_d$. Note here again that we implicitly "transpose" all definitions and statements in [MS03]. Monomial orders and the relationship with Gröbner bases are not discussed in [MS03].

In [MS03] it is shown that one can efficiently compute the determinant by a reduction algorithm with respect to $>_d$ (we omit the details):

Proposition 2.30 *Given a matrix $A \in k[\mathrm{x}]^{r\times r}$, one can compute the determinant of A in a number of field and bit operations which is polynomially bounded in r and $\deg(A)$.*

Note that in characteristic 0 (or if the ground field is not too small) one can also obtain this result by computing the determinant of A by interpolation. However, over small finite fields, field extensions might be necessary for this approach, and by current knowledge, no (deterministic) polynomial time algorithm is known for this task. (Such an algorithm exists however under the assumption of the Generalized Riemann Hypothesis ([AL86], [Evd89]).)

We also mention the following consequence of Proposition 2.30.

Proposition 2.31 *Let $A \in k[\mathrm{x}]^{r\times r}$. Then the degree of the adjoint matrix $A^\#$ is $\leq \deg(A)\cdot(r-1)$. Moreover, $A^\#$ can be computed in a number of field and bit operations which is polynomially bounded in r and $\deg(A)$.*

2.5.3.3 Hermite normal forms

We are now coming to *Hermite normal forms* and Hermite normal form bases (*HNF-bases*) of $k[x]$-modules. A very compact definition can be given with the notions of the previous subsubsection:

Definition 2.32 A matrix $M \in k[x]^{r\times s}$ is in *Hermite normal form* if it is in normal form with respect to $>_e$.

Remark 2.33 Explicitly, a matrix $M \in k[x]^{r\times s}$ is in Hermite normal form if and only if there exists some $t \in \mathbb{N}_0$ with $t \leq s$ and a strictly increasing map $\iota : \{t+1, s\} \longrightarrow \{1, \ldots, r\}$ such that

1. the first t columns of M are zero.
2. For $t+1 \leq j \leq s$,
   - $m_{\iota(j),j}$ is monic
   - for all $i > \iota(j)$, $m_{i,j} = 0$
   - for all $\ell > j$, $\deg(m_{\iota(j),j}) > \deg(m_{\iota(j),\ell})$.

Note that these conditions are analogous to the conditions for the Hermite normal form of matrices with integer entries as in [Coh96, Theorem 4.7.3].

Let now M be of full row rank. Then M is in Hermite normal form if and only if it is of the form $M = (O|N)$, where the entries of O are all zero, N is an $r \times r$-matrix and

- N is upper triangular
- for all $i = 1, \ldots, r$, $n_{i,i}$ is monic
- for all $j > i$, $\deg(n_{i,i}) > \deg(n_{i,j})$.

By Proposition 2.26 we already know that given a matrix $A \in k[x]^{r\times s}$, there exists a unique matrix which is right-equivalent to A and in Hermite normal form. This matrix is called the *Hermite normal form* of A.

We are now coming to the efficient computation of Hermite normal forms.

The reduction algorithm in the proof of Proposition 2.26 suffers from the problem that the degrees of the entries of intermediate matrices become extraordinarily large for certain inputs.

In [MS03] the following result is proven. The algorithm is similar to the algorithm for the computation of the determinant and again relies crucially on a reduction algorithm with respect to $>_d$ (!).

Proposition 2.34 *One can compute the Hermite normal form of a matrix $A \in k[\mathrm{x}]^{r\times s}$ of full row rank in a number of field and bit operations which is polynomially bounded in s and* $\deg(A)$.

Let us note that by computing a degree-reduced right-equivalent matrix the problem to compute the Hermite normal form of a matrix of full row rank immediately reduces to the problem to compute the Hermite normal form of a square matrix of full rank, and this is also the first step in the algorithm by Mulders and Storjohann.

There are various other efficient approaches for the computation of Hermite normal forms of square matrices over $k[\mathrm{x}]$ of full rank in the literature. We just mention the algorithm by P. Domich, R. Kanan, L. Trotter. This algorithm relies on computations "modulo the determinant" (see [DKT87]). Again the required number of field and bit operations is polynomially bounded in the size of the matrix and the degree.

We now apply Hermite normal forms to finitely generated $k[\mathrm{x}]$-submodules of finite dimensional $k(\mathrm{x})$-vector spaces.

Definition 2.35 Let V be an r-dimensional $k(\mathrm{x})$-vector space with $r < \infty$, let $\mathfrak{M}$ be a submodule of V of rank r, and let $\mathfrak{N}$ be any finitely generated submodule of V.

- The *denominator* of $\mathfrak{N}$ with respect to $\mathfrak{M}$ is the unique monic generator of the ideal $\{a \in k[\mathrm{x}] \mid a\,\mathfrak{N} \subseteq \mathfrak{M}\}$ of $k[\mathrm{x}]$.

- Let $b_1, \ldots, b_r$ be a $k[\mathrm{x}]$-bases of $\mathfrak{M}$, and let $\mathfrak{N}$ be a finitely generated submodule of V of rank s. Let $\boldsymbol{d}$ be the denominator of $\mathfrak{N}$ with respect to $\mathfrak{M}$. Then the *HNF-basis* of $\mathfrak{N}$ with respect to $b_1, \ldots, b_r$ is the unique $k[\mathrm{x}]$-basis $v_1, \ldots, v_s$ of $\mathfrak{N}$ such that the coordinate matrix of $\boldsymbol{d}v_1, \ldots, \boldsymbol{d}v_s$ with respect to $b_1, \ldots, b_r$ is in Hermite normal form.

We have the following obvious lemma.

Lemma 2.36 *Let V be an r-dimensional $k(\mathrm{x})$-vector space, and let U be a t-dimensional $k(\mathrm{x})$-vector subspace. Let $b_1, \ldots, b_r$ be a basis of V such that $b_1, \ldots, b_t$ is a basis of U. Moreover, let $\mathfrak{N}$ be an r-dimensional $k[\mathrm{x}]$-submodule of V with HNF-basis $v_1, \ldots, v_r$ with respect to $b_1, \ldots, b_r$. Then $v_1, \ldots, v_t$ is the HNF-basis of $\mathfrak{N} \cap U$ with respect to $b_1, \ldots, b_t$.*

Definition 2.37 Let $A \in k(\mathrm{x})^{r\times s}$. Then the *denominator* $\boldsymbol{d}$ of A is the denominator of the submodule of $k(\mathrm{x})^r$ generated by the columns of A. The *numerator matrix* associated to A is the matrix $\boldsymbol{d}A$.

Remark 2.38 Let $g \in k[\mathrm{x}]$ be monic and $A \in k(\mathrm{x})^{r\times s}$, and let h be the gcd of g and all entries of A. Then the denominator of $\frac{1}{g}A$ is $\mathrm{lcm}(\frac{h}{g}, 1)$. In particular, the denominator can be computed in a number of field and bit operations which is polynomially bounded in r, s, $\deg(g)$ and $\deg(A)$ with the Euclidian algorithm. Then the numerator matrix can of course also be computed with this complexity.

Convention 2.39 *If not stated otherwise, the following applies:*

- *We represent a matrix in $k(\mathrm{x})^{r\times s}$ by its denominator and its numerator matrix.*
- *Given a finite dimensional $k(\mathrm{x})$-vector space V with a fixed basis, we represent finite dimensional $k[\mathrm{x}]$-submodules of V by their HNF-basis, which in turn we represent by their coordinate matrix.*

This convention in particular applies to the following computational problem.

Proposition 2.40 *Let $A \in k(\mathrm{x})^{r\times s}$, where the entries are represented in the form $a_{i,j} = \frac{b_{i,j}}{c_{i,j}}$ with $b_{i,j}, c_{i,j} \in k[\mathrm{x}]$. Let M be the coordinate matrix (in HNF-form) of the $k[\mathrm{x}]$-module generated by the columns of A. Then:*

a) The degrees of the denominator of M as well as the numerator matrix of M are polynomially bounded in r, s and the maximum of the degrees of $b_{i,j}$ and $c_{i,j}$ for $i \in \{1, \dots, r\}$ and $j \in \{1, \dots, s\}$.

b) One can compute the HNF-basis of the module generated by the columns of A in a number of field and bit operations which is polynomially bounded in r, s and the maximum of the degrees of $b_{i,j}$ and $c_{i,j}$ for $i \in \{1, \dots, r\}$ and $j \in \{1, \dots, s\}$.

Proof. We first discuss the computational statement: We compute $c := \prod_{i,j} c_{i,j}$ and $\tilde{A} := ((b_{i,j}\frac{c}{c_{i,j}}))_{i,j} \in k[\mathrm{x}]$. Now we perform Hermite reduction on A. Let M be the resulting matrix. Finally, we compute the denominator of $\frac{1}{c}M$ and the numerator matrix.

Let m be the maximum of the degrees of the $b_{i,j}$ and $c_{i,j}$. Then $\deg(c) \le rsm$, $\deg(\tilde{A}) \le \deg(c) + \deg(((b_{i,j}))_{i,j}) \le rsm + m$. All computations can therefore be performed with the claimed complexity by Propositions 2.34 and 2.38.

The statement on the degrees is now also obvious. □

Finally, by imitating the theory of $\mathbb{Z}$-modules, we define:

Definition 2.41 Let $\mathfrak{M}$ be a free finitely generated $k[\mathrm{x}]$-module, and let $\mathfrak{N}$ be a $k[\mathrm{x}]$-submodule of $\mathfrak{M}$ of the same rank. Then we define the $k[\mathrm{x}]$-*index*

$[\mathfrak{M} : \mathfrak{N}]_{k[x]}$ *of* $\mathfrak{N}$ *in* $\mathfrak{M}$ as the unique monic polynomial which is a k-multiple of the determinant of the coordinate matrix of a basis of $\mathfrak{N}$ with respect to a basis of $\mathfrak{M}$.

Remark 2.42 The $k[x]$-index $[\mathfrak{M} : \mathfrak{N}]_{k[x]}$ is equal to the invariant $\chi(\mathfrak{M}/\mathfrak{N})$ defined in [Ser79, I, §5]. This invariant can be defined in a more general context, and in particular if A is a finitely generated free abelian group and B is a subgroup of A of the same rank, then $\chi(A/B) = [A : B]$.

Remark and Definition 2.43 Let $\mathfrak{M}$ and $\mathfrak{N}$ be as above, let $b_1, \ldots, b_r$ be a $k[x]$-basis of $\mathfrak{M}$, and let M be the coordinate matrix of the HNF-basis of $\mathfrak{N}$ with respect to $b_1, \ldots, b_r$. Then the system of elements $x^\alpha b_j$ with $\alpha < \deg(m_{j,j})$, $j = 1, \ldots, r$ is k-linearly independent. Let C be the vector space spanned by it. Then C is a direct complement of the k-vector space $\mathfrak{N}$ in $\mathfrak{M}$; we refer to C as the *canonical complement* of $\mathfrak{N}$ in $\mathfrak{M}$ and to the basis just defined as the *canonical complementary system* (with respect to $b_1, \ldots, b_r$). The projection $\mathfrak{M} \longrightarrow \mathfrak{M}/\mathfrak{N}$ induces an isomorphism of k-vector spaces $C \longrightarrow \mathfrak{M}/\mathfrak{N}$. We call the image of the canonical system the *canonical basis* of $\mathfrak{M}/\mathfrak{N}$ and the inverse of the isomorphism the *canonical lift* of $\mathfrak{M}/\mathfrak{N}$ to $\mathfrak{M}$ with respect to $b_1, \ldots, b_r$.

Remark 2.44 Let $\mathfrak{M} = k[x]^r$ and $b_i = \underline{e}_i$. Then the basis elements $x^\alpha b_j = x^\alpha \underline{e}_j$ in Remark and Definition 2.43 are the monomials which do not lie in $\mathrm{in}_{>_e}(\mathfrak{N})$. The statements are then a special case of a theorem of Macaulay's (cf. [Eis95, Theorem 15.3]).

Remark 2.45 The $k[x]$-index $[\mathfrak{M} : \mathfrak{N}]_{k[x]}$ is in particular equal to the determinant (= the product of diagonal entries) of the HNF-basis of $\mathfrak{N}$ with respect to some basis of $\mathfrak{M}$. Note also that $\deg([\mathfrak{M} : \mathfrak{N}]_{k[x]}) = \dim_k(\mathfrak{M}/\mathfrak{N})$.

Let now $d_1 | \cdots | d_r$ be the (monic) elementary divisors of $\mathfrak{N}$ in $\mathfrak{M}$. Then $[\mathfrak{M} : \mathfrak{N}]_{k[x]} = d_1 \cdots d_r$. On the other hand, the denominator of $\mathfrak{M}$ with respect to $\mathfrak{N}$ is equal to d_r. In particular, the denominator divides the $k[x]$-index.

2.5.3.4 Modules over extension fields

Let us as a generalization now consider the previous definitions and results over finite extension fields $\lambda|k$. All the following computations still take place on a generic field RAM, instantiated with the field k.

Let $\lambda|k$ be a finite field extension, and let $a_1, \ldots, a_{[\lambda:k]}$ be a basis of $\lambda|k$. We assume that a multiplication table of of λ with respect to this basis is known. (As a special case, $\lambda|k$ might be given by an irreducible polynomial,

and $a_1, \ldots, a_{[\lambda:k]}$ might be a polynomial basis.) The elements of λ are then represented via their coordinate vectors with respect to this basis.

Note that by Proposition 1.58 all earlier computational results from this subsection can be transferred to the more general setting of this subsubsection. The only difference is that now the required number of field and bit operations (in k) and bit operations is additionally polynomially bounded in $[\lambda : k]$.

Note that $a_1, \ldots, a_{[\lambda:k]}$ is also a basis of $\lambda[\mathrm{x}]$ over $k[\mathrm{x}]$ and of $\lambda(\mathrm{x})$ over $k(\mathrm{x})$. Given a polynomial $g(\mathrm{x}) \in \lambda[\mathrm{x}]$, it is computationally trivial to "expand" g with respect to $a_1, \ldots a_{[\lambda:k]}$, that is, to compute $g_1, \ldots, g_{[\lambda:k]} \in k[\mathrm{x}]$ with

$$g = \sum_{j=1}^{[\lambda:k]} g_j \, a_j \ . \tag{2.1}$$

Let now two polynomials $g(\mathrm{x}), h(\mathrm{x}) \in \lambda[\mathrm{x}]$ with $h(\mathrm{x}) \neq 0$ be given. We now want to compute $g_j \in k(\mathrm{x})$ with $\frac{g}{h} = \sum_{j=1}^{[\lambda:k]} g_j \, a_j$. If $h(\mathrm{x}) \in k[\mathrm{x}]$ this is again straightforward. If this is not the case, one easily reduce to this case by the formula

$$\frac{g}{h} = \frac{g \cdot N(h)/h}{N(h)} \ ,$$

where $N(h)$ is the norm of h with respect to $\lambda(\mathrm{x})|k(\mathrm{x})$. The required number of field and bit operations is then polynomially bounded in $\deg(g)$ and $\deg(h)$.

Let now V be an r-dimensional $\lambda(\mathrm{x})$-vector space with $r < \infty$, and let $\mathfrak{M}$ be a $\lambda[\mathrm{x}]$-submodule of V with a $\lambda[\mathrm{x}]$-basis $b_1, \ldots, b_r$. Now the elements

$$\widehat{b}_{i+(j-1)r} := b_i \, a_j \quad \text{for } i = 1, \ldots, r, \ j = 1, \ldots, [\lambda : k] \, . \tag{2.2}$$

form a $k[\mathrm{x}]$-basis of $\mathfrak{M}$.

Note that if $v \in V$ has coordinate vector $\underline{c} \in \lambda(\mathrm{x})^r$ with respect to $b_1, \ldots, b_r$ and if we write $c_i = \sum_{j=1}^{[\lambda:k]} c_{i,j} \, a_j$ with $c_{i,j} \in k(\mathrm{x})$, then the vector $\underline{d} \in k(\mathrm{x})^{r \cdot [\lambda:k]}$ defined by

$$d_{i+(j-1)r} := c_{i,j} \text{ for } i = 1, \ldots, r, \ j = 1, \ldots, [\lambda : k]$$

is the coordinate vector of v with respect to $\widehat{b}_1, \ldots, \widehat{b}_{r \cdot [\lambda:k]}$.

Note that at this point we have two natural representations of $\lambda[\mathrm{x}]$-submodules of V of rank r: First we can represent them by HNF-bases with respect to $b_1, \ldots, b_r$ and second, we can represent them by HNF-bases with respect to $\widehat{b}_1, \ldots, \widehat{b}_{r \cdot [\lambda:k]}$. Note that in the former case the coefficients lie in $\lambda[\mathrm{x}]$, and in the latter case the coefficients lie in $k[\mathrm{x}]$. Note in particular that in the former case the coefficients of the numerator matrix as well as

the denominator lie in $\lambda[\mathrm{x}]$, and in the latter case the coefficients of the numerator matrix as well as the denominator lie in $k[\mathrm{x}]$.

With the above remarks and the previous statements on the computation of Hermite normal forms, one obtains:

Proposition 2.46 *Let $\mathfrak{N}$ be a finitely generated $\lambda[\mathrm{x}]$-submodule of V with denominator $\boldsymbol{d}$ with respect to $\mathfrak{M}$. Then one can change between the representation of $\mathfrak{N}$ by the HNF-basis with respect to $b_1, \ldots, b_r$ and the representation of $\mathfrak{N}$ by the HNF-basis with respect to $\widehat{b}_1, \ldots, \widehat{b}_{r\cdot[\lambda:k]}$ in a number of field operations in k and bit operations which is polynomially bounded in r, $[\lambda : k]$, $\deg([\mathfrak{M} : \boldsymbol{d} \cdot \mathfrak{N}]_{\lambda[\mathrm{x}]})$ and $\deg(\boldsymbol{d})$.*

2.5.4 The ideal representation

2.5.4.1 The idea and some notation

One can use *ideal theory in function fields* to represent closed points and divisors. The ideal theoretic approach is crucial because it leads to an easy and efficient algorithm to compute Riemann-Roch spaces (Heß' algorithm).

In the ideal theoretic approach which we are going to describe now it is irrelevant whether the extension $k(\mathcal{C})|k$ is regular or not. Therefore all the considerations from now on until including subsubsection 2.5.4.4 hold not only if $\mathcal{C}$ is a curve over k but also if $\mathcal{C}$ is merely a *irreducible* (rather than *geometrically irreducible*) smooth, proper 1-dimensional k-scheme.

As stated in Section 2.2, we assume that $\mathcal{C}_{pm}$ is not equal to $V(Z) \subset \mathbb{P}^2_k$. Also following the general terminology and notation, we set $x := \frac{X}{Z}, y := \frac{Y}{Z} \in k(\mathbb{P}^2)$ and let $x_{|\mathcal{C}}$ resp. $y_{|\mathcal{C}} \in k(\mathcal{C})$ be the pull-backs of the rational functions x resp. y to $\mathcal{C}$. Let $f(x, y) := \frac{F(X,Y,Z)}{Z^{\deg(F)}} = F(x, y, 1) \in k[x, y]$.

Note that one of $x_{|\mathcal{C}}$ and $y_{|\mathcal{C}}$ is a separating element of the function field $k(\mathcal{C})$. (Otherwise $f(x, y)$ would be a p-th power.) Let us assume wlog. that $x_{|\mathcal{C}}$ is a separating element.

Let us consider the covering $x_{|\mathcal{C}} : \mathcal{C} \longrightarrow \mathbb{P}^1_k$ of curves as well as the corresponding separable field extension $k(\mathcal{C})|k(\mathrm{x})$ given by $\mathrm{x} \mapsto x_{|\mathcal{C}}$.

Convention 2.47 *In the following function field theoretic considerations, we identify $k(\mathrm{x})$ with its image in $k(\mathcal{C})$. We do however not perform this identification if we consider the elements of the function fields as functions on the curves.*

Notation 2.48 We set $r := [k(\mathcal{C}) : k(\mathrm{x})] = \deg(x_{|\mathcal{C}} : \mathcal{C} \longrightarrow \mathbb{P}^1_k)$, the extension degree of $k(\mathcal{C})$ over $k(\mathrm{x})$.

Note that we have a canonical isomorphism $k(\mathcal{C}) = k(x_{|\mathcal{C}}, y_{|\mathcal{C}}) \simeq k(\mathrm{x})[y]/(f(\mathrm{x}, y))$, and r is equal to the degree in y of $f(\mathrm{x}, y)$.

We now consider the subrings $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$ and $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})_\infty$ of $k(\mathcal{C})$; these rings are the integral closures of $k[\mathrm{x}]$ resp. $k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})}$ in $k(\mathcal{C})$. They are called *finite* and *infinite order* of $k(\mathcal{C})$ (with respect to $x_{|\mathcal{C}}$).

Now the closed points of $\mathcal{C}$ correspond to the places of $k(\mathcal{C})$ and these in turn correspond to maximal ideals of $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$ as well as $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})_\infty$.

The main idea for the ideal representation of points is to compute bases of these orders and to represent the points of $\mathcal{C}$ by "nice" $k[\mathrm{x}]$- respectively $k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})}$-bases of the corresponding maximal ideals. In order to have a better control over the size of this representation, we consider however prime ideals of $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\frac{1}{\mathrm{x}}(\mathbb{A}^1_k))$ with support "at infinity" instead of prime ideals in $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})_\infty$.

Now divisors can be represented in two ways: The first way is to take the representation of points as a basis for a *free representation*: The divisors are represented by their support and the coefficient vector, and the points are represented by maximal ideals of the two orders.

Another way to represent divisors is to consider the isomorphism

$$\begin{aligned}\mathrm{Div}(\mathcal{C}) \xrightarrow{\sim} \mathrm{Div}(((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)) \times \mathrm{Div}((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})_\infty) \\ \xrightarrow{\sim} I(((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)) \times I(((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})_\infty)\ ,\end{aligned} \tag{2.3}$$

where the first isomorphism is induced by pull-back and the second isomorphism is induced by the canonical isomorphisms between the divisor and ideal groups. Via this isomorphism every divisor corresponds to a pair of fractional ideals, a "finite" and an "infinity" ideal. To represent the "infinite" ideal, one furthermore applies the canonical inclusion $I(((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})_\infty) \hookrightarrow I(((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\frac{1}{\mathrm{x}}(\mathbb{A}^1_k)))$. We call this representation of divisors the *joint ideal representation*.

Notation 2.49 Let D be a divisor on $\mathcal{C}$. Then the associated "finite" fractional ideal in $I(((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k))$ is denoted by $I^{\mathrm{fin}}(D)$. The associated "infinite" fractional ideal in $I(((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\frac{1}{\mathrm{x}}(\mathbb{A}^1_k)))$ is denoted by $I^\infty(D)$, and the corresponding ideal in $I(((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})_\infty)$ (the localization of $I^\infty(D)$ at ∞) by $(I^\infty(D))_\infty$.

Note here that $I^\infty(D)$ by definition only has support "at infinity", and $(I^\infty(D))_\infty$ is the ideal generated by $I^\infty(D)$ inside $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})_\infty$.

We now need to describe how we represent the fractional ideals of the orders $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$ and $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\frac{1}{\mathrm{x}}(\mathbb{A}^1_k))$.

One can easily adapt usual methods to represent and manipulate ideals in number fields to the function field theoretic setting. In fact, the exposition

in this subsection relies on the Sections 2.4, 4.7, 4.8, 6.1 and 6.2 of the book [Coh96] by H. Cohen.

2.5.4.2 On the maximal orders

Let us fix the following notation:

Notation 2.50

- Let $f(\mathrm{x}, y) = \sum_{i=0}^{r} a_i(\mathrm{x}) y^i$ with $a_i(\mathrm{x}) \in k[\mathrm{x}]$, let $\tilde{y} := a_r(x) \cdot y$, and let $\tilde{f}(\mathrm{x}, \tilde{y}) := a_r(\mathrm{x})^{r-1} f(\mathrm{x}, \frac{\tilde{y}}{a_r(\mathrm{x})}) \in k[\mathrm{x}][\tilde{y}]$.
- Let $c := \max_{i=0}^{r} \lceil \frac{\deg(a_i(\mathrm{x}))}{r-i} \rceil$, let $\tilde{\tilde{y}} := \frac{\tilde{y}}{\mathrm{x}^c}$, and let $\tilde{\tilde{f}}(\mathrm{x}, \tilde{\tilde{y}}) := \frac{1}{\mathrm{x}^{rc}} \tilde{f}(\mathrm{x}, \mathrm{x}^c \tilde{\tilde{y}}) \in k[\frac{1}{\mathrm{x}}][\tilde{\tilde{y}}]$

Now $\tilde{y}_{|\mathcal{C}} \in k(\mathcal{C})$ is integral over $k[\mathrm{x}]$, as it is a root of the monic polynomial $\tilde{f}(\mathrm{x}, \tilde{y}) \in k[\mathrm{x}][\tilde{y}]$ over $k[\mathrm{x}]$. Moreover, $\tilde{\tilde{y}}$ is integral over $k[\frac{1}{\mathrm{x}}]$ as it is the root of the monic polynomial $\tilde{\tilde{f}}(\mathrm{x}, \tilde{\tilde{y}}) \in k[\frac{1}{\mathrm{x}}][\tilde{\tilde{y}}]$.

We now first consider the finite order. After this we briefly mention how the following considerations can be adapted to the infinite order.

Notation 2.51 Let $\delta(\mathrm{x}) \in k[\mathrm{x}]$ be the discriminant of $((x_{|\mathcal{C}})_* \mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k) | k[\mathrm{x}]$. Similarly, let $\mathrm{disc}(\tilde{f})$ be the discriminant of $\tilde{f}$ as a polynomial in $\tilde{y}$. Let $\boldsymbol{i} := [((x_{|\mathcal{C}})_* \mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k) : k[\mathrm{x}][y]/(\tilde{f})]_{k[\mathrm{x}]}$.

Notation 2.52 Let $w_1, \ldots, w_r$ be the HNF-basis of $((x_{|\mathcal{C}})_* \mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k)$ with respect to the $k(\mathrm{x})$-basis $1, \tilde{y}_{|\mathcal{C}}, \ldots, \tilde{y}_{|\mathcal{C}}^{r-1}$.

Proposition 2.53

a) *i)* $\mathrm{disc}(\tilde{f}) = \delta\, \boldsymbol{i}^2$.

ii) $\deg(\mathrm{disc}(\tilde{f}))$ *(and in particular* $\deg(\boldsymbol{i})$*) is polynomially bounded in* d.

iii) $w_1 = 1$.

iv) *The denominator* $\boldsymbol{d}$ *of* $((x_{|\mathcal{C}})_* \mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k)$ *with respect to* $k[\mathrm{x}][y]/(\tilde{f})$ *divides* $\boldsymbol{i}$.

v) *Let* $W \in k[\mathrm{x}]^{r \times r}$ *be the coordinate matrix of* $\boldsymbol{d}w_1, \ldots, \boldsymbol{d}w_r$ *with respect to* $1, \tilde{y}_{|\mathcal{C}}, \ldots, \tilde{y}_{|\mathcal{C}}^{r-1}$. *Then* $\deg(W) = \deg(\boldsymbol{d})$.

b) *One can compute the HNF-basis of* $((x_{|\mathcal{C}})_* \mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k)$ *with respect to* $1, \tilde{y}_{|\mathcal{C}}, \ldots, \tilde{y}_{|\mathcal{C}}^{r-1}$ *with a deterministic algorithm in a number of field and bit operations which is polynomially bounded in* d.

Proof of a).

i) Let more generally $v_1, \dots, v_r$ be any basis of $k(\mathcal{C})|k(\mathrm{x})$. Then we have $\mathrm{disc}(v_1, \dots, v_r) = \det(\mathrm{Tr}(v_i v_j)_{i,j})$. Now let A be the coordinate matrix of $v_1, \dots, v_r$ with respect to another basis $\tilde{v}_1, \dots, \tilde{v}_r$. Then clearly $\mathrm{disc}(v_1, \dots, v_r) = \det(A)^2 \,\mathrm{disc}(\tilde{v}_1, \dots, \tilde{v}_r)$.

ii) This follows for example from $\mathrm{disc}(\tilde{f}) = (-1)^{\frac{1}{2}n(n-1)} \cdot \mathrm{Res}(\tilde{f}, \frac{\mathrm{d}\tilde{f}}{\mathrm{d}\tilde{y}})$.

iii) Clearly $((x_{|\mathcal{C}})_* \mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k) \cap k(\mathrm{x}) = k[\mathrm{x}]$. This implies that $w_1 = 1$ with Lemma 2.36.

iv) This is a special case of Remark 2.45.

v) We have $\deg(W) \geq \boldsymbol{d}$ because $w_1 = 1$ and therefore the first column of W is $\boldsymbol{d}\underline{e}_1$. Moreover, for all $i = 1, \dots, r$, $\tilde{y}^{i-1}_{|\mathcal{C}} \in ((x_{|\mathcal{C}})_* \mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k)$, thus $\boldsymbol{d}\underline{e}_i$ is in the span of the columns of W. As W is a non-singular matrix in upper diagonal form, this implies that the degrees of the diagonal elements of W are $\leq \deg(\boldsymbol{d})$. As W is in Hermite normal form, this implies that $\deg(W) \leq \deg(\boldsymbol{d})$.

An algorithm for Part b) is presented in Section 2.7.

Convention 2.54 *In the following we assume that the data in the proposition and the corresponding data for the infinite order $((x_{|\mathcal{C}})_* \mathcal{O}_{\mathcal{C}})(\frac{1}{\mathrm{x}}(\mathbb{A}^1_k))$ have been computed.*

(This means that the following algorithms take as input a curve represented by a plane model as well as data as in Proposition 2.53 as well as the corresponding data for the infinite order and some further input which is described in detail below. All complexity related statements below are relative to this input.)

We now represent fractional $((x_{|\mathcal{C}})_* \mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k)$-ideals as described in Convention 2.39:

Convention 2.55 *We always represent fractional $((x_{|\mathcal{C}})_* \mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k)$-ideals by their HNF-bases with respect to $w_1, \dots, w_r$, and the HNF-bases are represented by the denominator and the coordinate matrix in $k[\mathrm{x}]^{r \times r}$ in Hermite normal form.*

The following lemmata complement Proposition 2.53.

Lemma 2.56 *The $k[\mathrm{x}]$-index $\boldsymbol{i}$ as well as the discriminant δ can be computed in a number of field and bit operations which is polynomially bounded in d.*

Proof. We have $\boldsymbol{i} = \det(\boldsymbol{d}W^{-1}) = \boldsymbol{d}^r \det(W)^{-1}$ and $\det(W) = w_{1,1} \cdots w_{r,r}$. Moreover, $\delta = \frac{\mathrm{disc}(\tilde{f})}{\boldsymbol{i}^2}$. □

Lemma 2.57 *The degree of the coordinate vector of $\tilde{y}_{|\mathcal{C}}$ with respect to $w_1, \ldots, w_r$ is $\leq d$. The coordinate vector can be computed in a number of field and bit operations which is polynomially bounded in d.*

Proof. By the shape of the matrix W, $\tilde{y}_{|\mathcal{C}}$ is a $k[\mathrm{x}]$-linear combination of $w_1 = 1$ and w_2; let $\tilde{y}_{|\mathcal{C}} = a + bw_2$ with $a, b \in k[\mathrm{x}]$. Let $W = ((w_{i,j}))_{i,j}$. Then

$$\begin{pmatrix} 0 \\ \boldsymbol{d} \end{pmatrix} = a \begin{pmatrix} \boldsymbol{d} \\ 0 \end{pmatrix} + b \begin{pmatrix} w_{1,2} \\ w_{2,2} \end{pmatrix} .$$

The second row implies that $\deg(b) \leq \deg(\boldsymbol{d})$. As $\deg(w_{1,2}) < \deg(\boldsymbol{d})$ this implies together with the first row that $\deg(a) < \deg(\boldsymbol{d})$.

The computational result is obvious. □

This lemma and Proposition 2.53 iv) and v) imply:

Lemma 2.58 *The degree of the multiplication table of $w_1, \ldots, w_r$ (which is an element of $k[\mathrm{x}]^{r \times r \times r}$) is polynomially bounded in d. The table can be computed with a number of field and bit operations which is polynomially bounded in d.*

On the infinite order

Let us fix the following definition: Let for some $a \in k[\frac{1}{\mathrm{x}}]$ $\deg_{\frac{1}{\mathrm{x}}}(a)$ be the degree of a in $\frac{1}{\mathrm{x}}$. Analogously to the definitions for matrices in $k[\mathrm{x}]$, we define for a matrix M over $k[\frac{1}{\mathrm{x}}]$ the *degree of M in $\frac{1}{\mathrm{x}}$*, $\deg_{\frac{1}{\mathrm{x}}}(M)$, as the maximum of the degrees in $\frac{1}{\mathrm{x}}$ of the entries of M.

Proposition 2.53 holds mutatis mutandis also for the infinite order: One first has to substitute x by $\frac{1}{\mathrm{x}}$, $\tilde{y}$ by $\tilde{\tilde{y}}$, and instead of considering $k[\mathrm{x}]$-bases one has to consider $k[\frac{1}{\mathrm{x}}]$-bases. One also has to substitute the usual degree by the degree in $\frac{1}{\mathrm{x}}$. Then one has to substitute $\tilde{f}$ by $\tilde{\tilde{f}}$, the HNF basis $w_1, \ldots, w_r$ by the HNF basis of the infinite order with respect to $1, \tilde{\tilde{y}}_{|\mathcal{C}}, \ldots, \tilde{\tilde{y}}_{|\mathcal{C}}$, and one has to adopt the definitions of the index, the denominator and the matrix W accordingly. The degree d of $F(X, Y, Z)$ is however left as it is.

Note here that indeed $\deg_{\frac{1}{\mathrm{x}}}(\mathrm{disc}(\tilde{\tilde{f}}))$ is polynomially bounded in d as $\deg_{\frac{1}{\mathrm{x}}}(\tilde{\tilde{f}})$ is polynomially bounded in d. This implies that the degree of the coordinate matrix in v) as well as the running time in field and bit operations in b) are polynomially bounded in d.

2.5.4.3 Ideal arithmetic

An important aspect of the joint ideal representation of divisors is that one can apply the usual ideal arithmetic. From a computational point of

view, one can easily adapt the algorithms described in [Coh96, 4.7, 4.8] from the number field theoretic to the function field theoretic setting. A crucial ingredient is here the computation of Hermite normal forms.

In the following, we concentrate on fractional ideals of the finite order $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$ and its fractional ideals. Similar definitions and results hold for fractional ideals of $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\frac{1}{\mathrm{x}}(\mathbb{A}^1_k))$.

Note again that in all the following computational statements we assume that the fractional ideals of the finite order are given by HNF-bases with respect to $w_1, \ldots, w_r$.

Terminology 2.59 We refer to a non-trivial ideal of $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$ as a *proper ideal*.[5]

Terminology 2.60 By the *norm* $\mathrm{N}(I)$ of a fractional ideal I of $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$ we mean the norm of I with respect to $k(\mathrm{x})$.

Lemma 2.61 *Let I be a proper ideal of $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$, and let $M = (m_{i,j})_{i,j} \in k(\mathrm{x})^{r\times r}$ be a coordinate matrix of I with respect to $w_1, \ldots, w_r$ (not necessarily in Hermite normal form). Then $\mathrm{N}(I) = (\det(M))$. Moreover, if M is in Hermite normal form, $I \cap k[\mathrm{x}] = (m_{1,1})$.*

Proof. Let $I = \mathfrak{P}$ be a prime ideal, let $\mathfrak{p} := I \cap k[\mathrm{x}]$, let $\mathfrak{p} = (P)$ with P monic, and let f be the residue degree. Then the residue field $k(\mathfrak{P})$ is an f-dimensional $k(\mathfrak{p})$-vector space. This means that as $k[\mathrm{x}]$-module $k(\mathfrak{P})$ is isomorphic to $(k[\mathrm{x}]/\mathfrak{p})^f$. Thus if S is the Smith normal form of M, the non-trivial entries of S are either 1 or P, and there are exactly f entries equal to P. In particular, the determinant of S is P^f. This implies that $(\det(M)) = (P^f) = \mathfrak{p}^f = \mathrm{N}(I)$.

For general ideals the statement follows because both the norm and the determinant are multiplicative.

The last statement follows from $w_1 = 1$. □

Definition 2.62 The *degree* of a proper $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$-ideal I is $\deg(I) := \dim_k(((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)/I)$.

Let now I be a fractional $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$-ideal. Then we define I_+ and I_- as the unique proper ideals with $I = I_+\, I_-^{-1}$.

The *degree* of I is $\deg(I) := \deg(I_+) - \deg(I_-)$ and the *height* of I is $\max\{\deg(I_+), \deg(I_-)\}$.

[5] Note that this terminology is at odds the usual terminology in commutative algebra where one calls an ideal not equal to the full ring a proper ideal.

Remark 2.63 Note that $\deg(I) = \deg(\mathrm{N}(I))$. In particular, if I is given by a coordinate matrix $M \in k(\mathrm{x})$ with respect to $w_1, \ldots, w_r$ then $\deg(I) = \deg(\det(M))$.

The following lemma is obvious:

Lemma 2.64 *Let I be a fractional $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$-ideal. Then the denominator of I divides $\mathrm{N}(I_-)$. In particular, the degree of the denominator $\boldsymbol{d}$ of I is bounded by $\mathrm{ht}(I)$. Moreover, the degree of the proper ideal $\boldsymbol{d}I$ is bounded by $(r+1)\cdot\mathrm{ht}(I)$. In particular, let $M \in k[\mathrm{x}]^{r\times r}$ be the coordinate matrix of $\boldsymbol{d}I$ in Hermite normal form. Then $\deg(M) \leq (r+1)\cdot\mathrm{ht}(I)$.*

In analogy of the sum of ideals we define:

Definition 2.65 Let I a and J be two fractional ideals of $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$. Then the *sum* of I and J the $k[\mathrm{x}]$-submodule of $k(\mathcal{C})$ generated by I and J. We denote the sum of I and J by $I + J = (I, J)$.

The sum of fractional ideals should not be confused with their product. Note also the following formula for fractional ideals I and J of $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$:

$$I \cdot J = (I \cap J) \cdot (I + J) \tag{2.4}$$

Proposition 2.66 *Given two fractional ideals I, J of $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$, one can compute the product $I \cdot J$ and the sum $I + J = (I, J)$ in a number of field and bit operations which is polynomially bounded in d, $\mathrm{ht}(I)$ and $\mathrm{ht}(J)$.*

Proof. If I is generated by $g_1, \ldots, g_r$ and J is generated by $h_1, \ldots, h_r$, then $I \cdot J$ is generated by $\{g_i h_j | i, j = 1, \ldots, r\}$ and $I + J$ is generated by $g_1, \ldots, g_r, h_1, \ldots, h_r$. The result follows from Lemma 2.64 and Proposition 2.40. □

Recall that the *codifferent* of $k(\mathcal{C})$ (or $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$) with respect to $k[\mathrm{x}]$ is defined as

$$\mathfrak{C} := \{ g \in k(\mathcal{C}) \mid \mathrm{Tr}(g \cdot ((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)) \subseteq k[\mathrm{x}] \} .$$

The *different* $\mathfrak{D}$, which is a proper ideal, is then defined as the inverse of this fractional ideal.

Lemma 2.67 *Let I be a fractional ideal of $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$, and let $h \in k(\mathcal{C})$. Then $h \in I^{-1}\mathfrak{C}$ if and only if $\mathrm{Tr}(hI) \subseteq k[\mathrm{x}]$.*

Proof. Note that $I^{-1}\mathfrak{C} = \{h \in k(\mathcal{C}) \mid hI \subseteq \mathfrak{C}\}$.

Let first $h \in I^{-1}\mathfrak{C}$. Then $hI \subseteq \mathfrak{C}$. Therefore $\mathrm{Tr}(hI) = \mathrm{Tr}(hI \cdot 1) \subseteq k[\mathrm{x}]$.

Now let $\mathrm{Tr}(hI) \subseteq k[\mathrm{x}]$. Then $\mathrm{Tr}(hI \cdot ((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)) \subseteq k[\mathrm{x}]$, therefore $hI \subseteq \mathfrak{C}$. Thus $h \in I^{-1}\mathfrak{C}$. □

The following lemma and its proof are analogous to [Coh96, Proposition 4.8.19].

Lemma 2.68 *Let I be a fractional ideal of $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k))$, and let $M = (m_{i,j})_{i,j} \in k(\mathrm{x})^{r\times r}$ be a coordinate matrix of I with respect to $w_1, \ldots, w_r$. Let T be the $r \times r$-matrix with entries $t_{i,j} = \mathrm{Tr}(w_i w_j)$. Then T is regular. Let $h_1, \ldots, h_r \in k(\mathcal{C})$ be the elements whose coordinate matrix with respect to $w_1, \ldots, w_r$ is $(M^tT)^{-1}$. Then $h_1, \ldots, h_r$ form a $k[\mathrm{x}]$-basis of $I^{-1}\mathfrak{C}$.*

Proof. Let $g_1, \ldots, g_r$ be the $k[\mathrm{x}]$-basis of I whose coordinate matrix with respect to $w_1, \ldots, w_r$ is M.

The matrix T is regular as the extension $k(\mathcal{C})|k(\mathrm{x})$ is separable by assumption. (It is here that we use this condition for the first time.)

Let $N := M^tT$. Then $n_{i,j} = \sum_{\ell=1}^r m_{\ell,i}\mathrm{Tr}(w_\ell w_j) = \mathrm{Tr}(\sum_{\ell=1}^r m_{\ell,i} w_\ell\, w_j) = \mathrm{Tr}(g_i w_j)$.

Let now $\underline{h} \in k(\mathrm{x})^r$, and let $h := \sum_{j=1}^r h_j w_j$. By the previous lemma, $h \in I^{-1}\mathfrak{C}$ if and only if $\mathrm{Tr}(hI) \subseteq k[\mathrm{x}]$. This is equivalent to: $\forall i = 1, \ldots, r :$ $\mathrm{Tr}(g_i h) \subseteq k[\mathrm{x}]$, that is, $\forall i = 1, \ldots, r : \sum_{j=1}^r h_j \mathrm{Tr}(g_i w_j) \subseteq k[\mathrm{x}]$, that is, $N\underline{h} \in k[\mathrm{x}]^r$. This in turn is equivalent to $\underline{h}$ being in the $k[\mathrm{x}]$-module generated by the columns of N^{-1}. □

This lemma immediately implies the following computational result:

Proposition 2.69

a) One can compute the codifferent $\mathfrak{C}$ and the different $\mathfrak{D}$ in a number of field and bit operations which is polynomially bounded in d.

b) One can compute the inverse of a fractional ideal I of $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$ in a number of field and bit operations which is polynomially bounded in d and $\mathrm{ht}(I)$.

Proof. By the previous lemma and Proposition 2.34 on the computation of the Hermite normal form, the following holds: Given a fractional ideal I, one can compute $I^{-1}\mathfrak{C}$ in a number of field and bit operations which is polynomially bounded in d and $\mathrm{ht}(I)$. In particular, one can compute $\mathfrak{C}$ in a number of field and bit operations which is polynomially bounded in d. By the formula $\mathfrak{D} = (\mathfrak{C}^2)^{-1}\mathfrak{C}$ one can also compute $\mathfrak{D}$ in a number of field and bit operations which is polynomially bounded in d. Now point b) follows from the formula $I^{-1} = I^{-1}\mathfrak{C}\mathfrak{D}$. □

This result and Equation (2.4) imply:

Proposition 2.70 *Given two fractional ideals I, J of $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$, one can compute the intersection $I \cap J$ in a number of field and bit operations which is polynomially bounded in d, $\mathrm{ht}(I)$ and $\mathrm{ht}(J)$.*

We also have:

Lemma 2.71 *Given a fractional ideal I, one can compute the proper ideals I_+ and I_- with $I = I_+ I_-^{-1}$ in a number of field and bit operations which is polynomially bounded in d and $\mathrm{ht}(I)$.*

Proof. One can for example use the formulae $I_+ = I \cap (1)$ and $I_-^{-1} = I + (1)$.□

For the following proposition, recall again that we represent fractional ideals by HNF-bases with respect to the basis $w_1, \dots w_r$ of $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$.

Proposition 2.72 *Let $s \in \mathbb{N}$, and let $a_{i,j}, b_{i,j} \in k[\mathrm{x}]$ for $i = 1, \dots, r, j = 1, \dots, s$. Let $g_j := \sum_{i=1}^r \frac{a_{i,j}}{b_{i,j}} w_i \in k(\mathcal{C})$ for $j = 1, \dots, s$. Then the height of the fractional ideal $(g_1, \dots, g_s)$ is polynomially bounded in r, s and the maximum of the degrees of the $a_{i,j}$ and $b_{i,j}$. Moreover, one can compute this ideal in a number of field and bit operations which is polynomially bounded in d, s and the maximum of the degrees of the $a_{i,j}$ and $b_{i,j}$.*

Proof. Note that the elements $y^\ell g_j$ for $\ell = 0, \dots, r-1$ and $j = 1, \dots, s$ generate the ideal as a $k[\mathrm{x}]$-module.

The computational statement now follows from Proposition 2.40.

Let now $\frac{1}{\boldsymbol{d}} M$ be the coordinate matrix of the HNF-basis of the ideal, where $\boldsymbol{d}$ is the denominator. Then again by Proposition 2.40 the degrees of both $\boldsymbol{d}$ and M are polynomially bounded in the given data. Let J be the proper ideal given by M. Then $(g_1, \dots, g_s) = \frac{1}{\boldsymbol{d}} J$, and the degree of J is equal to the degree of the determinant of M. Therefore, the height of $(g_1, \dots, g_s)$ can be bounded as claimed. □

Remark 2.73 A similar statement holds if g is not represented with respect to $w_1, \dots, w_r$ but with respect to $1, \tilde{y}_{|\mathcal{C}}, \dots, \tilde{y}_{|\mathcal{C}}^{r-1}$ or $1, y_{|\mathcal{C}}, \dots, y_{|\mathcal{C}}^{r-1}$.

2.5.4.4 Divisor arithmetic

We now relate the previous ideal-theoretic considerations with divisors, using the joint ideal representation.

Definition 2.74 For a divisor D on $\mathcal{C}$, we define D_+ and D_- as the unique effective divisors with $D = D_+ - D_-$.

The *height* of a divisor D on $\mathcal{C}$ is $\mathrm{ht}(D) := \max\{\deg(D_+), \deg(D_-)\}$.[6]

[6] In [Heß01], the height of a divisor D is defined as $\deg(D_+) + \deg(D_-)$.

Remark 2.75 The *height* of a fractional ideal of $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$ is the height of the corresponding divisor.

Definition 2.76 Let $D_1 = \sum_{P\in\mathcal{C}} v_P(D_1)\,P$ and $D_2 = \sum_{P\in\mathcal{C}} v_P(D_2)\,P$ be two divisors on $\mathcal{C}$. Then the *minimum* of D_1 and D_2 is $\min\{D_1, D_2\} := \sum_{P\in\mathcal{C}} \min\{v_P(D_1), v_P(D_2)\}\cdot P$, and the *maximum* is $\max\{D_1, D_2\} := \sum_{P\in\mathcal{C}} \max\{v_P(D_1), v_P(D_2)\}\cdot P$.

Remark 2.77 We have $I^{\text{fin}}(D_1) \cap I^{\text{fin}}(D_2) = I^{\text{fin}}(\max\{D_1, D_2\})$ and $I^{\text{fin}}(D_1) + I^{\text{fin}}(D_2) = I^{\text{fin}}(\min\{D_1, D_2\})$, and similarly for the "infinite" ideals.

Recall that the *joint ideal representation* of a divisor consists of a pair of ideals $I(((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)) \times I(((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})_\infty)$ obtained by isomorphism (2.3). The "infinite" ideal is represented by the corresponding ideal in $I(((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\frac{1}{\mathrm{x}}(\mathbb{A}^1_k)))$ with support "at infinity". These ideals are represented by HNF-bases as above.

Propositions 2.66, 2.69 and 2.70 as well as the corresponding result for the "infinite" ideals immediately imply:

Proposition 2.78 *Given two divisors D_1, D_2 in joint ideal representation, one can compute $D_1 + D_2$, $\min\{D_1, D_2\}$ and $\max\{D_1, D_2\}$ with a deterministic algorithm in a number of field and bit operations which is polynomially bounded in d, $\mathrm{ht}(D_1)$ and $\mathrm{ht}(D_2)$. Similarly, one can compute the inverse of a divisor D in joint ideal representation in a number of field and bit operations which is polynomially bounded in d and $\mathrm{ht}(D)$.*

Proposition 2.72 implies:

Proposition 2.79 *Let $g = \sum_{i=1}^r \frac{a_i}{b_i} w_i \in k(\mathcal{C})$ with $a_i, b_i \in k[\mathrm{x}]$, represented by the vector $(a_1, \ldots, a_r, b_1, \ldots, b_r)$. Then the height of the principal divisor (g) is polynomially bounded in d and the degree of the vector $(a_1, \ldots a_r, b_1, \ldots, b_r)$. Moreover, this divisor in joint ideal representation can be computed in a number of field and bit operations which is polynomially bounded in the same data.*

Proof. By Proposition 2.72 the principal fractional ideals (g) of both $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$ and $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\frac{1}{\mathrm{x}}(\mathbb{A}^1_k))$ can be computed with the claimed complexity, and their heights are polynomially bounded in d and the degree of the vector in question.

Note that we are not yet finished at this point because we demand that the "infinite" ideal has support "at infinity". The result now follows from the next lemma. □

Lemma 2.80 *Given a fractional ideal I of $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\frac{1}{\mathrm{x}}(\mathbb{A}^1_k))$ one can compute the ideal (I) in $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})_\infty$, represented by the corresponding ideal in $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\frac{1}{\mathrm{x}}(\mathbb{A}^1_k))$, in a number of field and bit operations which is polynomially bounded in d and $\mathrm{ht}(I)$.*

Proof. Let first I be a proper ideal. Note first that the principal fractional ideal (x^{-1}) of $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\frac{1}{\mathrm{x}}(\mathbb{A}^1_k))$ is a proper ideal whose support consists exactly of all prime ideals "at infinity". Now for any $i > 0$, the fractional ideal $(\mathrm{x}^{-i})+I$ has support "at infinity". Moreover, $\mathrm{x}^{-\deg(I)} \in (I) \subseteq ((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})_\infty$. Thus $(\mathrm{x}^{-\deg(I)}) + I \subseteq ((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\frac{1}{\mathrm{x}}(\mathbb{A}^1_k))$ is the desired ideal.

We come to the computation. We first compute I_- and I_+. Note that the degrees of these ideals can immediately be read off from the coordinate matrices of the HNF-bases. Now we compute the ideals $J_+ := (\mathrm{x}^{-\deg(I_+)}) + I_+$ and $J_- := (\mathrm{x}^{-\deg(I_-)}) + I_-$ and finally $J_+J_-^{-1}$, which is the desired ideal. All these computations can be performed with the claimed complexity by Lemma 2.71, Proposition 2.72 and Proposition 2.69. □

Recall that the *different ideal sheaf* of the covering $x_{|\mathcal{C}} : \mathcal{C} \longrightarrow \mathbb{P}^1_k$ is defined as $\mathrm{Ann}(\Omega_{\mathcal{C}/\mathbb{P}^1_k})$, the annihilator of the $\mathcal{O}_\mathcal{C}$-module $\Omega_{\mathcal{C}/\mathbb{P}^1_k}$. The *ramification divisor* (or *different divisor*) R is then the (effective) divisor defined by this ideal sheaf. Note that by definition,

$$\mathcal{O}_R \simeq \Omega_{\mathcal{C}/\mathbb{P}^1_k} \,. \tag{2.5}$$

Along the lines of the proof of [Neu91, Satz 2.7], one can prove that the module of sections of the different ideal sheaf over $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$ is isomorphic to the different ideal $\mathfrak{D}$ defined above. This statement together with the corresponding statement for $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\frac{1}{\mathrm{x}}(\mathbb{A}^1_k))$ imply:

Proposition 2.81 *One can compute the ramification divisor R of $x_{|\mathcal{C}} : \mathcal{C} \longrightarrow \mathbb{P}^1_k$ in joint ideal representation in a number of field and bit operations which is polynomially bounded in d.*

We now have the important formula of divisors on $\mathcal{C}$

$$\mathrm{div}(\mathrm{d}x_{|\mathcal{C}}) = -2(x_{|\mathcal{C}})_- + R \,. \tag{2.6}$$

To prove this formula, note first that we have the exact sequence

$$0 \longrightarrow x^*_{|\mathcal{C}}(\Omega_{\mathbb{P}^1_k}) \longrightarrow \Omega_\mathcal{C} \longrightarrow \Omega_{\mathcal{C}/\mathbb{P}^1_k} \longrightarrow 0 \tag{2.7}$$

(see [Har77, IV, Proposition 2.1]). Together with (2.5) this means that as subsheaf of $\Omega_\mathcal{C}$, $x^*_{|\mathcal{C}}(\Omega_{\mathbb{P}^1_k})$ is isomorphic to $\Omega_\mathcal{C}(-R)$. This implies that for any $s \in \Gamma(\mathcal{C}, x^*_{|\mathcal{C}}(\Omega_{\mathbb{P}^1_k}))$ with $s \neq 0$,

$$\mathrm{div}_{\Omega_\mathcal{C}}(s) = \mathrm{div}_{x^*_{|\mathcal{C}}(\Omega_{\mathbb{P}^1_k})}(s) + R \,. \tag{2.8}$$

Moreover, we have on $\mathbb{P}^1_k$

$$\mathrm{div}(\mathrm{dx}) = -2(\mathrm{x})_- \ . \tag{2.9}$$

Note now that $x^*_{|\mathcal{C}}(\mathrm{dx}) = \mathrm{d}(x^*_{|\mathcal{C}}(\mathrm{x})) = \mathrm{d}x_{|\mathcal{C}}$. Altogether we obtain

$$\begin{aligned}
&-2(x_{|\mathcal{C}})_- = -2x^*_{|\mathcal{C}}(\mathrm{x})_- = x^*_{|\mathcal{C}}(\mathrm{div}(\mathrm{dx})) = \\
&\mathrm{div}_{x^*_{|\mathcal{C}}(\Omega_{\mathbb{P}^1_k})}(x^*_{|\mathcal{C}}(\mathrm{dx})) = \mathrm{div}_{\Omega_{\mathcal{C}}}(x^*_{|\mathcal{C}}(\mathrm{dx})) - R = \\
&\mathrm{div}(\mathrm{d}x_{|\mathcal{C}}) - R \ .
\end{aligned}$$

□

By Proposition 2.81, (2.6) and Lemma 2.80 we obtain:

Proposition 2.82 *One can compute the canonical divisor* $\mathrm{div}(\mathrm{d}x_{|\mathcal{C}})$ *in joint ideal representation in a number of field and bit operations which is polynomially bounded in* d.

2.5.4.5 Curves over extension fields

As a generalization we now consider curves over finite extension fields of k. This subsubsection is in a certain sense a continuation of subsubsection 2.5.3.4. In particular, as in subsubsection 2.5.3.4, all computations take place over k.

Let $\lambda|k$ be a finite field extension. Let $\mathcal{C}$ be a curve over λ or more generally a smooth and proper 1-dimensional irreducible λ-scheme. Analogously to the previous considerations, we assume that $\mathcal{C}$ is represented by a plane model $\mathcal{C}_{pm}$ of degree d in $\mathbb{P}^2_\lambda$, not equal to $V(Z)$. As above, let $F(X,Y,Z) \in \lambda[X,Y,Z]$ be a homogeneous polynomial defining $\mathcal{C}_{pm}$. We define x, y and $f(x,y) \in \lambda[x,y]$ as in subsubsection 2.5.4.1, and we also use the notations of subsubsection 2.5.4.2 (with λ instead of k).

Furthermore, let $a_1, \ldots, a_{[\lambda:k]}$ be a basis of $\lambda|k$, again with known multiplication table.

Similarly to above, we first consider the extension $\lambda(\mathcal{C})|\lambda(\mathrm{x})$. Let $w_1, \ldots, w_r$ be the HNF-basis of $((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_\lambda))$ with respect to $1, \tilde{y}_{|\mathcal{C}}, \ldots, \tilde{y}^{r-1}_{|\mathcal{C}}$.

Note that by Propositions 1.58 and 2.53, $w_1, \ldots, w_r$ can be computed with a deterministic algorithm in a number of field operations (in k) and bit operations which is polynomially bounded in d and $[\lambda : k]$. More generally, by Proposition 1.58 all results from on ideal arithmetic from subsubsection 2.5.4.2 and subsubsection 2.5.4.3 can be transferred to the more general setting of this subsubsection. The only difference is that now the required number of field operations (in k) and bit operations is additionally polynomially bounded in $[\lambda : k]$.

If one proceeds similarly with the "infinite" order, the corresponding results can be transferred too, and consequently the results from subsubsection 2.5.4.4 can then be transferred in the same way.

Now we consider the extension $\lambda(\mathcal{C})|k(\mathrm{x})$ (of degree $[\lambda : k] \cdot r$), and we regard $((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_\lambda))$ as a $k[\mathrm{x}]$-module.

Following (2.2), let

$$\widehat{w}_{i+(j-1)r} := w_i\, a_j \quad \text{for } i = 1, \ldots, r,\ j = 1, \ldots, [\lambda : k]\,. \tag{2.10}$$

We can now represent fractional ideals of $((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_\lambda))$ via HNF-bases with respect to $\widehat{w}_1, \ldots, \widehat{w}_{r\cdot[\lambda:k]}$.

By Lemma 2.64, the following proposition is a special case of Proposition 2.46.

Proposition 2.83 *Let I be a fractional ideal of $((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_\lambda))$. Then one can change between the representation of I by the HNF-basis with respect to $w_1, \ldots, w_r$ and the representation of I by the HNF-basis with respect to $\widehat{w}_1, \ldots, \widehat{w}_{r\cdot[\lambda:k]}$ in a number of field operations in k and bit operations which is polynomially bounded in d and* $\mathrm{ht}(I)$.

By this result the computational statements in subsubsections 2.5.4.2 and subsubsection 2.5.4.3 excluding Remark 2.73 can also be transferred to the representation with respect to $\widehat{w}_1, \ldots, \widehat{w}_{r\cdot[\lambda:k]}$. Again, the only difference is that the number of field operations in k and bit operations required is now additionally polynomially bounded in $[\lambda : k]$. (This can also be checked directly by going through the proofs of these results.)

An analogous statement to Remark 2.73 also holds: One merely has to consider the bases consisting of $\tilde{y}_{|\mathcal{C}}^{i-1}\, a_j$ or of $y_{|\mathcal{C}}^{i-1}\, a_j$ for $i = 1, \ldots, r$ and $j = 1, \ldots, [\lambda : k]$. (This can for example again be seen with Proposition 2.46.)

We now turn to an important application: The computation of intersections of fractional ideals "along a constant field extension".

For this, let $\mathcal{C}$ be a curve over k, represented as described above. As above, we assume that we are given the HNF-basis $w_1, \ldots, w_r$ of $((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k))$ with respect to $1, \tilde{y}_{|\mathcal{C}}, \ldots, \tilde{y}_{|\mathcal{C}}^{r-1}$.

Let $\lambda|k$ again be a finite field extension, represented by a multiplication table with respect to a k-basis $a_1, \ldots, a_{[\lambda:k]}$ of λ with $a_1 = 1$.

Note that under the isomorphism $\lambda(\mathcal{C}) \simeq k(\mathcal{C}) \otimes_k \lambda$,

$$((x_{|\mathcal{C}_\lambda})_*\mathcal{O}_{\mathcal{C}_\lambda})(\mathbb{A}^1_\lambda)) \simeq ((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k)) \otimes_k \lambda\,. \tag{2.11}$$

(The left-hand side is the normalization of the right-hand side, but as k is perfect, $((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k))$ is smooth, and this property is invariant under field extensions.)

Now we have:

Lemma 2.84 *Let I be a fractional ideal of $((x_{|\mathcal{C}_\lambda})_*\mathcal{O}_{\mathcal{C}_\lambda})(\mathbb{A}^1_\lambda))$ in $\lambda(\mathcal{C})$, and let $g_1, \ldots, g_{[\lambda:k]\cdot r}$ be the HNF-basis of I with respect to $\widehat{w}_1, \ldots, \widehat{w}_{[\lambda:k]\cdot r}$. Then $g_1, \ldots, g_r$ is the HNF-basis of $I \cap k(\mathcal{C})$ with respect to $w_1, \ldots, w_r$.*

Indeed, this is a special case of Lemma 2.36.

Remark 2.85 If the ideal I in Lemma 2.84 is instead given by a generating set $g_1, \ldots, g_s$, where each element is represented in the form $g_j = \sum_{i=1}^{r\cdot[\lambda:k]} \frac{a_{i,j}}{b_{i,j}} \widehat{w}_i$ with $a_{i,j}, b_{i,j} \in k[\mathrm{x}]$, one can compute the intersection $I \cap k(\mathrm{x})$ in a number of field and bit operations which is polynomially bounded in $d, [\lambda : k], s$ and the maximum of the heights of the $a_{i,j}$ and $b_{i,j}$. (Briefly, one first computes an HNF-basis with respect to $w_1, \ldots, w_{r\cdot[\lambda:k]}$ and then applies Lemma 2.84.)

2.5.4.6 Changing representations

We address the question to efficiently change between the (free) coordinate, the free ideal and the joint ideal representations. We show that one can change from the free representations to the joint ideal representation (deterministically) in a number of field and bit operations which is polynomially bounded in d and $\mathrm{ht}(D)$, where D is the divisor under consideration.

On the basis of our computational model, every algorithm which computes a free ideal representation from the joint ideal representation of divisors has to be randomized. Indeed, factorization of polynomials is a special case of factorization of ideals, and for this task the algorithm has to use the command FIELDFACTOR, which means that it has to be randomized.

We remark here that it is an interesting open problem to derive a deterministic polynomial time algorithm to factor polynomials over finite fields (on a RAM / Turing machine).

Proposition 2.86 *Given a fractional ideal I of $((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k)$, one can compute the factorization of I with a randomized algorithm in an expected number of field and bit operations which is polynomially bounded in d and* $\mathrm{ht}(I)$.

Proof. Let us first assume that I is a proper ideal.

Let us first fix some notation and make some non-computational observations: Let $((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k)/I \simeq A_1 \oplus \cdots \oplus A_\ell$ with local algebras A_i, and let $\mathfrak{m}_i$ be the corresponding maximal ideals. Let $\mathfrak{p}_i$ be the preimage of $\mathfrak{m}_i$ in $((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k)$, $f_i := \dim_k(A_i/\mathfrak{m}_i)$ and $e_i := \frac{\dim_k(A_i)}{f_i}$ (for $i = 1, \ldots, \ell$). Then the factorization of I is $I = \mathfrak{p}_1^{e_1} \cdots \mathfrak{p}_\ell^{e_\ell}$. Explicitly, if for $i = 1, \ldots, \ell$ $b_1, \ldots, b_s \in ((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k)$ are such that modulo I they form a basis of $\mathfrak{m}_i$. Then $\mathfrak{p}_i := (b_1, \ldots, b_s) + I$.

Let now – as always – I be given by the coordinate matrix M (with respect to $w_1, \ldots, w_r$) in Hermite normal form. We proceed as follows:

We consider the canonical basis of $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)/I$ with respect to $w_1, \ldots w_r$ (see Definition 2.43). We compute the multiplication table of this basis. For this we first compute the multiplication table of $w_1, \ldots, w_r$ (see Lemma 2.58). Then we compute the desired multiplication table with a reduction algorithm with respect to the columns of M. This is possible in a number of field and bit operations which is polynomially bounded in d an d $\mathrm{ht}(I)$.

Then we determine the factorization of the algebra $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)/I$ and determine the maximal ideals $\mathfrak{m}_i$ with respect to this basis (that is, for each maximal ideal, we obtain coordinate vectors of basis elements). We apply the canonical lift to the basis elements to obtain elements in $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$ (see again Definition 2.43). Note that the lifting is computationally trivial. These computations can be performed with the claimed expected number of field and bit operations by Propositions 1.49 and 1.50

For every $i = 1, \ldots, \ell$, we have then obtained elements $b_1, \ldots b_s$ and the index e_i as above. Finally, we compute the HNF-bases of the ideals $\mathfrak{p}_i = (b_1, \ldots, b_s) + I$. What concerns the complexity of these computations note first that $s = \dim_k(\mathfrak{m}_i) \leq \dim_k(A_i) \leq \deg(I)$ and $\ell \leq \deg(I)$. Now these computations can also be performed with the claimed complexity by Proposition 2.72.

Now let I be an arbitrary fractional ideal. We then compute I_+ and I_- (see Lemma 2.71). After this we determine their factorizations. □

This result as well as the corresponding result for fractional ideals of $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\frac{1}{\mathrm{x}}(\mathbb{A}^1_k))$ imply the second statement of the following proposition. The first statement follows immediately from Proposition 2.66.

Proposition 2.87 *Let D be a divisor on $\mathcal{C}$. Then one can change from the free to the joint ideal representation with a deterministic algorithm in a number of field and bit operations which is polynomially bounded in d and $\mathrm{ht}(D)$. Conversely, one can change from the joint to the free ideal representation of D with a randomized algorithm in an expected number of field and bit operations which is polynomially bounded in d and $\mathrm{ht}(D)$.*

Note for the following proposition that we have fixed a birational map $\pi : \mathcal{C} \longrightarrow \mathcal{C}_{pm}$.

Proposition 2.88 *Given a point $P \in \mathcal{C}_{pm}(k) \subseteq \mathbb{P}^2(k)$, one can compute the divisor $\pi^{-1}(P) = \pi^*(P)$ on $\mathcal{C}$ in joint ideal representation in a number of field and bit operations which is polynomially bounded in d.*

Sketch of a proof. Let $P = (X(P) : Y(P) : Z(P)) \in \mathbb{P}^2(k)$. We proceed with several case distinctions.

Let first $\underline{Z(P) \neq 0}$. Then $\pi^{-1}(P) = \min\{(x_{|\mathcal{C}} - \frac{X(P)}{Z(P)})_+, (y_{|\mathcal{C}} - \frac{Y(P)}{Z(P)})_+\}$.

Let now $\underline{Z(P) = 0}$. If $\underline{X(P) \neq 0}$ then $\pi^{-1}(P) = \min\{(\frac{1}{x_{|\mathcal{C}}})_+, (\frac{y_{|\mathcal{C}}}{x_{|\mathcal{C}}} - \frac{Y(P)}{X(P)})_+\}$. If $\underline{Y(P) \neq 0}$ then $\pi^{-1}(P) = \min\{(\frac{1}{y_{|\mathcal{C}}})_+, (\frac{x_{|\mathcal{C}}}{y_{|\mathcal{C}}} - \frac{X(P)}{Y(P)})_+\}$.

The computations can be performed in a number of field and bit operations which is polynomially bounded in d by Proposition 2.78. □

Proposition 2.89 *Given a field extension $\lambda|k$ (represented by a multiplication table) and a point $P \in \mathcal{C}_{pm}(\lambda)$, one can compute the set of closed points of $\mathcal{C}$ lying over the closed point defined by P with a randomized algorithm in an expected number of field and bit operations which is polynomially bounded in d and $\deg([\lambda : k])$. Here the closed points of $\mathcal{C}$ are represented in joint ideal representation.*

Proof. First one computes the divisor $\pi^{-1}(P)$ on $\mathcal{C}_\lambda$ in joint ideal representation. This is possible as claimed by the previous proposition and Proposition 1.58. Then one computes the intersection of the corresponding ideals with the coordinate rings over k (see subsubsection 2.5.4.5, and in particular Lemma 2.84). Finally, one factors the ideals (see Proposition 2.86). □

If the point P lies in $\mathcal{C}_{ns}$, then $\pi^{-1}(P)$ is a prime divisor, and we can avoid factorization. We therefore obtain:

Proposition 2.90 *Let a closed point $P \in \mathcal{C}$, not lying over a singular point of $\mathcal{C}_{pm}$, be given in coordinate representation as described in subsection 2.5.2. Then one can compute the ideal representation (that is, the corresponding prime ideal of either the finite or infinite order) with a deterministic algorithm in a number of field and bit operations which is polynomially bounded in d and $\deg(P)$.*

Proposition 2.91 *Given a closed point $P \in \mathcal{C}$ in ideal representation, one can compute with a randomized algorithm a field extension $\lambda|k$ (of degree polynomially bounded in $\deg(P)$, given by a defining separable polynomial) such that $\pi(P)$ is given by a point in $\mathbb{P}^2(\lambda)$, and coordinates of such a point (that is, coordinates of $\pi(P)$) in an expected number of field and bit operations which is polynomially bounded in d and $\deg(P)$.*

Proof. Let first P be represented by an ideal of the infinite order. Then $\pi(P)$ lies in $V(Z)$. Therefore, we can proceed as follows: We factorize the polynomial $F(X, Y, 0)$ to obtain the points in the intersection $\mathcal{C}_{pm} \cap V(Z) = V(F(X, Y, Z), Z)$. Let $F(X, Y, 0) = \prod_{i=1}^{a} F_i(X, Y)^{e_i}$ be the factorization. Then each factor F_i corresponds to a closed point Q_i in $\mathcal{C}_{pm} \cap V(Z)$. Coordinates of these points can be obtained as follows: If $F_i = c \cdot Y$ for $c \in k^*$,

then the point is given by $(0:1:0)$. Otherwise, let $f_i(\mathrm{x}) := F(\mathrm{x},1)$. Then with $\lambda := k[\mathrm{x}]/(f_i)$ and α the residue class of x in λ, the closed point associated to $(\alpha:1:0)$ is Q_i. For each point Q_i (represented by coordinates as described), we compute the set of closed points of $\mathcal{C}$ lying over Q_i, and we check if P is contained in this set. If this is the case, we have determined $\pi(P)$.

By Proposition 2.89 this computation can be performed in an expected number of field and bit operations which is polynomially bounded in d and $\deg(P)$.

Let now P be represented by an ideal $\mathfrak{p}$ of the finite order. Let m be the monic generator of the intersection $\mathfrak{p} \cap k[\mathrm{x}]$. Note that m is irreducible; let $K := k[\mathrm{x}]/(m(\mathrm{x}))$, and let α be the residue class of x in this field. Then there exists a point in $(\mathcal{C}_{pm})_K \cap V(X-\alpha Z) \subseteq \mathbb{P}^2_K$ which lies over $\pi(P)$ under the map $(\mathcal{C}_{pm})_K \longrightarrow \mathcal{C}_{pm}$.

The computation is as follows: Recall that $\mathfrak{p}$ is given by its HNF-basis with respect to $w_1, \ldots, w_r$, which in turn is given by the coordinate matrix M, and $m = m_{1,1}$ (see Lemma 2.61). We compute the points in $(\mathcal{C}_{pm})_K \cap V(X-\alpha Z) \subseteq \mathbb{P}^2_K$, that is, we factorize $F(\alpha Z, Y, Z)$; let again $F(\alpha Z, Y, Z) = \prod_{i=1}^{a} F_i(Y,Z)$ be the factorization. Then for each $i = 1, \ldots, a$ we compute primitive elements over k and their minimal polynomials for the resulting field extensions (see Proposition 2.1). Finally, as before for each point Q_i corresponding to the factor F_i we compute the set of closed points of $\mathcal{C}$ lying over Q_i, and again we determine in which set P is contained. □

2.5.4.7 Computing Riemann-Roch spaces

The joint ideal representation of divisors is particularly important because it allows for a simple and efficient computation of Riemann-Roch spaces via Heß' algorithm ([Heß01]). We present the algorithm here.

Let us first fix the computational problem we consider: Given a divisor D on $\mathcal{C}$ in joint ideal representation, we want to compute a basis of the space $L(D)$. More precisely, we want to compute the coordinate matrix of such a basis with respect to $1, y_{|\mathcal{C}}, \ldots, y_{|\mathcal{C}}^{r-1}$. As usual, we represent the coordinate matrix by its denominator and the numerator matrix.

The algorithm relies on the following easy fact: Let D be a divisor on $\mathcal{C}$. Then

$$L(D) = I^{\mathrm{fin}}(-D) \cap (I^{\infty}(-D))_{\infty} .$$

Note that for $n \in \mathbb{Z}$, $(I^{\infty}(-(D + n(\mathrm{x})_-)))_{\infty} = (I^{\infty}(-D + n\mathrm{x}))_{\infty} = \mathrm{x}^n \cdot I^{\infty}(-D)$. As a special case of Proposition 2.22 we then obtain:

Proposition 2.92 *There exist uniquely determined integers $d_1 \geq \cdots \geq d_r$ and a basis $v_1, \ldots, v_r$ of $I^{\mathrm{fin}}(-D)$ such that $\mathrm{x}^{d_1}v_1, \ldots, \mathrm{x}^{d_r}v_r$ is a $k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})}$-basis of $(I^{\infty}(-D))_{\infty}$. With these data the following holds:*

- *For all $n \in \mathbb{Z}$, the set*

$$\bigcup_{i=1}^{r} \{\mathrm{x}^j v_i \mid 0 \leq j \leq d_i + n\}$$

  is a k-basis of $L(D + n \cdot (\mathrm{x})_-)$.

- *Let $\iota(n) := \max\{j \in \{1, \ldots, r\} \mid d_j + n \geq -1\}$. Then $\dim_k(D + n \cdot (\mathrm{x})_-) - \dim_k(D - (n-1) \cdot (\mathrm{x})_-) = \iota(n-1)$.*

Remark 2.93 The uniquely determined integers $d_1 \geq \cdots \geq d_r$ are called the $k[\mathrm{x}]$-*invariants* of D in [Heß01].

Recall that by the proof of Proposition 2.22 we have the following explicit result:

Lemma 2.94 *Let $\tilde{v}_1, \ldots, \tilde{v}_r$ be a $k[\mathrm{x}]$-basis of $I^{\mathrm{fin}}(-D)$, and let $b_1, \ldots, b_r$ be a $k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})}$ be a basis of $I^{\infty}(-D)_{\infty}$. Now let $A \in k(\mathrm{x})^{r \times r}$ be the coordinate matrix of $\tilde{v}_1, \ldots, \tilde{v}_r$ with respect to $b_1, \ldots, b_r$. Let $AU = M$ with $U \in k[\mathrm{x}]^{r \times r}$ unimodular and M degree-reduced. Let $M = TD$ with $D = \mathrm{diag}(\mathrm{x}^{-d_1}, \ldots, \mathrm{x}^{-d_r})$ and $T \in k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})}$ unimodular.*

Let now $v_1, \ldots, v_r \in k(\mathrm{x})$ be such that their coordinate matrix with respect to $\tilde{v}_1, \ldots, \tilde{v}_r$ is U. Then

a) $v_1, \ldots, v_r$ from a $k[\mathrm{x}]$-basis of $I^{\mathrm{fin}}(-D)$.

b) The coordinate matrix of $v_1, \ldots, v_r$ with respect to $b_1, \ldots, b_r$ is M.

c) The coordinate matrix of $\mathrm{x}^{d_1}v_1, \ldots, \mathrm{x}^{d_r}v_r$ with respect to $b_1, \ldots, b_r$ is T. In particular, $\mathrm{x}^{d_1}v_1, \ldots, \mathrm{x}^{d_r}v_r$ form a $k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})}$-basis of $I^{\infty}(-D)_{\infty}$.

This immediately leads to an algorithm to compute a basis $v_1, \ldots, v_r$ as in Proposition 2.92 and the $k[\mathrm{x}]$-invariants of D:

Algorithm for computation of Riemann-Roch spaces

Input: A curve given by a plane model and the data described in subsubsection 2.5.4.2, a divisor D on $\mathcal{C}$ in joint ideal representation.

1. Determine the HNF-bases $\tilde{v}_1, \ldots, \tilde{v}_r$ of $I^{\mathrm{fin}}(-D)$ and $b_1, \ldots, b_r$ of $I^{\infty}(-D)_{\infty}$.
2. Compute the coordinate matrix of $\tilde{v}_1, \ldots, \tilde{v}_r$ with respect to $b_1, \ldots, b_r$ in the form $\frac{1}{g}N$, where $g \in k[\mathrm{x}]$ is the denominator

and $N \in k[\mathrm{x}]^{r\times s}$ the numerator matrix.

3. Compute a degree-reduced matrix M which is right-equivalent to N, where the degrees of the columns of M monotonically increase from left to right.
4. Compute $v_1, \ldots, v_r$ via point b) in the lemma.
5. Let d_i' be the degree of the i^{th} column of M.
   Let $d_i \longleftarrow -d_i' - \deg(g)$ for $i = 1, \ldots, r$.
6. Output $v_1, \ldots, v_r$ and $d_1, \ldots, d_r$.

We now show that all steps can be performed in a number of field and bit operations which is polynomially bounded in d and $\mathrm{ht}(D)$.

Step 1 can be performed in polynomial time in $\mathrm{ht}(D)$ and d by Proposition 2.69.

To analyze, Step 2, let us first fix some notation:

- Let A_1 be the coordinate matrix of $\tilde{v}_1, \ldots, \tilde{v}_r$ with respect to $w_1, \ldots, w_r$,
- let A_2 be the coordinate matrix of $w_1, \ldots, w_r$ with respect to $1, y_{|\mathcal{C}}, \ldots, y_{|\mathcal{C}}^{r-1}$,
- let $A_3 := \mathrm{diag}(1, \mathrm{x}^c, \cdots, \mathrm{x}^{c(r-1)})$,
- let A_4 be the coordinate matrix of the HNF-basis of $((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\frac{1}{\mathrm{x}}(\mathbb{A}^1_k))$ with respect to $1, \frac{y_{|\mathcal{C}}}{\mathrm{x}^c}, \ldots, \frac{y_{|\mathcal{C}}^{r-1}}{\mathrm{x}^{c(r-1)}}$,
- and let A_5 be the coordinate matrix of $b_1, \ldots, b_r$ with respect to the HNF-basis of $((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\frac{1}{\mathrm{x}}(\mathbb{A}^1_k))$.

Note that $A_5^{-1}A_4^{-1}A_3A_2A_1$ is the coordinate matrix of $v_1, \ldots, v_r$ with respect to $b_1, \ldots, b_r$.

Note that the matrices A_1, A_2, A_4, A_5 are given with the input of the algorithm, however they are represented differently: A_1 and A_2 are represented in the form $\frac{1}{\boldsymbol{d}_i}N_i$ with $\boldsymbol{d}_i \in k[\mathrm{x}]$ and $N_i \in k[\mathrm{x}]^{r\times r}$, where $\boldsymbol{d}_i$ is the denominator. A_4 and A_5 are represented in the form $\frac{1}{\tilde{\boldsymbol{d}}_i}\tilde{N}_i$ with $\tilde{\boldsymbol{d}}_i \in k[\frac{1}{\mathrm{x}}]$ and $\tilde{N}_i \in k[\frac{1}{\mathrm{x}}]^{r\times r}$, where again $\tilde{\boldsymbol{d}}_i$ is the denominator, but with respect to $k[\frac{1}{\mathrm{x}}]$.

Note now that $\deg(N_1)$ and $\deg(\boldsymbol{d}_1)$ are polynomially bounded in d and $\mathrm{ht}(D)$ by Lemma 2.64, and $\deg(N_2)$ and $\deg(\boldsymbol{d}_2)$ are polynomially bounded in d by Proposition 2.53 v). Similarly, $\deg_{\frac{1}{\mathrm{x}}}(\tilde{N}_3)$ and $\deg_{\frac{1}{\mathrm{x}}}(\tilde{\boldsymbol{d}}_3)$ are polynomially bounded in d by Proposition 2.53 v), and $\deg_{\frac{1}{\mathrm{x}}}(\tilde{N}_4)$ and $\deg_{\frac{1}{\mathrm{x}}}(\tilde{\boldsymbol{d}}_4)$ are polynomially bounded in d and $\mathrm{ht}(D)$ by Lemma 2.64.

Let now for $i = 4, 5$ $N_i := \mathrm{x}^{\deg_{\frac{1}{\mathrm{x}}}(\tilde{N}_i)}\tilde{N}_i \in k[\mathrm{x}]^{r\times r}$ and $\boldsymbol{d}_i := \mathrm{x}^{\deg_{\frac{1}{\mathrm{x}}}(\boldsymbol{d}_i)}\tilde{\boldsymbol{d}}_i$. Then $A_i = \frac{\mathrm{x}^{\deg_{\frac{1}{\mathrm{x}}}(\boldsymbol{d}_i)-\deg_{\frac{1}{\mathrm{x}}}(\tilde{N}_i)}}{\boldsymbol{d}_i}N_i$.

The degrees of all polynomials $\boldsymbol{d}_i$ and all matrices N_i are now polynomially bounded in d and $\mathrm{ht}(D)$. Moreover, one can therefore easily compute $\boldsymbol{d}_i$ and N_i for $i = 4, 5$ in a number of field and bit operations which is polynomially bounded in d and $\mathrm{ht}(D)$.

Now for $i = 4, 5$ the adjoint matrix $N_i^{\#}$ can be computed in a number of field and bit operations which is polynomially bounded in r and $\deg(N_i)$ by Proposition 2.31, and the degree of the resulting matrix is also polynomially bounded in r and $\deg(N_i)$. We conclude that these computations are possible in a number of field and bit operations which is polynomially bounded in d and $\mathrm{ht}(D)$, and the degrees of the resulting matrices also are polynomially bounded in these quantities.

Therefore, the matrix $N_5^{\#} N_4^{\#} A_3 N_2 N_1$ can also be computed in a number of field and bit operations which is polynomially bounded in d and $\mathrm{ht}(D)$. Finally, the matrix

$$A_5^{-1} A_4^{-1} A_3 A_2 A_1 = \frac{\mathrm{x}^{\deg_{\frac{1}{\mathrm{x}}}(\tilde{N}_4)+\deg_{\frac{1}{\mathrm{x}}}(\tilde{N}_5)-\deg_{\frac{1}{\mathrm{x}}}(\tilde{\boldsymbol{d}}_4)-\deg_{\frac{1}{\mathrm{x}}}(\tilde{\boldsymbol{d}}_5)} \boldsymbol{d}_4 \boldsymbol{d}_5}{\boldsymbol{d}_1 \boldsymbol{d}_2 \det(N_5) \det(N_4)} N_5^{\#} N_4^{\#} A_3 N_2 N_1$$

(that is, its denominator and numerator matrix) can also be computed in a number of field and bit operations which is polynomially bounded in d and $\mathrm{ht}(D)$ (see Remark 2.38).

In Step 4 explicitly we have to compute the coordinate matrix of $\tilde{v}_1, \ldots, \tilde{v}_r$ with respect to $1, y_{|\mathcal{C}}, \ldots, y_{|\mathcal{C}}^{r-1}$. This matrix is $\mathrm{diag}(1, a_r(\mathrm{x}), \ldots, a_r(\mathrm{x})^{r-1}) \cdot A_3^{-1} A_4 A_5 \cdot \frac{1}{g} M$. Similarly to Step 3, this task can again be performed in a number of field and bit operations which is polynomially bounded in d and $\mathrm{ht}(D)$.

Step 5 can obviously also be carried out as desired.

We obtain:

Proposition 2.95 *Given a divisor D on $\mathcal{C}$ in joint ideal representation, one can compute the $k[\mathrm{x}]$-invariants $d_1, \ldots, d_r$ of D and a basis $v_1, \ldots, v_r$ of $I^{\mathrm{fin}}(D)$ as in Proposition 2.92 (given by its coordinate matrix with respect to $1, y_{|\mathcal{C}}, \ldots, y_{|\mathcal{C}}^{r-1}$) in a number of field and bit operations which is polynomially bounded in d and $\mathrm{ht}(D)$. The degrees of the denominator and the numerator matrix of the coordinate matrix are then also polynomially bounded in d and $\mathrm{ht}(D)$.*

In particular, one can compute the space $L(D)$ in a number of field and bit operations which is polynomially bounded in d and $\mathrm{ht}(D)$, and the degrees of the denominator and the numerator matrix are then again polynomially bounded in d and $\mathrm{ht}(D)$.

By Equation (2.6) the height of $\mathrm{div}(\mathrm{d}x_{|\mathcal{C}})$ is polynomially bounded in d. With Proposition 2.82 and the previous proposition we obtain:

Proposition 2.96 *One can compute the genus of $\mathcal{C}$ in a number of field and bit operations which is polynomially bounded in d.*

We have the following application of the computation of Riemann-Roch spaces:

Proposition 2.97 *Given a divisor D on $\mathcal{C}$ in joint ideal representation, one can with a deterministic algorithm determine in a number of field and bit operations which is polynomially bounded in d and $\mathrm{ht}(D)$ if there exists an effective divisor which is linearly equivalent to D and if this is the case compute such a divisor.*

Proof. Such a divisor exists if and only if $L(D)$ is non-empty. Moreover, if this is the case and $g \in L(D)$, then $D + (g)$ is an effective divisor which is linearly equivalent to D.

We therefore compute $L(D)$. If $L(D)$ is non-empty, we choose some element $g \in L(D)$ (this can be done in a deterministic way). Then we compute the ideal $D + (g)$.

These operations can be performed with the claimed complexity by Proposition 2.95 and Proposition 2.72 together with Remark 2.73. □

Remark 2.98 Let us consider curves over finite fields: Let $\mathcal{C}$ be a curve over $\mathbb{F}_q$, and let again D be a divisor on $\mathcal{C}$. Then the following holds:

a) One can with a deterministic algorithm determine in a time (in bit operations) which is polynomially bounded in $\log(q)$, d and $\mathrm{ht}(D)$ if there exists an effective divisor which is linearly equivalent to D.

b) If this is the case one can with a randomized algorithm compute a random divisor which is uniformly distributed in the linear system $|D|$ in an expected time which is polynomially bounded in $\log(q)$, d and $\mathrm{ht}(D)$.

Indeed, the first statement immediately follows from the proposition. For the second statement, one merely has to choose the function $g \in L(D)$ uniformly at random. Note that this is easily possible by choosing its coordinate vector with respect to the computed basis of $L(D)$ uniformly at random.

2.5.4.8 Heß' algorithm and non-archimedean lattices

All computations in Heß' algorithm take place on matrices with entries in $k[\mathrm{x}]$. There is however a connection between the ideal theoretic approach to Riemann-Roch spaces and "non-archimedean lattices" which is interesting for two reasons: First one can see certain analogies to Minkowski's lattice theory for ideals in number fields. And second, several important works on

the ideal theoretic approach, in particular the book by Hensel and Landsberg ([HL02]), can be interpreted via this "lattice theory". We now briefly describe these connections.

Minkowski theory Let first K be a number field of degree r, and let $\sigma_1, \ldots, \sigma_r : K \longrightarrow \mathbb{C}$ be the r embeddings of K in to $\mathbb{C}$. (We do not distinguish between real and complex embeddings.) There is an obvious operation of $\mathrm{Gal}(\mathbb{C}|\mathbb{R})$ on $\{\sigma_1, \ldots, \sigma_r\}$, and this operation induces an operation of $\mathrm{Gal}(\mathbb{C}|\mathbb{R})$ on $\mathbb{C}^r$. Now the "Minkowski space" for K is the fixed set of $\mathbb{C}^r$ under this operation. If I is a fractional ideal of K, the associated lattice is the image of I in the Minkowski space.

There is another more intrinsic approach to the Minkowski space which we describe now. Note that we have an isomorphism

$$K \otimes_{\mathbb{Q}} \mathbb{C} \xrightarrow{\sim} \mathbb{C}^r \tag{2.12}$$

induced by the embeddings $\sigma_1, \ldots, \sigma_r$. Now on the left-hand side we have the canonical operation of $\mathrm{Gal}(\mathbb{C}|\mathbb{R})$, and on the right-hand side we have the operation just defined. Now the isomorphism in (2.12) is $\mathrm{Gal}(\mathbb{C}|\mathbb{R})$-invariant. It follows that we have an induced $\mathbb{R}$-vector space isomorphism between $K \otimes_{\mathbb{Q}} \mathbb{R}$ and the Minkowski space. Under this isomorphism, the lattice associated to a fractional ideal I of K is the image of I under the canonical embedding of K into $K \otimes_{\mathbb{Q}} \mathbb{R}$.

An analog for function fields In analogy to these considerations for number fields, it is natural to consider the following construction for the separable extension $k(\mathcal{C})|k(\mathrm{x})$ of function fields:

We follow the notations from above, and we let I be a proper ideal in $((x_{|\mathcal{C}_\lambda})_* \mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k)$. We now consider the r-dimensional $k((\mathrm{x}^{-1}))$-vector space $k(\mathcal{C}) \otimes_{k(\mathrm{x})} k((\mathrm{x}^{-1}))$, and we identify I with its image under the canonical inclusion. Now as already remarked in Remark 2.23, one can consider I as a "non-archimedean lattice".

Let now $\sigma_1, \ldots, \sigma_r : k(\mathcal{C}) \longrightarrow k((\mathrm{x}^{-1}))^{\mathrm{sep}}$ be the embeddings of $k(\mathcal{C})|k(\mathrm{x})$ into $k((\mathrm{x}^{-1}))^{\mathrm{sep}}$. Again these embeddings induce an isomorphism

$$k(\mathcal{C}) \otimes_{k(\mathrm{x})} k((\mathrm{x}^{-1}))^{\mathrm{sep}} \xrightarrow{\sim} \left(k((\mathrm{x}^{-1}))^{\mathrm{sep}}\right)^r . \tag{2.13}$$

Now again we have an obvious operation by $\Gamma_{k((\mathrm{x}^{-1}))} := \mathrm{Gal}(k((\mathrm{x}^{-1}))^{\mathrm{sep}}|k((\mathrm{x}^{-1})))$ on the left-hand side and on $\{\sigma_1, \ldots, \sigma_r\}$, and thus on the right-hand side. Isomorphism (2.13) is now $\Gamma_{k((\mathrm{x}^{-1}))}$-invariant, and in particular the image of $k(\mathcal{C}) \otimes_{k(\mathrm{x})} k((\mathrm{x}^{-1}))$ under isomorphism (2.13) is equal to the fixed set of $(k((\mathrm{x}^{-1}))^{\mathrm{sep}})^r$ under the operation of $\Gamma_{k((\mathrm{x}^{-1}))}$.

Let $\mathfrak{p}_\infty$ be the place of $k(\mathrm{x})|k$ "at infinity", and let $\mathfrak{p}_1, \ldots, \mathfrak{p}_h$ be the places of $k(\mathcal{C})|k$ over $\mathfrak{p}_\infty$. Then we have a $k((\mathrm{x}^{-1}))$-isomorphism

$$k(\mathcal{C}) \otimes_{k(\mathrm{x})} k((\mathrm{x}^{-1})) \xrightarrow{\sim} \prod_{i=1}^{h} k(\mathcal{C})_{\mathfrak{p}_i} \ . \tag{2.14}$$

Let $r_i := [k(\mathcal{C})_{\mathfrak{p}_i} : k((\mathrm{x}^{-1}))]$. Then we have r_i embeddings $\sigma_{i,j} : k(\mathcal{C})_{\mathfrak{p}_i} \longrightarrow k((\mathrm{x}^{-1}))^{\mathrm{sep}}$ over $k((\mathrm{x}^{-1}))$, and these embeddings induce isomorphisms

$$k(\mathcal{C})_{\mathfrak{p}_i} \otimes_{k((\mathrm{x}^{-1}))} k((\mathrm{x}^{-1}))^{\mathrm{sep}} \xrightarrow{\sim} (k((\mathrm{x}^{-1}))^{\mathrm{sep}})^{r_i} \ . \tag{2.15}$$

Let $\iota_i : k(\mathcal{C}) \longrightarrow k(\mathcal{C})_{\mathfrak{p}_i}$ be the canonical embedding. Then the r embeddings of $k(\mathcal{C})$ into $k((\mathrm{x}^{-1}))^{\mathrm{sep}}$ are $\sigma_{i,j} \circ \iota_i$ for $i = 1, \ldots, h, j = 1, \ldots, r_i$.

We are now coming back to the determination of the space $L(D)$ for a divisor D on $\mathcal{C}$. Let $I := I^{\mathrm{fin}}(-D)$. As already stated, we now regard I as a "lattice" in $k(\mathcal{C}) \otimes_{k(\mathrm{x})} k((\mathrm{x}^{-1}))$. We want to express $L(D)$ via the "lattice" I and some condition "at infinity". Let for this $|\cdot|_i$ be the non-archimedean absolute value on $k(\mathcal{C})_{\mathfrak{p}_i}$ given by $|a|_i = e^{-v_{\mathfrak{p}_i}(a)}$. Let $I^\infty(-D)_\infty = \prod_i \mathfrak{p}_i^{c_i}$, and let $\mathfrak{M}_\infty$ be the $k[[\mathrm{x}^{-1}]]$-submodule of $k(\mathcal{C}) \otimes_{k(\mathrm{x})} k((\mathrm{x}^{-1}))$ corresponding to

$$\{\underline{a} = (a_i)_i \in \prod_{i=1}^{h} k(\mathcal{C})_{\mathfrak{p}_i} \mid \ |a_i|_i \leq e^{c_i}\}$$

under isomorphism (2.14).

Now we clearly have $L(D) = I \cap \mathfrak{M}_\infty$. Note here that $\mathfrak{M}_\infty$ can be interpreted as a (higher dimensional) "non-archimedean cuboid", and thus $L(D)$ is the intersection of a "lattice" with a "cuboid". This idea can now be converted back to the number theoretic situation, and in fact the analogously defined set $H^0(\mathfrak{a})$ for a "complete ideal" $\mathfrak{a}$ of a number field defined for example in [Neu91, III, §3] is the intersection of a lattice with a higher-dimensional cuboid.[7]

Note also that Proposition 2.92 and Lemma 2.94 on the determination of $L(D + n \cdot (\mathrm{x})_-)$ still hold if $k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})}$ is replaced by $k[[\mathrm{x}^{-1}]]$ and $I^\infty(-D)_\infty$ is replaced by $\mathfrak{M}_\infty$. (See also Remark 2.23.)

We now claim that we have

$$I^\infty(-D)_\infty \otimes_{k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})}} k[[\mathrm{x}^{-1}]] = \mathfrak{M}_\infty \ . \tag{2.16}$$

Note first that the result holds for the trivial divisor, that is, isomorphism (2.14) restricts to a $k[[\mathrm{x}^{-1}]]$-isomorphism

$$((x_{|\mathcal{C}})_* \mathcal{O}_\mathcal{C})_\infty \otimes_{k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})}} k[[\mathrm{x}^{-1}]] \xrightarrow{\sim} \prod_{i=1}^{h} \widehat{\mathfrak{O}}_{\mathfrak{p}} \ , \tag{2.17}$$

[7] Note the minus signs in the definition of $o(D)$ in [Neu91].

where $\widehat{\mathfrak{O}}_{\mathfrak{p}_i}$ is the completion of the local ring at the place $\mathfrak{p}_i$. Indeed, this is a special case of [Eis95, Corollary 7.6.].

Note now that the semi-local ring $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})_\infty$ is a principal ideal domain. Let u be a generator of $I^\infty(-D)_\infty$. Then the right-hand side is $u\cdot((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})_\infty \otimes_{(\frac{1}{\mathrm{x}})} k((\mathrm{x}^{-1}))$, which corresponds to $\prod_i \iota_i(u)\widehat{\mathfrak{O}}_{\mathfrak{p}_i} = \prod_i \mathfrak{p}_i^{-c_i}\widehat{\mathfrak{O}}_{\mathfrak{p}_i}$ under isomorphism (2.14), which in turn corresponds to $\mathfrak{M}_\infty$ again under isomorphism (2.14).

Note that by (2.16) every basis $k[\frac{1}{\mathrm{x}}]_{(\frac{1}{\mathrm{x}})}$-basis of $I^\infty(-D)_\infty$ is also a $k[[\mathrm{x}^{-1}]]$-basis of $\mathfrak{M}_\infty$. The statements in Proposition 2.92 and Lemma 2.94 are thus special cases of the corresponding statements if one replaces $k(\mathrm{x})_{(\frac{1}{\mathrm{x}})}$ by $k((\mathrm{x}^{-1}))$ and $I^\infty(-D)_\infty$ by $\mathfrak{M}_\infty$.

Instead of a basis of $I^\infty(-D)_\infty$ one might now in particular use a $k[[\mathrm{x}^{-1}]]$-basis of $\mathfrak{M}_\infty$ of the following form: For $i = 1, \ldots, h$, let $b_{i,1}, \ldots, b_{i,r_i}$ be a basis of $\mathfrak{p}_i\widehat{\mathfrak{O}}_{\mathfrak{p}_i}$. Then the basis of $k(\mathcal{C}) \otimes_{k(\mathrm{x})} k((\mathrm{x}^{-1}))$ corresponding to $(\delta_{i,\ell}b_{\ell,j})_\ell$ for $i = 1, \ldots, h$ and $j = 1, \ldots, r_i$ under isomorphism (2.14) is a basis of $\mathfrak{M}_\infty$.

The approach by Hensel and Landsberg We are now coming to the explicit determination of $L(D)$ following Hensel and Landsberg ([HL02]) and its connection with Heß' algorithm and the discussion so far. Hensel and Landsberg only consider function fields over the complex numbers but this can easily be generalized to an arbitrary algebraically closed ground field. They first perform a coordinate change on $\mathbb{P}^1_k$ such that D does have no support "at infinity" and such that the covering $\mathcal{C} \longrightarrow \mathbb{P}^1_k$ is unramified "at infinity".

We assume that all these conditions are satisfied. In this case $h = r$ and $k(\mathcal{C})_{\mathfrak{p}_i} = k((\mathrm{x}^{-1}))$ for all $i = 1, \ldots, r$. Now given a $k[\mathrm{x}]$-basis $\tilde{v}_1, \ldots, \tilde{v}_r$ of I, Hensel and Landsberg consider the matrix $((\iota_i(\tilde{v}_j)))_{i,j} \in k((\mathrm{x}^{-1}))$. (This is the coordinate matrix of $\tilde{v}_1, \ldots, \tilde{v}_r$ with respect to the basis of $k(\mathcal{C}) \otimes_{k(\mathrm{x})} k((\mathrm{x}^{-1}))$ which corresponds to the standard basis of the $k((\mathrm{x}^{-1}))$-vector space $\prod_{i=1}^h k(\mathcal{C})_{\mathfrak{p}_i} = k((\mathrm{x}^{-1}))^r$ under isomorphism (2.14).) Let us call this matrix the *expansion matrix* at ∞ of $\tilde{v}_1, \ldots, \tilde{v}_r$. (The matrix depends on the ordering of the places of $k(\mathcal{C})$ "at infinity".)

With a "reduction algorithm" they show that there always exists a $k[\mathrm{x}]$-basis $v_1, \ldots, v_r$ of I whose expansion matrix at ∞ is degree-reduced. Let M be this matrix, and let $d_i := -\deg(\underline{m}_i)$. Then again the $\mathrm{x}^j v_i$ with $0 \le j \le d_i$ form a basis of $L(D)$. This result is of course a special case of the considerations above, but it can also be easily seen directly: For all $\underline{b} \in k[\mathrm{x}]^r$, $\max_{i=1}^r \mathrm{valdeg}(\iota_i(\sum_j b_j v_j)) = \mathrm{valdeg}(M\underline{b}) = \max_{j=1}^r \mathrm{valdeg}(b_j\underline{m}_j)$, and this is ≤ 0 if and only if $\deg(b_i) \le d_i$ for all $i = 1, \ldots, r$.

Note that by (2.16) the matrices considered by Hensel and Landsberg and

by Heß are related via the multiplication by a $k[[\mathrm{x}^{-1}]]$-unimodular matrix from the left. Note here the interesting fact that as stated in Remark 2.23, the linear transformation given by such a matrix can be seen as a "non-archimedean isometry".

Let us still assume that there is no ramification "at infinity". Then a $k(\mathrm{x})$-basis of $k(\mathcal{C})$ such that the expansion matrix at ∞ is degree-reduced is called *normal* at ∞ by Hensel and Landsberg. Note that by the discussion above (in particular the isomorphism in (2.17)) and the fact that "degree-reducedness" is invariant under multiplication by a $k[[\mathrm{x}^{-1}]]$-univariate matrix from the left, a system is normal at ∞ if and only if the coordinate matrix with respect to any $k[[\mathrm{x}^{-1}]]$-basis of $(x_{|\mathcal{C}}\mathcal{O}_{\mathcal{C}})_\infty \otimes_{k(\mathrm{x})} k[[\mathrm{x}^{-1}]]$ is degree-reduced.

The work by Hensel and Landsberg contains a generalization of the notion of normal basis at a point of $\mathbb{P}^1_k$ to ramified points (for $k = \mathbb{C}$). This generalization is not used in their proof of the Riemann-Roch Theorem but it can also be used for the description of Riemann-Roch spaces.

We describe this approach now for the point ∞. For this it is crucial to assume that ∞ is only *tamely ramified*.

Let $e_i = r_i$ be the ramification index of $\mathfrak{p}_i$ over $\mathfrak{p}_\infty$, and let e be the least common multiple of the e_i. Then Hensel and Landsberg consider a matrix whose entries are Laurent series in $\mathrm{x}^{-\frac{1}{e}}$, defined as follows (in slightly updated terminology):

For every $i = 1, \ldots, h$, we have e_i embeddings $\sigma_{i,j} : k(\mathcal{C})_{\mathfrak{p}_i} \longrightarrow k((\mathrm{x}^{-\frac{1}{e}}))$ over $k((\mathrm{x}^{-1}))$. Let us choose an e^{th} root of unity in $k((\mathrm{x}^{-\frac{1}{e}}))$. Then the embeddings $\sigma_{i,j}$ can be numbered such that the following holds: Let $\sigma_{i,1}(u) = \sum_{\ell=a}^{\infty} u_{i,\ell}\mathrm{x}^{-\frac{\ell}{e}}$. Then $\sigma_{i,j}(u) = \sum_{\ell=a}^{\infty} \zeta_\ell^{(j-1)\ell} u_{i,\ell}\mathrm{x}^{-\frac{\ell}{e}}$.

Let $\tilde{v}_1, \ldots, \tilde{v}_r$ be a $k(\mathrm{x})$-basis of $k(\mathcal{C})$, and let us fix an ordering on the embeddings of $k(\mathcal{C})$ into $k((\mathrm{x}^{-\frac{1}{e}}))$. Then Hensel and Landsberg consider the matrix A in $k((\mathrm{x}^{-\frac{1}{e}}))$ whose j^{th} column is given by the images of $\tilde{v}_j$ under all embeddings into $k((\mathrm{x}^{-\frac{1}{e}}))$ (with the fixed ordering). Again we call this matrix the *expansion matrix* at ∞ of $\tilde{v}_1, \ldots, \tilde{v}_r$.

We now need to generalize some notions from subsubsection 2.5.3.1 on degree-reduced matrices: We define the *valuation-degree* of a Laurent series in $\mathrm{x}^{-\frac{1}{e}}$ with leading coefficient $\mathrm{x}^{\frac{a}{e}}$ to be $\frac{a}{e}$. With this definition we generalize the definitions of subsubsection 2.5.3.1 to Laurent series in $\mathrm{x}^{-\frac{1}{e}}$.

Now Hensel and Landsberg call the system $\tilde{v}_1, \ldots, \tilde{v}_r$ *normal* at ∞ if the matrix A defined above satisfies $\mathrm{valdeg}(\det(A)) = \sum_{j=1}^{r} \mathrm{valdeg}(\underline{a}_j)$. An equivalent condition is the matrix $\mathrm{lc}(A)$ of leading coefficients of A is non-singular.[8] Moreover, it is easy to see that this condition implies that for all

[8] More generally, they define a notion of a normal $k(\mathrm{x})$-basis with respect to any point

$\underline{b} \in k[\mathrm{x}]^r$, $\mathrm{valdeg}(A\underline{b}) = \max_{j=1}^r \mathrm{valdeg}(b_j \underline{a}_j)$.

Hensel and Landsberg show:

Every $k[\mathrm{x}]$-submodule of $k(\mathcal{C})$ of rank r has a $k[\mathrm{x}]$-basis which is normal at ∞.

This has the following application for Riemann-Roch spaces: Let still D be a divisor on $\mathcal{C}$ which has no support "at infinity", and let still $I := I^{\mathrm{fin}}(-D)$. Let $v_1, \ldots, v_r$ be normal at ∞. Let $M \in k((\mathrm{x}^{-\frac{1}{e}}))$ be the expansion matrix at ∞ and let $\underline{b} \in k[\mathrm{x}]^r$. Then $\sum_j b_j v_j$ lies in $L(D)$ if and only if $\mathrm{valdeg}(M\underline{b}) \leq 0$. Now this is the case if and only if $\mathrm{valdeg}(b_i) \leq -\mathrm{valdeg}(\underline{m}_i)$ for all $i = 1, \ldots, r$. With $d_i := -\lceil \mathrm{valdeg}(\underline{m}_i) \rceil$, this implies that a basis for $L(D)$ is given by $\mathrm{x}^j v_i$ for $0 \leq j \leq d_i$.

We now give a proof that every $k[\mathrm{x}]$-submodule $\mathfrak{M}$ of $k(\mathcal{C})$ of full rank has a basis which is normal at ∞.

Let t_i $(i = 1, \ldots, h)$ be the uniformizing element of $k(\mathcal{C})_{\mathfrak{p}_i}$ which is mapped to $\mathrm{x}^{-\frac{1}{e_i}}$ under $\sigma_{i,1}$. Then $1, t_i, \ldots, t_i^{e_i - 1}$ form a $k[[\mathrm{x}^{-1}]]$-basis of $\widehat{\mathfrak{O}}_{\mathfrak{p}_i}$. Let now B be the ordered basis of $k(\mathcal{C}) \otimes_{k(\mathrm{x})} k((\mathrm{x}^{-1}))$ whose elements correspond to $(\delta_{k,\ell} t_\ell^{j-1})_\ell$ for $k = 1, \ldots, h, j = 1, \ldots, e_i$ under isomorphism (2.14) (ordered lexicographically where the first index is dominating).

Let now $v_1, \ldots, v_r$ be a $k[\mathrm{x}]$-basis of $\mathfrak{M}$ such that the following holds: Let N be the coordinate matrix of $v_1, \ldots, v_r$ with respect to the basis B. Then $\mathrm{lc}(N)$ is an upper triangular matrix of the form

$$\begin{pmatrix} 1 & * & * & \cdots & * \\ & 1 & * & \cdots & * \\ & & 1 & \cdots & * \\ & & & \ddots & * \\ & & & & 1 \end{pmatrix} .$$

Such a basis always exists. This follows for example from an obvious generalization of the reduction algorithm for $>_d$ in the proof of Proposition 2.26 to Laurent series and a final permutation.

We claim that $v_1, \ldots, v_r$ is then also normal at ∞.

The embeddings $\sigma_{i,j}$ induce an isomorphism of $k((\mathrm{x}^{-\frac{1}{e}}))$-vector spaces

$$k(\mathcal{P})_{\mathfrak{p}_i} \otimes_{k((\mathrm{x}^{-1}))} k((\mathrm{x}^{-\frac{1}{e}})) \xrightarrow{\sim} k((\mathrm{x}^{-\frac{1}{e}}))^r \ . \tag{2.18}$$

Let C be the coordinate matrix of the basis B with respect to the basis of $k(\mathcal{C}) \otimes_{k(\mathrm{x})} k((\mathrm{x}^{-\frac{1}{e}}))$ which corresponds to the standard basis of $k((\mathrm{x}^{-\frac{1}{e}}))^r$.

---

of $\mathbb{P}^1_k$.

Then C has the form

$$\begin{pmatrix} C_1 & & & \\ & C_2 & & \\ & & \ddots & \\ & & & C_h \end{pmatrix},$$

where the matrix C_i is an $e_i \times e_i$-matrix whose (a,j)-entry is $\zeta_{e_i}^{(a-1)\cdot(j-1)} \mathrm{x}^{\frac{(j-1)}{e_i}}$.

Let A be the expansion matrix at ∞ of $v_1, \ldots, v_r$. Then $A = CN$.

Let now $N = \begin{pmatrix} N_1 \\ \vdots \\ N_h \end{pmatrix}$, where N_i has e_i rows. Then A has the form

$$\begin{pmatrix} C_1N_1 & * & * & * \\ & C_2N_2 & * & * \\ & & \ddots & * \\ & & & C_hN_h \end{pmatrix}.$$

As all elements of column j of C_i have degree $\frac{j-1}{e_i}$ and $\mathrm{lc}(N_i)$ is upper triangular with unity on the diagonal, $\mathrm{lc}(C_iN_i) = \mathrm{lc}(C_i) = ((\zeta_{e_i}^{(a-1)\cdot(j-1)}))_{a,j}$. In particular, $\mathrm{lc}(C_iN_i)$ is non-singular. Now by the shape of A, $\det(\mathrm{lc}(A)) = \det(\mathrm{lc}(C_1N_1)) \cdots \det(\mathrm{lc}(C_iN_i))$, thus $\mathrm{lc}(A)$ is also non-singular. This establishes the claim.

A warning We would like to issue a *warning* at this point: It is not true in general that every non-singular matrix in $k((\mathrm{x}^{-\frac{1}{e}}))^{r\times r}$ can with $k[\mathrm{x}]$-column operations be transformed into a matrix whose matrix of leading coefficients is non-singular.

An obvious generalization of degree-reducedness is as follows: A matrix in $k((\mathrm{x}^{-\frac{1}{e}}))$ is *degree-reduced* with respect to x if for all $\underline{b} \in k[\mathrm{x}]^r$, $\mathrm{valdeg}(A\underline{b}) = \max_{j=1}^{r} \mathrm{valdeg}(b_j \underline{a}_j)$. As already stated the matrix of a normal system is degree reduced. It is however not true in general that the matrix of leading coefficients of a non-singular degree-reduced matrix is non-singular. Consider for example the matrix

$$\begin{pmatrix} \mathrm{x}+1 & \mathrm{x}^{\frac{1}{2}} \\ \mathrm{x}^{\frac{1}{2}} & 1 \end{pmatrix},$$

which is degree-reduced with respect to x.

With an obvious "reduction algorithm" one can however prove that for any matrix $A \in k((\mathrm{x}^{-\frac{1}{e}}))^{r\times s}$ there exists a degree-reduced matrix which is related to A by $k[\mathrm{x}]$-column operations.

What concerns the determination of the Riemann-Roch space $L(D)$, clearly instead of a basis of I which is normal at ∞ in the sense of Hensel and Landsberg one might just consider an ideal basis whose matrix is degree-reduced. This approach is for example pursued by W. Schmidt in [Sch91]. More information on matrices in $k((\mathrm{x}^{-\frac{1}{e}}))$ which are degree-reduced with respect to x can also be found in [Heß01].

2.5.5 The global representation

One can represent effective divisors by linear subspaces of the spaces of global sections of invertible sheaves. The theoretical background of this representation is given by the following lemma.

Lemma 2.99 *Let $\mathcal{L}$ be an invertible sheaf of degree $\geq 2g+1$ on $\mathcal{C}$. Then the space $\Gamma(\mathcal{C},\mathcal{L})$ has dimension $\deg(\mathcal{L})-g+1$. For an effective divisor D on $\mathcal{C}$ of degree $\leq \deg(\mathcal{L})-2g$, $\mathcal{L}(-D)$ is base point free, the space $\Gamma(\mathcal{C},\mathcal{L}(-D))$ has codimension $\deg(D)$ in $\Gamma(\mathcal{C},\mathcal{L})$, and D is the base divisor of $(\mathcal{L},\Gamma(\mathcal{C},\mathcal{L}(-D)))$. In particular, D is uniquely determined by $\Gamma(\mathcal{C},\mathcal{L}(-D))$ inside $\Gamma(\mathcal{C},\mathcal{L})$.*

Proof. We have $\deg(\mathcal{L}(-D)) = \deg(\mathcal{L}) - \deg(D) \geq 2g$. This implies that $\mathcal{L}(-D)$ is base point free; see [Har77, IV, Corollary 3.2]. In particular both $\mathcal{L}$ and $\mathcal{L}(-D)$ are non-special. Now the dimension statements follow from the Riemann-Roch Theorem. Let $D' \geq D$ be the base divisor of $(\mathcal{L},\Gamma(\mathcal{C},\mathcal{L}(-D)))$. We then have $\Gamma(\mathcal{C},\mathcal{L}(-D')) \simeq \Gamma(\mathcal{C},\mathcal{L}(-D))$. Now if $D' \neq D$, then $\mathcal{L}(-D)$ would not be base point free. □

One can therefore represent an effective divisor D by the pair $(\mathcal{L},\Gamma(\mathcal{C},\mathcal{L}(-D)))$, where $\mathcal{L}$ is an invertible sheaf of degree $\geq 2g+\deg(D)$. We call this representation of effective divisors the *joint global representation*. The *free global representation* is the free representation of divisors obtained by restricting this representation to prime divisors and then applying the usual construction for a free representation.

For computational purposes one can for example choose a divisor E and set $\mathcal{L} := \Gamma(\mathcal{C},\mathcal{O}(E))$. Then one can represent $\Gamma(\mathcal{C},\mathcal{O}(E)) = L(E)$ by a k-basis consisting of elements in the function field $k(\mathcal{C})$, and one can represent a basis of $\Gamma(\mathcal{C},\mathcal{O}(E-D)) = L(E-D)$ by their coordinate matrix with respect to the basis of $\Gamma(\mathcal{C},\mathcal{O}(E))$.

The following lemma is helpful if one wants to change between the ideal and the global representation.

Lemma 2.100 *Let B be a base point free divisor on $\mathcal{C}$. Then $L(B)$ generates the fractional ideal $I^{\mathrm{fin}}(-B)$ as $((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k)$–module. Similarly,*

it generates the fractional ideal corresponding to the restriction of $-B$ *to* $(x_{|\mathcal{C}})^{-1}(\frac{1}{\mathrm{x}}(\mathbb{A}^1_k))$ *as* $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\frac{1}{\mathrm{x}}(\mathbb{A}^1_k))$*-module.*[9]

Proof. Clearly $L(B)$ is contained in both ideals (it is in fact the intersection of the two ideals). Now let $P \in \mathcal{C}$. Then as B is base point free by assumption, $L(B)$ generates the fractional ideal $\mathfrak{m}_P^{v_P(B)}$ as an $\mathcal{O}_{\mathcal{C},P}$-module. This implies the statements. □

Remark 2.101 In the context of Lemma 2.99, let $\mathcal{L} = \mathcal{O}(E)$ for a divisor E on $\mathcal{C}$. Then the divisor $E - D$ is base point free, and Lemma 2.100 in particular applies to this divisor.

We now discuss the change from the joint ideal to the joint global representation with respect to sheaves of the form $\mathcal{O}(n \cdot (x_{|\mathcal{C}})_-)$ and conversely. For this we restrict ourselves to curves with plane models such that $x_{|\mathcal{C}}$ is a separating element. We assume that the finite and infinite orders as well as the $k[\mathrm{x}]$-invariants and a basis $v_1, \ldots, v_r$ as in Proposition 2.95 of the trivial divisor have already been computed. Note that these computations can be performed in a number of field and bit operations which is polynomially bounded in d by Propositions 2.72 and 2.95.

Proposition 2.102 *Under the above assumptions the following holds:*

Given an effective divisor D *in joint ideal representation and* $n \in \mathbb{N}$ *such that* $n \cdot r \geq 2g + \deg(D)$ *and such that* n *is polynomially bounded in* d *and* $\deg(D)$*, one can compute the coordinate representation of a basis of* $L(n \cdot (x_{|\mathcal{C}})_- - D)$ *with respect to the basis of* $L(n \cdot (x_{|\mathcal{C}})_-)$ *described in Proposition 2.92 in a number of field and bit operations which is polynomially bounded in* d *and* $\deg(D)$*.*

Conversely, given an effective divisor D *represented by the corresponding subspace* $L(n \cdot (x_{|\mathcal{C}})_- - D)$ *of* $L(n \cdot (x_{|\mathcal{C}})_-)$*, where* n *is as above, one can compute the joint ideal representation of* D *in a number of field and bit operations which is polynomially bounded in* d *and* $\deg(D)$*.*

Proof. For the first claim, we proceed as follows: First the divisor $-D$ is computed (see Proposition 2.78). Then a basis of $L(n \cdot (x_{|\mathcal{C}})_- - D)$ as in Proposition 2.95 is computed. Now, the elements of this basis are expressed in terms of the basis of $L(n \cdot (x_{|\mathcal{C}})_-)$ via a linear algebra computation. Note that the sizes of the matrices to be considered are polynomially bounded in d and $\deg(D)$ because the degrees of the denominators and the numerator matrices of the bases of the two spaces are polynomially bounded in d and $\deg(D)$ (see again Proposition 2.95).

[9] If B has support outside the preimages of the points ∞ and 0 on $\mathbb{P}^1_k$, this ideal is not equal to $I^\infty(-B)$.

We come to the second claim. We first compute $(x_{|\mathcal{C}})_- - D$ in joint ideal representation: The "finite" ideal of the ideal representation of $(x_{|\mathcal{C}})_- - D$ can be computed as claimed by Lemma 2.100 / Remark 2.101 and Proposition 2.72 (on the computation of ideals from generators). In the same way, one can compute the ideal I (that is, a HNF-basis thereof) of $((x_{|\mathcal{C}})_* \mathcal{O}_{\mathcal{C}})(\frac{1}{\mathrm{x}}(\mathbb{A}^1_k))$ which corresponds to the restriction of $n \cdot (x_{|\mathcal{C}})_- - D$ to $(x_{|\mathcal{C}})^{-1}(\frac{1}{\mathrm{x}}(\mathbb{A}^1_k))$. At this point we do not have yet a joint ideal representation of $(x_{|\mathcal{C}})_- - D$ as we require that the "infinite ideal" only has support "at infinity". We can however compute the desired infinite ideal by Proposition 2.80 as claimed.

Now D can be computed as claimed by Proposition 2.78. □

Given that we represent the curve $\mathcal{C}$ by a plane model $\mathcal{C}_{pm}$, another natural choice for $\mathcal{L}$ is $\mathcal{O}_{\mathcal{C}}(n) = \pi^*(\mathcal{O}_{\mathbb{P}^2_k}(n))$.

Note that $\mathcal{O}_{\mathcal{C}}(n) \simeq \mathcal{O}(n \cdot \mathrm{div}(Z_{|\mathcal{C}}))$ (via division by $Z^n_{\mathcal{C}}$). In particular, if $\mathrm{div}(X_{|\mathcal{C}})$ and $\mathrm{div}(Z_{|\mathcal{C}})$ have disjoint support then $\mathcal{O}_{\mathcal{C}}(n) \simeq \mathcal{O}(n \cdot (x_{|\mathcal{C}})_-)$.

It is however more natural to not perform this division by $Z^n_{|\mathcal{C}}$ and rather consider the following approach. We consider the inclusion into the associated sheaf of meromorphic sections $\mathcal{O}_{\mathcal{C}}(n) \hookrightarrow \mathcal{M}(\mathcal{O}_{\mathcal{C}}(n)) \stackrel{\mathrm{Def}}{=} \mathcal{O}_{\mathcal{C}}(n) \otimes_{\mathcal{O}_{\mathcal{C}}} \mathcal{M}_{\mathcal{C}}$. This inclusion induces an inclusion $\Gamma(\mathcal{C}, \mathcal{O}_{\mathcal{C}}(n)) \hookrightarrow \Gamma(\mathcal{C}, \mathcal{M}(\mathcal{O}_{\mathcal{C}}(n))) \simeq \mathrm{Quot}(k[X,Y,Z]/(F))_n$ (the degree n part of $\mathrm{Quot}(k[X,Y,Z]/(F))$).

Explicitly, the space $\Gamma(\mathcal{C}, \mathcal{O}_{\mathcal{C}}(n))$ as a subspace of $\mathrm{Quot}(k[X,Y,Z]/(F))_n$ is $Z^n \cdot L(n \cdot \mathrm{div}(Z_{|\mathcal{C}}))$, and this allows an efficient computation of a basis of $\Gamma(\mathcal{C}, \mathcal{M}(\mathcal{O}_{\mathcal{C}}(n)))$ inside $\mathrm{Quot}(k[X,Y,Z]/(F))_n$ via Heß' algorithm. (Of course, Z can be substituted by any other linear form.)

Proposition 2.103

a) Given $\mathcal{C}$ (represented by $\mathcal{C}_{pm}$) and $n \in \mathbb{N}$ there exists a homogeneous element $H \in k[X,Y,Z]/(F)$ whose total degree is polynomially bounded in d such that $H \cdot \Gamma(\mathcal{C}, \mathcal{O}_{\mathcal{C}}(n)) \subseteq k[X,Y,Z]/(F)$.

b) One can compute such an element H and a basis of $H \cdot \Gamma(\mathcal{C}, \mathcal{O}_{\mathcal{C}}(n))$ inside $(k[X,Y,Z]/(F))_{n+\deg(H)}$ in a number of field and bit operations which is polynomially bounded in d and n.

c) Let us assume that $x_{|\mathcal{C}}$ is a separating element and that the finite and infinite order have been computed. Let us furthermore assume that an element H and a basis $B_1, \ldots, B_\ell$ of $H \cdot \Gamma(\mathcal{C}, \mathcal{O}_{\mathcal{C}}(n))$ as in b) have been computed. We represent linear subspaces of $\Gamma(\mathcal{C}, \mathcal{O}_{\mathcal{C}}(n))$ via bases which we represent by their coordinate vectors with respect to $\frac{B_1}{H}, \ldots, \frac{B_\ell}{H}$. Then for divisors of degree $\leq n \cdot r - 2g$, one can change between the joint ideal representation of divisors and the representation via subspaces of $\Gamma(\mathcal{C}, \mathcal{O}_{\mathcal{C}}(n))$ in a number of field and bit operations which is polynomially bounded in d, $\deg(D)$ and n.

Proof. The first statement follows immediately from Proposition 2.95 applied to $D = \mathrm{div}(Z_{|\mathcal{C}})$.

For b) one can use this algorithm:

1. Set up the ideal representation with respect to $x_{|\mathcal{C}}$ or $y_{|\mathcal{C}}$ (depending on whether $x_{|\mathcal{C}}$ is a separating element).
2. Compute $\mathrm{div}(Z_{|\mathcal{C}})$ in ideal representation.
3. Apply Heß' algorithm to compute a basis $v_1, \ldots, v_{\dim(L(n \cdot Z_{|\mathcal{C}}))}$ of $L(n \cdot Z_{|\mathcal{C}})$
4. Write the basis in the form $v_i = \frac{G_i}{H}$ with homogeneous elements $H, G_i \in k[X,Y,Z]/(F)$.
5. Output H and $G_i \cdot Z^n$ for all i.

By Proposition 2.53, Step 1 can be performed with a number of field and bit operations which is polynomially bounded in d. For Step 2, note that as $V(X,Y,Z) = \emptyset$ $\mathrm{div}(Z_{|\mathcal{C}}) = \max\{(x_{|\mathcal{C}})_-, (y_{|\mathcal{C}})_-\}$. Furthermore $(x_{|\mathcal{C}})_- = \max\{-(x_{|\mathcal{C}}), 0\}$ and $(y_{|\mathcal{C}})_- = \max\{-(y_{|\mathcal{C}}), 0\}$. Therefore one can compute $\mathrm{div}(Z_{|\mathcal{C}})$ (deterministically) in a number of field and bit operations which is polynomially bounded in d by Proposition 2.66. Step 3 can be performed as claimed by Proposition 2.95, and the forth step can also be performed as claimed.

For the last statement, see the proof of Proposition 2.102. □

Remark 2.104 Let $\mathcal{L}$ be any invertible sheaf on $\mathcal{C}$, let D_1 and D_2 be effective divisors on $\mathcal{C}$, and let s be a non-trivial global section of $\mathcal{L}$ such that $\mathrm{div}(s) = D_1 + D_1'$ for some effective divisor D_1'. Then multiplication by s gives an isomorphism

$$\mathcal{O}(D_1 - D_2) \longrightarrow \mathcal{L}(-D_1' - D_2) \,,$$

which induces an isomorphism

$$L(D_1 - D_2) \longrightarrow \Gamma(\mathcal{C}, \mathcal{L}(-D_1' - D_2)) \,.$$

Now in [KM04b] K. Khuri-Makdisi gives an efficient algorithm based on easy linear algebra computations for the following computational problem; see [KM04b, Theorem / Algorithm 5.5]:

Let $\mathcal{L}$ be some invertible sheaf, and let us fix bases of $\Gamma(\mathcal{C}, \mathcal{L})$, $\Gamma(\mathcal{C}, \mathcal{L}^{\otimes 2})$ and $\Gamma(\mathcal{C}, \mathcal{L}^{\otimes 3})$. For the computation the only information on $\mathcal{L}$ we require is the matrix of the multiplication maps

$$\Gamma(\mathcal{C}, \mathcal{L}) \otimes \Gamma(\mathcal{C}, \mathcal{L}) \longrightarrow \Gamma(\mathcal{C}, \mathcal{L}^{\otimes 2})$$

and

$$\Gamma(\mathcal{C}, \mathcal{L}) \otimes \Gamma(\mathcal{C}, \mathcal{L}^{\otimes 2}) \longrightarrow \Gamma(\mathcal{C}, \mathcal{L}^{\otimes 3})$$

with respect to the fixed bases. Now let D_1, D_2 be two effective divisors with $\deg(D_i) \geq 2g+1$ and $\deg(D_i) \leq \deg(\mathcal{L}) - 2g - 1$, each represented in joint global representation with respect to $\mathcal{L}$, thus via the subspaces $\Gamma(\mathcal{C}, \mathcal{L}(-D_i))$ of $\Gamma(\mathcal{C}, \mathcal{L})$.

The problem is now to compute a global section s of $\mathcal{L}$ as above and the subspace $\Gamma(\mathcal{C}, \mathcal{L}(-D_1' - D_2))$ of $\Gamma(\mathcal{C}, \mathcal{L})$, where D_1' is as above. Moreover, if $|D_1 - D_2|$ is non-empty, an effective divisor linearly equivalent to $D_1 - D_2$ in global representation with respect to $\mathcal{L}$ shall be computed.

Note that if $\mathcal{L} = \mathcal{O}(E)$, where E is some divisor on $\mathcal{C}$, one can regard the section s as a function on $\mathcal{C}$, and one then obtains $L(D_1 - D_2) = \frac{1}{s} \cdot \Gamma(\mathcal{C}, \mathcal{L}(-D_1' - D_2))$. Thus if the fixed basis of $\Gamma(\mathcal{C}, \mathcal{L}) = L(E)$ in $k(\mathcal{C})$ is also given, one obtains a basis of the space $L(D_1 - D_2)$.

Another result by Khuri-Makdisi concerns the computation of a free representation. We already know that one can in an efficient way pass from a joint to a free ideal representation; see Proposition 2.86. So one can also pass from a joint to a free global representation via the ideal representation. In [KM04a, Sections 6 and 7] it is shown how one can pass directly from a joint to a free global representation.

2.6 Representing divisor classes and computations in the class group

Let still $\mathcal{C}$ be a curve over a field k, represented by a plane model $\mathcal{C}_{pm}$. Let g be the genus of $\mathcal{C}$.

We are now considering the task to represent divisor classes and compute with them.

We have already discussed several ways to represent divisors, and we have discussed algorithmic aspects related to divisors as well. The most straightforward representation of divisor classes would thus arguably be to simply use the canonical homomorphism $\mathrm{Div}(\mathcal{C}) \longrightarrow \mathrm{Cl}(\mathcal{C})$ as a representation of the divisor class group by the divisor group.

The problem with this approach is however that even for divisors of degree 0 on a fixed curve $\mathcal{C}$, the length of the input (the divisor used for the representation) would be arbitrarily large.

This problem can however be avoided as by the Riemann-Roch Theorem every divisor of degree $\geq g$ is linearly equivalent to an effective divisor.

Let us fix a divisor D_0 of degree ≥ 1.

The following definition is [Heß01, Definition 8.1].

Definition 2.105 Let $\tilde{D}$ be an effective divisor on $\mathcal{C}$. Then $\tilde{D}$ is *(maximally) reduced along* D_0 if the linear system $|\tilde{D} - D_0|$ is empty.

Remark 2.106 Let $\tilde{D}$ be an effective divisor which is reduced along D_0. Then

- $\dim(L(\tilde{D})) \leq \deg(D_0)$, that is, $\dim_k(|\tilde{D}|) < \deg(D_0)$.
- $\deg(\tilde{D}) < g + \deg(D_0)$.

Both statements follow immediately from the conditions and the Riemann-Roch Theorem.

Definition 2.107 Let now D be any divisor on $\mathcal{C}$, and let $\tilde{D}$ be an effective divisor reduced along D_0 such that $D \sim \tilde{D} + rD_0$ for some $r \in \mathbb{Z}$. Then $\tilde{D}$ is called a *reduction* of D along D_0.

Lemma 2.108 *Let D be a divisor on $\mathcal{C}$. Then there exists a reduction of D along D_0. The set of reductions of D along D_0 forms a complete linear system. Moreover, it depends only on the divisor class of D (and on D_0).*

Proof. Note first that the divisor $D + (\lceil \frac{g}{\deg(D_0)} \rceil - \deg(D))D_0$ has degree $\geq g$, thus in particular it is linear equivalent to an effective divisor. If now r is minimal such that $|D + rD_0|$ is non-empty, then $D + rD_0$ is linearly equivalent to a reduced divisor.

Obviously, if $\tilde{D}$ is a reduction, then so is every divisor linear equivalent to $\tilde{D}$. For the converse let $\tilde{D}$ and $\tilde{D}'$ be two reductions with $\tilde{D} - rD_0 \sim D \sim \tilde{D}' - r'D_0$. Let wlog. $r \geq r'$. Then $\tilde{D} - (r - r')D_0 \sim \tilde{D}'$. Therefore $r = r'$ as otherwise $\tilde{D}$ would not be reduced.

The last statement is obvious. □

The first point of Remark 2.106 and this lemma imply:

Lemma 2.109 *Let* $\deg(D_0) = 1$. *Then the reduction of a divisor D on $\mathcal{C}$ along D_0 is unique.*

Proposition 2.110 *Given a divisor in joint ideal representation, one can compute with a deterministic algorithm a reduction of a divisor D on $\mathcal{C}$ along D_0 in a number of field and bit operations which is polynomially bounded in d,* $\deg(D_0)$ *and* $\mathrm{ht}(D)$.

Proof. This follows from the proof of Lemma 2.108 and Propositions 2.78 and 2.97. □

Remark 2.111 Let us consider the corresponding computational problem over finite fields. Then given a curve $\mathcal{C}$ over $\mathbb{F}_q$ and two divisors D and D_0 as above one can with a randomized algorithm compute a uniformly randomly distributed reduction of D along D_0 in a time which is polynomially bounded in $\log(q)$, d, $\deg(D_0)$ and $\mathrm{ht}(D)$.

This follows from Remark 2.98.

Every divisor class $a \in \mathrm{Cl}(\mathcal{C})$ can be represented by a tuple consisting of a reduction $\tilde{D}$ of any divisor defining a and the number r with $[\tilde{D}]-r[D_0] = a$. For classes in $\mathrm{Cl}^0(\mathcal{C})$, one can omit the number r.

Proposition 2.112 *Let us represent divisor classes as described, where the divisors are given in joint ideal representation. Then given two divisor classes $a, b \in \mathrm{Cl}(\mathcal{C})$, one can compute the sum $a+b \in \mathrm{Cl}(\mathcal{C})$ and the difference $a - b \in \mathrm{Cl}(\mathcal{C})$ in a number of field operations which is polynomially bounded in d and $\mathrm{ht}(D_0)$ and in a number of bit operations which is polynomially bounded in $d, \deg(D_0), \log(\deg(a))$ and $\log(\deg(b))$.*

Proof. This follows from Proposition 2.78 on divisor arithmetic and Proposition 2.110. □

Remark 2.113 Note that the height of the canonical divisor $\mathrm{div}(\mathrm{d}x_{|\mathcal{C}})$ on $\mathcal{C}$ considered in subsubsection 2.5.4.4 is $\max\{2r, \deg(R)\}$ with the notations of that subsubsection. Therefore, this divisor has a height which is polynomially bounded in d. By applying the proposition to this divisor, one obtains that the arithmetic in $\mathrm{Cl}^0(\mathcal{C})$ can be performed in a number of field and bit operations which is polynomially bounded in d.

By Lemma 2.109 we obtain a particularly nice representation of the divisor class group if we have a divisor of degree 1 and height polynomially bounded in d.

For arbitrary curves such a divisor does not always exist. For example, by the Riemann-Roch theorem, a curve of genus 1 has a divisor of degree 1 if and only if it has a rational point. However, not every genus 1 curve has a rational point.

On the other hand, every curve over a finite field does have a divisor of degree 1.

Proposition 2.114 *Given a curve over a finite field $\mathbb{F}_q$, one can compute with a randomized algorithm a divisor of degree 1 and height in $\mathcal{O}(\log(d))$ in joint or free ideal representation in an expected time which is polynomially bounded in d and $\log(q)$.*

Proof. Assume we have some $r_0 \in \mathbb{N}$ and points $Q_+ \in \mathcal{C}(\mathbb{F}_{q^{r_0+1}})$ and $Q_- \in \mathcal{C}(\mathbb{F}_{q^{r_0}})$. Let P_+ and P_- be the corresponding closed points of $\mathcal{C}$. Let f_+ be the residue field degree of P_+ and f_- the residue degree of P_-. Then $f_+ | r_0 + 1$ and $f_- | r_0$. Let $e_+ := \frac{r_0+1}{f_+}$ and $e_- := \frac{r_0}{f_-}$. Then $e_+P_+ - e_-P_-$ is a divisor of degree 1. The height of the divisor is $\leq r_0 + 1$.

We now have to prove that
- one can always find such points Q_+ and Q_- in an expected time which is polynomially bounded in d and $\log(q)$,
- one can at the same time choose r_0 to be in $\mathcal{O}(\log(d))$
- given Q_+ and Q_-, one can compute the divisor of degree 1 in the claimed time.

Recall that by the Hasse-Weil bound, for $r \in \mathbb{N}$, we have

$$\#\mathcal{C}(\mathbb{F}_{q^r}) \geq q^r + 1 - 2g\, q^{r/2} \, .$$

Therefore,

$$\#\mathcal{C}_{ns}(\mathbb{F}_{q^r}) \geq q^r + 1 - 2gq^{r/2} - (\frac{(d-1)(d-2)}{2} - 1) \, ,$$

where $\mathcal{C}_{ns}$ is the non-singular part of $\mathcal{C}_{pm}$.

Let $r_0 := \lceil 4 \log_2(2d) \rceil$, and let $r \geq r_0$. Then $q^{r/2} \geq 2^{r_0/2} \geq (2d)^2 = 4d^2$, and therefore $\frac{1}{4}q^r \geq d^2 q^{r/2} > 2gq^{r/2}$ and $\frac{1}{4}q^r \geq d^2 q^{r/2} > \frac{(d-1)(d-2)}{2}$. Thus $\#\mathcal{C}_{ns}(\mathbb{F}_{q^r}) \geq \frac{q^r}{2}$.

By the following lemma, if one intersects $(\mathcal{C}_{pm})_{\mathbb{F}_{q^r}}$ with a uniformly distributed random line in $\mathbb{P}^2_{\mathbb{F}_{q^r}}$, the probability that the line contains a point from $\mathcal{C}(\mathbb{F}_{q^r})$ is $\geq \frac{1}{2d}$.

We thus proceed as follows: For $r = r_0$ and $r_0 + 1$ we intersect $\mathcal{C}_{\mathbb{F}_{q^r}}$ with uniformly randomly chosen lines in $\mathbb{P}^2_{\mathbb{F}_{q^r}}$. We compute all $\mathbb{F}_{q^r}$-rational points in the intersection, and for each such point we check if it lies in $\mathcal{C}_{ns}$. We repeat this until we have found the desired points Q_+ and Q_-. After we have found the two points Q_+ and Q_-, we compute P_+ and P_- and finally $e_+P_+ - e_-P_-$.

Let us check that these steps can indeed by carried out with the claimed complexity. First, r_0 can easily be computed. It is also easy to choose a line in $\mathbb{P}^2_{\mathbb{F}_{q^r}}$ uniformly at random. The computation of the intersection can be performed as claimed by Lemma 2.116 below. The check if the points lie in $\mathcal{C}_{pm}$ can be performed by evaluating the partial derivatives of $F(X, Y, Z)$. We already know that the expected number of lines we have to consider for $r = r_0$ and $r = r_0 + 1$ is $\leq 2d$. The computation of P_+ and P_+ and finally $e_+P_+ - e_-P_-$ is possible with the claimed complexity by Propositions 2.90 and 2.78. □

Lemma 2.115 *Let $\#\mathcal{C}_{ns}(\mathbb{F}_q) \geq \frac{q}{2}$. Now let L be a uniformly distributed random line in $\mathbb{P}^2_{\mathbb{F}_q}$. Then the probability that L intersects $\mathcal{C}_{pm}$ in at least one point in $\mathcal{C}_{ns}(\mathbb{F}_q)$ is larger than $\frac{1}{2d}$.*

Proof. Each point in $\mathbb{P}^2(\mathbb{F}_q)$ (in particular each element from $\mathcal{C}_{ns}(\mathbb{F}_q)$) lies on $q+1$ lines. Moreover, on each line lie at most d elements from $\mathcal{C}_{ns}(\mathbb{F}_q)$. This means that $\geq \frac{1}{d}(q+1) \cdot \frac{q}{2}$ lines contain an element from $\mathcal{C}_{ns}(\mathbb{F}_q)$.

In total there are q^2+q+1 lines. Therefore, the probability that a uniformly randomly distrubted line contains an element from $\mathcal{C}_{ns}(\mathbb{F}_q)$ is $> \frac{1}{2d}$. □

Lemma 2.116 *Given a non-trivial homogeneous polynomial $F(X,Y,Z) \in \mathbb{F}_q[X,Y,Z]$ and a non-trivial linear homogeneous polynomial $G(X,Y,Z) \in \mathbb{F}_q[X,Y,Z]$, one can compute the set of $\mathbb{F}_q$-rational points in $V(F,G)$ in an expected number of field and bit operations which is polynomially bounded in* $\deg(F)$ *and* $\log(q)$.

Proof. Let $G = a_X X + a_Y Y + a_Z Z$. Let us wlog. assume that $a_Y = 1$. Then the point $(0:1:0)$ does not lie in $V(F,G)$, and the scheme-theoretic image of $V(F,G)$ under the projection to the (X,Z)-coordinates is given by $F(X, -a_X X - a_Z Z, Z) \in \mathbb{F}_q[X,Z]$. One can therefore perform the computation as follows: One factors this polynomial, and for each linear factor one computes the corresponding Y-coordinate via G. □

The computations of discrete logarithms with the index calculus method the following proposition is crucial.

Proposition 2.117 *Let us consider curves over finite fields. As always we represent curves by plane models, and let d be the degree of the plane model. We represent divisor classes by divisors which are reduced along a divisor D_0 of degree 1 and of a height which is polynomially bounded in d. Moreover, we represent divisors in* free ideal representation. *Then the arithmetic (addition and subtraction) in the degree 0 class groups can we a randomized algorithm be performed in an expected time which is polynomially bounded in d and $\log(q)$, where q is the cardinality of the ground field. If the curves are represented by plane models of degree $\mathcal{O}(g)$, one can with a randomized algorithm perform the arithmetic in the degree 0 class groups in an expected time which is polynomially bounded in g and $\log(q)$; one can then with a randomized algorithm perform the arithmetic in an expected time which is polynomially bounded in $\log(\#\operatorname{Cl}^0(\mathcal{C}))$.*

Proof. The first statement follows from Proposition 2.112 and Proposition 2.87, the second statement is an immediate consequence, and the third

statement follows from the second statement and the inequality $\#\,\mathrm{Cl}^0(\mathcal{C}) \geq (\sqrt{q}-1)^{2g}$. □

Recall here that by Proposition 2.6 ([Heß05, Theorem 56]), any curve over a finite field has a plane model of degree $\mathcal{O}(g)$. Thus one can represent curves over finite fields in such a way that the second and the third statement hold for all curves.

Remark 2.118 If instead of the free ideal representation one uses the joint ideal representation, statements analogous to the ones in the proposition hold with a deterministic algorithm.

2.7 Appendix: Computing the maximal order

We address in this section the problem to compute the order $((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k)$ of $k(\mathcal{C})$, given the homogeneous polynomial $F(X,Y,Z)$ defining the plane model, or given $f(x,y) \in k[x,y]$ or the associated monic polynomial $\tilde{f}(\mathrm{x},\tilde{y}) \in k[\mathrm{x}][\tilde{y}]$.

Recall that we usually assume that $\mathcal{C}$ is geometrically irreducible. All the following considerations however also apply if $\mathcal{C}$ is only a smooth and proper irreducible k-scheme. Equivalently, for the following considerations we not assume that $k(\mathcal{C})|k$ is a regular field extension.

Moreover, in subsection 2.5.4, we assumed that x is a separating element. We drop this assumption as well.

For separable extensions $k(\mathcal{C})|k(\mathrm{x})$ it was shown by Chistov that the complexity of the related problem for finite fields $k = \mathbb{F}_q$ are polynomially bounded in the input length (in a bit-oriented model such as on a bit-RAM or on a Turing machine).

If $p \leq d$, his algorithm relies however on factorization of univariate polynomials over k (he uses Berlekamp's algorithm). Therefore his algorithm does not lead to a deterministic algorithm in our computational models but only to a randomized algorithm. Another aspect of Chistov's approach is the use of expansions whereas we strive for expansion free methods.

We present here a completely different algorithm which relies heavily on HNF-bases and which does not use factorization of polynomials. As a side effect, we can remove the prerequisite that the field extension be separable.

The algorithm can be seen as a variant of several algorithms already present in the literature. However, to our knowledge, it has not been previously shown that one can compute the order $((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k)$ with a number of field and bit operations which is polynomially bounded in d without polynomial factorization (but with the ability to compute p^{th} roots). This

also applies to algorithms which can be obtained by obvious variations of algorithms for the computation of maximal orders in number fields. In particular, contrary to the statement in Section 3 of [Heß01], an obvious variant for function fields of the algorithm by Buchmann and Lenstra ([BL94]) does require factorization of polynomials in the same way as the algorithm by Chistov does.

At the end of this section, some further historical remarks on the algorithm can be found.

We now first describe some theoretical background of the algorithm in a more abstract setting. The following two lemmata and Proposition 2.121 are essentially the content of [GR71, Anhang §3, 3, Sätze 6,7]. We follow closely [GLS01] which in turn is based on [dJ98].

Let A be a commutative reduced noetherian ring and let I be an ideal of A, containing a non-zero-divisor u. Let $\mathrm{Quot}(A)$ be the total quotient ring of A, and let $\tilde{A}$ be the normalization of A.

Lemma 2.119 *The homomorphism of A-modules*

$$\{g \in \mathrm{Quot}(A) \mid gI \subseteq A\} \longrightarrow \mathrm{Hom}_A(I, A)\ ,\ g \mapsto (h \mapsto hg)$$

is an isomorphism. Likewise is the restriction

$$\{g \in \mathrm{Quot}(A) \mid gI \subseteq I\} \longrightarrow \mathrm{Hom}_A(I, I)\ .$$

Proof. The homomorphism $\mathrm{Hom}_A(I, A) \longrightarrow \{g \in \mathrm{Quot}(A) \mid gI \subseteq A\}$ $\varphi \mapsto \frac{\varphi(u)}{u}$ is clearly an inverse to the first homomorphism. The second statement is then obvious. □

We set

$$A' := \{g \in \mathrm{Quot}(A) \,|\, gI \subseteq I\}\ .$$

Lemma 2.120 *A' is a commutative ring and*

$$A \subseteq A' \subseteq \{g \in \mathrm{Quot}(A) \cap \tilde{A} \,|\, gI \subseteq A\} \subseteq \{g \in \mathrm{Quot}(A) \,|\, gI \subseteq \sqrt{I}\}\ .$$

In particular, if $I = \sqrt{I}$, then

$$A' = \{g \in \mathrm{Quot}(A) \cap \tilde{A} \,|\, gI \subseteq A\}\ .$$

Proof. It is obvious that A' is a commutative ring and that $A \subseteq A'$.

For the second inclusion, we have to show that $A' \subseteq \tilde{A}$. By the Cayley-Hamilton Theorem for finitely generated modules ([Eis95, Theorem 4.3]), the endomorphism $I \longrightarrow I\ ,\ h \mapsto hg$ satisfies a monic polynomial with coefficients in A. As the homomorphism $A' \longrightarrow \mathrm{End}_A(I)$ is injective, so does g.

For the last inclusion, let $g \in \mathrm{Quot}(A) \cap \tilde{A}$ with $gI \subseteq A$. Let $t^n + a_{n-1}t^{n-1} + \cdots + a_0$ be a monic polynomial in $A[t]$ which is satisfied by g. Now let $h \in I$. Then $(gh)^n = -a_{n-1}(gh)^{n-1} \cdot h - \cdots - a_0 \cdot h^n \in I$. Hence $gh \in \sqrt{I}$. □

Let

$$\mathrm{NN}(A) := \{\mathfrak{p} \in \mathrm{Spec}(A) \mid A_\mathfrak{p} \text{ is not normal}\} .$$

be the *non-normal locus* of A. Note that $\mathrm{NN}(A)$ is the closed subset of $\mathrm{Spec}(A)$ defined by the *conductor ideal* $\mathfrak{c} := \mathrm{Ann}_A(\tilde{A}/A)$.

We now have the following *criterion for normality.*

Proposition 2.121 *Let A be a reduced noetherian commutative ring, and let $I \subseteq A$ be an ideal satisfying*

a) I is a radical ideal

b) $\mathrm{NN}(A)$ is contained in the closed subset of $\mathrm{Spec}(A)$ defined by I.

Then $A = \tilde{A}$ if and only if $A = A'$, where A' is defined as above.

Proof. If $A = \tilde{A}$ then $A = A'$ by the lemma above.

So let the three conditions be satisfied, and let $A = A'$. Then by b) $I \subseteq \sqrt{\mathfrak{c}}$. As A is noetherian, there exists a minimal $j \in \mathbb{N}_0$ with $I^j \subseteq \mathfrak{c}$, that is, $I^j\tilde{A} \subseteq A$. If $j = 0$, the result follows.

So let us assume that $j > 0$. Let $h \in \tilde{A}$ and $g \in I^{j-1}$ such that $hg \notin A$. Then $hg \in \tilde{A}$ and $hgI \subseteq hI^j \subseteq A$, thus by the previous lemma $gh \in A' = A$. This is a contradiction. We conclude that $j = 0$, thus $A = \tilde{A}$. □

The following formula gives a convenient description of A' for algorithmic purposes.

$$A' = \frac{1}{u} \cdot (I :_A uI) \tag{2.19}$$

For a *proof* note first that

$$A' \subseteq \frac{1}{u}I \subseteq \frac{1}{u}A \tag{2.20}$$

Indeed, as $u \in I$, we have $u \cdot A' \subseteq I$ by definition of A'.

The inclusion $\frac{1}{u}(I :_A uI) \subseteq A'$ is obvious. So let $g \in A'$. Then $ug \in A$ and $ugI \subseteq uI$. Therefore $ug \in (I :_A uI)$. □

We now return to our application, the computation of the finite order $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$. Let us first fix the following definition.

Definition 2.122 An *order* of $k(\mathcal{C})$ *with respect to* x or *with respect to* $k[\mathrm{x}]$ is an a subring of $k(\mathcal{C})$ which contains $k[\mathrm{x}]$ and is finitely generated over $k[\mathrm{x}]$.

Remark 2.123 A subring A of $k(\mathcal{C})$ is an order with respect to $k[\mathrm{x}]$ if and only if A contains x and A is contained in $((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k)$.

Proposition 2.121 and Lemma 2.120 and Equation (2.19) imply:

Proposition 2.124 *Let $\mathcal{O}$ be an order of $k(\mathcal{C})$ with respect to $k[\mathrm{x}]$. Let $u \in k[\mathrm{x}] - \{0\}$, and let I be an ideal of $\mathcal{O}$ containing u. Let*

$$\mathcal{O}' := \{g \in k(\mathcal{C}) \mid gI \subseteq I\} .$$

Then

a) $\mathcal{O}'$ is an order with $\mathcal{O} \subseteq \mathcal{O}' \subseteq \frac{1}{u}\mathcal{O}$.

b) Let U be the kernel of

$$\mathcal{O} \longrightarrow \mathrm{End}_k(I/(u)) \ , \ g \mapsto (\overline{h} \mapsto \overline{gh}) .$$

Then $\mathcal{O}' = \frac{1}{u}U$.

c) Let now $u \in k[\mathrm{x}]$ be such that every prime divisor of $[((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k) : \mathcal{O}]_{k[\mathrm{x}]}$ also divides u. (For example, u might be a multiple of the discriminant of $\mathcal{O}$.) Let $I := \sqrt{u \cdot \mathcal{O}}$ be the radical of the ideal $u \cdot \mathcal{O}$ in $\mathcal{O}$. Then $\mathcal{O} = \mathcal{O}'$ if and only if $\mathcal{O} = ((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k)$.

Remark 2.125 In the classical terminology of number fields, $\mathcal{O}'$ would be called "the order associated to the module I".

Proof of the proposition. Part a) follows immediately from (2.20). Part b) is a reformulation of (2.19).

For part c) note that the prime divisors of $\mathfrak{c} \cap k[\mathrm{x}]$ (where $\mathfrak{c}$ is the conductor of $\mathcal{O}$) are equal to the prime divisors of $[((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k) : \mathcal{O}]_{k[\mathrm{x}]}$. Therefore the element u in the proposition is contained in $\sqrt{\mathfrak{c}}$. The result now follows from Proposition 2.121. □

We are now coming to the algorithm. Recall that the task is to compute the order $((x_{|\mathcal{C}})_*\mathcal{O}_{\mathcal{C}})(\mathbb{A}^1_k)$, given the homogeneous polynomial $F(X, Y, Z)$ defining the plane model or the polynomial $f(\mathrm{x}, y) \in k[\mathrm{x}, y]$ or the associated monic polynomial $f(\mathrm{x}, \tilde{y}) \in k[\mathrm{x}][\tilde{y}]$.

Convention 2.126 *In the following we represent all $k[\mathrm{x}]$-submodules of $k(\mathrm{x})$ of rank r by the corresponding HNF-bases with respect to $1, y_{|\mathcal{C}}, \ldots, y_{|\mathcal{C}}^{r-1}$. As usual, here the HNF-basis is represented by the denominator and the coordinate matrix in $k[\mathrm{x}]^{r\times r}$.*

The algorithm is based on successive enlargements of orders until one reaches the maximal order $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$ and is very easy to describe:

We start with $\mathcal{O} = k[\mathrm{x}][\tilde{y}]/(\tilde{f})$. Then we successively substitute $\mathcal{O}$ by $\mathcal{O}'$, where $\mathcal{O}'$ is as in Proposition 2.124 with $u = \mathrm{disc}(\tilde{f})$. We terminate if $\mathcal{O} = \mathcal{O}'$.

By Proposition 2.124 it is clear that we obtain an algorithm which computes the finite order, provided that we can compute $\mathcal{O}'$ from $\mathcal{O}$.

What concerns the complexity, note that we always have $\deg([((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k) : \mathcal{O}']_{k[\mathrm{x}]}) < \deg([((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k) : \mathcal{O}]_{k[\mathrm{x}]})$. As $\deg([((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k) : k[\mathrm{x}][\tilde{y}]/(\tilde{f})]_{k[\mathrm{x}]}) \leq \frac{\deg(\mathrm{disc}(\tilde{f}))}{2}$, we need at most $\lfloor\frac{\deg(\mathrm{disc}(\tilde{f}))}{2}\rfloor$ substitutions, thus the number of substitutions is polynomially bounded in d.

By the next lemma, we obtain the desired result:

Proposition 2.127 *One can compute the finite order $((x_{|\mathcal{C}})_*\mathcal{O}_\mathcal{C})(\mathbb{A}^1_k)$ of $k(\mathcal{C})$ in a number of field and bit operations which is polynomially bounded in d.*

Lemma 2.128 *Given a polynomial $u(\mathrm{x})$ which divides $\mathrm{disc}(\tilde{f})$ and an order $\mathcal{O}$ of $k(\mathcal{C})$ with respect to $k[\mathrm{x}]$, one can with a deterministic algorithm compute $\mathcal{O}'$ in a number of field and bit operations which is polynomially bounded in d.*

Proof. Let $v_1, \ldots, v_r$ be the HNF-basis of $\mathcal{O}$ with respect to $1, \tilde{y}_{|\mathcal{C}}, \ldots, y^{r-1}_{|\mathcal{C}}$.

We use the formula $\mathcal{O}' = \frac{1}{u}U$ in part b) of Proposition 2.124.

As above, let I be the radical of (u) in $\mathcal{O}$. Note that $I/(u)$ is the nilradical of the local finite k-algebra $\mathcal{O}/(u)$.

We can therefore proceed as follows: First we compute the multiplication table of $\mathcal{O}/(u)$ with respect to the canonical basis of $\mathcal{O}/(u)$ (cf. Definition 2.43) (with respect to $v_1, \ldots, v_r$). Secondly we compute a basis $\overline{h}_1, \ldots, \overline{h}_k$ of the nilradical (that is, we compute coordinate vectors of the elements $\overline{h}_i$ with respect to the canonical basis). Then we compute the multiplication table of $\mathcal{O}/(u) \times I/(u) \longrightarrow I/(u)$ with respect to the canonical basis of $\mathcal{O}/(u)$ and the basis of $I/(u)$ just computed.

Let for $i, j = 1, \ldots, k$, $\varphi_{i,j} : I/(u) \longrightarrow I/(u)$ be the endomorphism given by $\overline{h}_i \longrightarrow \delta_{i,j}\overline{h}_j$. Note that the multiplication table immediately gives the matrix of the homomorphism $\mathcal{O}/(u) \longrightarrow \mathrm{End}_k(I/(u))$, $\overline{g} \mapsto (\overline{h} \mapsto \overline{gh})$ with respect to the canonical basis of $\mathcal{O}/(u)$ and the basis $(\varphi_{i,j})_{i,j}$.

Finally, we compute a basis $\overline{b}_1, \ldots, \overline{b}_\ell$ of this matrix. Then U is generated as a $k[\mathrm{x}]$-module by the canonical lifts of $\overline{b}_1, \ldots, \overline{b}_\ell$ to $\mathcal{O}$ and $uv_1, \ldots, uv_r$ (see again Definition 2.43 and note that the lifting is computationally trivial).

We obtain the final result by a Hermite normal form computation.

As for the complexity, note first that $\deg(u)$ is polynomially bounded in d (because $\operatorname{disc}(\tilde{f})$ is polynomially bounded in d). The canonical complementary system of (u) in $\mathcal{O}$ has $\deg(u) \cdot r$ elements (which is polynomially bounded in d) each of which has degree $< \deg(u)$. It is then clear that the multiplication table can be computed in a number of field and bit operations which is polynomially bounded in d. By Proposition 1.50 the computation of the nilradical can then be performed (with a deterministic algorithm) in a number of field and bit operations which is polynomially bounded in d.

The computation of $\overline{b}_1, \ldots, \overline{b}_\ell$ involves linear algebra over k with matrices whose size is polynomially bounded in d. Finally, the Hermite normal form computation takes place on a matrix of size polynomially bounded in d and degree polynomially bounded in d. All these computations can be performed in a number of field and bit operations which is polynomially bounded in d. □

Some historical remarks

The computational problem to determine maximal orders in function fields is closely related to two well-studied computational problems: The computation of maximal orders in number fields and the computation of normalizations of arbitrary finitely generated commutative k-algebras.

For both these problems, various algorithms have been proposed, and the algorithm above can be seen as a variant of algorithms for both these problems. We try to give here an account of the historical development as far as it is relevant for the present algorithm.

We start with the problem to compute maximal orders of number fields. Key ideas of the algorithm presented above were already present in an algorithm developed in the 1960ies by H. Zassenhaus. It seems that the algorithm was not formally published by him for a long time. The algorithm does however appear in a proceeding of an workshop of the "Mathematisches Forschungsinstitut Oberwolfach" from 1967 (see [Zas67]). In this work Zassenhaus in fact considers the following more general problem: He defines an *order* as a (not necessarily commutative) ring O which is as $\mathbb{Z}$-module free of finite rank. The computational problem is now: Given such an order (represented by its multiplication table), compute an order which is contained in $O \times_{\mathbb{Z}} \mathbb{Q}$, contains O, and is maximal!

As a special case of the results for arbitrary rings, the following criterion is stated for orders of number fields (see pp. 102-103 of [Zas67]): Let p be a prime, and let O be an order in a number field. Then the *p-radical* of O is defined as $\sqrt{p \cdot O}$. Now the index of O in the maximal order is divisible by p (one says "O is p-maximal") if and only if the order associated to the

p-radical of O is equal to O.

This leads to the following algorithm to compute maximal orders in number fields, starting with any order O: First the discriminant of O is factored. Then one successively enlarges the order along the following procedure: One iterates over all prime factors which divide the discriminant at least twice, and for each such factor p one proceeds as follows: Let us assume that we have already constructed the order $\mathcal{O}$. Then one successively substitutes $\mathcal{O}$ with $\mathcal{O}'$, where $\mathcal{O}'$ is the order associated to the p-radical in $\mathcal{O}$. One moves on the next prime if $\mathcal{O} = \mathcal{O}'$.

The algorithm in [Zas67] is a bit more complex but this is only because Zassenhaus considers non-commutative orders.

We make several remarks here: First Zassenhaus' criterion on "p-maximality" of orders can be viewed as a "local version" of Proposition 2.124. In fact, the criterion can be viewed as a special case of Proposition 2.121 (namely, one sets $A := \mathcal{O}_p$). We note also that the Zassenhaus' criterion is incorrectly attributed to both M. Pohst and H. Zassenhaus by H. Cohen in his book [Coh96].

Zassenhaus' algorithm immediately leads an algorithm for the computation of maximal orders in function fields, provided one assumes that factorization of univariate polynomials is possible. The algorithm we presented above can then be viewed as a variant of this algorithm which avoids factorization of polynomials.

In the meantime several variants of Zassenhaus' algorithm have been given. We mention in particular the *Round 2 algorithm* and the algorithm by Buchmann and Lenstra.

The Round 2 algorithm is based on Zassenhaus' algorithm but additionally the so-called *Dedekind criterion* is used: Let $f(t) \in \mathbb{Z}[t]$ be irreducible, and let the order $\mathcal{O}$ of $\mathbb{Q}[t]/(f(t))$ be such that $[\mathcal{O} : \mathbb{Z}[t]/(f(t))]$ is prime to p. Then the Dedekind criterion is a simple criterion with which one can decide if p does not divide the index of $\mathcal{O}$ in the maximal order. Moreover, if this is not the case, one can easily compute the enlarged order $\mathcal{O}'$. The Round 2 algorithm can for example be found in [Coh96].

Before we come to the algorithm by Buchmann and Lenstra we remark that in 1989 Chistov published a work on the relationship between the computation of commutative maximal orders ([Chi89]) and factorization. He claims that the computation of maximal orders in number fields is polynomially equivalent to determine the largest square-free divisor of an integer. Building on the fact that the corresponding computational problem for polynomials over finite fields is algorithmically easy, he also claims that the maximal order of a global function field $L|\mathbb{F}_q$ respect to a separable field extension $L|\mathbb{F}_q(\mathrm{x})$ and the ring $\mathbb{F}_q[\mathrm{x}]$ can be computed in a time which is

polynomially bounded in the input length. This algorithm is based on a previous algorithm of his for computations of approximations of factorizations of polynomials over local fields ([Chi87]). As the method used by Chistov is completely different than Zassenhaus' method, and there is no relationship to the algorithm presented above, we do not describe his algorithm here.

We just remark how in Chistov's algorithm for global function fields factorization of polynomials is used, respectively avoided: For global function fields whose characteristic is less-or-equal to the extension degree, he uses Berlekamp's algorithm to factor the discriminant. In the other case, Chistov uses what might be called "passive factorization" of the discriminant: He considers divisors of the discriminant, and he computes modulo these divisors as if he was computing in a field. If now one hits during the execution of the algorithm some element which is not invertible, one has obtained a further factorization of the discriminant, and one proceeds recursively. This approach means that in our computational models, the algorithm by Chistov would require randomization in the "small characteristic" case.

We also mention that Chistov's work is very brief, and some steps of the algorithm are not described at all. From the wording in the article by Chistov, the author of this work has the impression that these missing steps might be given in previous works. But unfortunately, such works are not cited by Chistov.

Now inspired by the result by Chistov, J. Buchmann and H.W. Lenstra revisited Zassenhaus' ideas in an article from 1994 (see [BL94]). Just like Chistov, they employed the idea of "passive factorization". They are only concerned with the number field case, but their algorithm can easily be adapted to global function fields. We remark again that just as Chistov's work, an obvious generalization of the algorithm to arbitrary ground fields would require polynomial factorization for the "small characteristic case" (see Step 1 of Algorithm 6.6 in [BL94]).

We are now coming to the second computational problem, the computation of normalizations of arbitrary commutative k-algebras. After various previous works by several authors, T. de Jong gave in 1998 an algorithm for the computation of normalizations of arbitrary finitely generated commutative algebras over fields. (See the citations in [dJ98] and [GLS01] for an overview over previous algorithms.) In fact, de Jong realized that one can easily build an algorithm on the previous (purely theoretical) criterion for normality by Grauert and Remmert ([GR71]) from the year 1971 which can essentially be found above.

De Jong has a representation by generators and relations in mind. The algorithm we have presented above can be seen as a variant of the algorithm by de Jong for the special case of orders in function fields and with HNF-

bases.

It is interesting to note that despite their similarity the work by Zassenhaus and subsequent works based on it on the one hand and the work by Grauert, Remmert and finally de Jong on the other hand seem to be completely independent:

As already stated, Zassenhaus' criterion on "p-maximality" can be viewed as a special case of results by Grauert and Remmert. We found however no indication that Zassenhaus or Grauert and Remmert were aware of this. Also, it seems that various authors who later worked on variants of Zassenhaus' algorithm were not aware of the results by Grauert and Remmert.

As stated, de Jong's work builds on the work by Grauert and Remmert. However, it seems that he was also ignorant of the fact that a variant of his algorithm was already given by Zassenhaus before Grauert and Remmert published their work. The same applies to Greuel, Lossen and Schulze ([GLS01]) who in the introduction to their overview article from the year 2001 write on the computation of normalizations: "The problem [...] is to know when to stop, that is, to have an effective criterion for a ring to be normal. It had escaped the computer algebra community that such a criterion has been known for more than thirty years, having been discovered by Grauert and Remmert. It was rediscovered by de Jong."

Chapter 3

Computing discrete logarithms

3.1 Introduction

We now come to the heart of this work: The computation of discrete logarithms in degree 0 class groups of curves, including the groups of rational points of elliptic curves, over finite fields.

All results on the computation of discrete logarithms we present follow the so-called *index calculus method*. Very briefly, the index calculus method consists of the following: One fixes a set of prime divisors $\mathcal{F}$, called the *factor base*, and one searches for *relations* between the factor base elements and the input elements. If one has enough relations, one tries to derive the discrete logarithm by linear algebra.

An outline over this chapter is as follows: In the next section we present the index calculus method. In particular, we give a "general algorithm" which relies on five subroutines, namely a subroutine for the computation of the group order, a subroutine for integer factorization, a subroutine for construction of a factor base and for precomputation, a subroutine for relation generation, and a subroutine for sparse linear algebra. Let us note here that the "precomputation" we mentioned might in particular include the construction of a so-called tree of large prime relations. Later in this chapter we then give subroutines for the discrete logarithm problem for specific classes of curves.

In the following section we discuss an index calculus algorithm with double large prime variation for curves of a fixed genus. In Section 3.4 we focus on the following computation problem: Let a natural number g_0 be fixed. Now the task is to compute discrete logarithms of elements of degree 0 class groups of arbitrary curves of genus $\geq g_0$ over finite fields. Then main result Sections 3.3 and 3.4 is arguably that one can compute discrete

logarithms in (suitably represented) degree 0 class groups of curves $\mathcal{C}/\mathbb{F}_q$ of genus at least 3 over finite fields in an expected time of $\tilde{\mathcal{O}}(\#\operatorname{Cl}^0(\mathcal{C}))^{\frac{4}{9}})$.

Finally, in the last section, we present and analyze index calculus algorithms for elliptic curves over extension fields. These algorithms rely on what we call a *decomposition algorithm* which in turn is based in solving systems of multigraded polynomials. We then in particular show that there exists an infinite family of increasing finite fields such that the elliptic curve discrete logarithm problem over these fields can be solved in subexponential time in the input length.

If not stated otherwise, the computational model for deterministic (resp. randomized) algorithms is the bit-RAM model (resp. randomized bit-RAM model).

3.2 The index calculus method

In this section we give a general outline to the index calculus method.

3.2.1 The setting

Let us first fix the following general definition:

Definition 3.1 Let G be a finite group (written multiplicatively), and let $a \in G$ and $b \in \langle a \rangle$. Then the *discrete logarithm* of b *with respect* to a is the smallest non-negative integer x with $a^x = b$. Here a is called the *base of the logarithm*. One writes $x = \log_a(b)$.

Remark 3.2 In the "Disquisitiones Arithmeticae", C.F. Gauß introduced the name *index* for what is now called the discrete logarithm of some element $\mathbb{F}_p^*$ (p any prime number) with respect to a multiplicative generator of $\mathbb{F}_p$. This classical terminology was for example still used by J. Pollard in [Pol78], but nowadays it seems to be not so common anymore.

Convention 3.3 *As we are concerned with the computation of discrete logarithms in degree 0 class groups of curves over finite fields, we use additive notation in the following. This means: Given a finite abelian group G, written additively, $a \in G$ and $b \in \langle a \rangle$, the discrete logarithm of b with respect to a is the smallest non-negative integer x with $x \cdot a = b$.*

We are interested in the discrete logarithm problem in degree 0 class groups of curves over finite fields. This means that given a curve $\mathcal{C}$ over a finite field $\mathbb{F}_q$ and two elements $a, b \in \operatorname{Cl}^0(\mathcal{C})$ with $b \in \langle a \rangle$, we want to compute the discrete logarithm of b to the base a. Here we assume that the

curve and the divisor classes are represented as in Chapter 2: The curve is given by a plane model, which in turn is given by a homogeneous polynomial $F(X, Y, Z)$; a divisor D_0 of degree 1 is fixed, and the divisor classes are given by divisors which are reduced along D_0; all these divisors are given by any of the representations described in the previous chapter.

We now give an overview over one variant of the so-called index calculus method, assuming – for this introductory description – that the degree 0 class group is cyclic and generated by a.

Let $x \in \{0, \ldots, \mathrm{ord}(a) - 1\}$ with $x \cdot a = b$ be the unknown discrete logarithm we want to compute.

One first chooses a so-called *factor base* $\mathcal{F}$ consisting of prime divisors of $\mathcal{C}$,[1] and one enumerates the elements of $\mathcal{F}$, say $\mathcal{F} = \{F_1, F_2, \ldots, F_k\}$ with pairwise distinct F_j. Furthermore, one chooses a divisor D_1 of degree 1. (One might set $D_1 := D_0$ but this is not necessary.)

Then one generates *relations* between the input elements a, b, the elements of the factor base and D_1:

$$\sum_j r_{i,j}[F_j] - (\sum_j r_{i,j} \deg(F_j)) \cdot [D_1] = \alpha_i a + \beta_i b \tag{3.1}$$

with $r_{i,j}, \alpha_i, \beta_i \in \mathbb{Z}$.[2]

If one has generated more than k relations, one can – under certain additional assumptions – determine the unknown discrete logarithm x. The basic idea is here to eliminate the factor base elements via linear algebra such that one obtains a relation between the input elements a and b.

In the cases we consider we can always compute the group order and its factorization in a way which is (asymptotically) not time-critical. Then all computations can be performed "modulo the group order and its factors". We describe this now in greater detail.

Let $N := \#\,\mathrm{Cl}^0(\mathcal{C})$. One then proceeds as outlined above: One fixes a factor base and one generates relations. However, now the coefficients $r_{i,j}, \alpha_i, \beta_i$ lie in $\mathbb{Z}/N\mathbb{Z}$.

Let us now for simplicity for the moment assume that N is prime. Let now $k+1$ relations be generated; let $R = ((r_{i,j}))_{i,j} \in (\mathbb{Z}/N\mathbb{Z})^{(k+1)\times k}$ be the *relation matrix*, that is, the matrix whose rows are the coefficients occurring in the relations.

Now the left-kernel of the matrix R is non-trivial. One computes some non-trivial row vector $\underline{\gamma} \in (\mathbb{Z}/N\mathbb{Z})^{1\times(k+1)}$ with $\underline{\gamma}R = 0$. Note that this

[1] In this section, we always speak of prime divisors instead of closed points, in order to highlight the factorization of divisors into prime divisors.

[2] Let $D_1, \ldots, D_n$ be divisors and $c_1, \ldots, c_m$ be divisor classes on $\mathcal{C}$. Then a *relation* between $D_1, \ldots, D_n$ and $c_1, \ldots, c_m$ is a tuple $(r_1, \ldots, r_n, s_1, \ldots, s_m) \in \mathbb{Z}^{m+n}$ with $\sum_j r_j[D_j] = \sum_j s_j c_j$. We always describe relations by giving the corresponding equation.

means that

$$\sum_i \gamma_i r_{i,j} = 0 \tag{3.2}$$

for all $j = 1, \ldots, k$. We now have

$$\begin{array}{l} \sum_i \gamma_i \alpha_i a + \sum_i \gamma_i \beta_i b = \sum_i \gamma_i (\alpha_i a + \beta_i b) = \\ \sum_i \sum_j \gamma_i r_{i,j}[F_j] = \sum_j (\sum_i \gamma_i r_{i,j})[F_j] = 0 \ . \end{array} \tag{3.3}$$

If now $\sum_i \gamma_i \beta_i \neq 0$, then with $\xi := (\sum_i \gamma_i \alpha_i)\,(\sum_i \gamma_i \beta_i)^{-1} \in \mathbb{Z}/N\mathbb{Z}$, we have

$$\xi \cdot a = b \in \mathrm{Cl}^0(\mathcal{C}) \ .$$

This means that the unique representative $x \in \{0, \ldots, N-1\}$ of ξ is the discrete logarithm of b with respect to a we are looking for.

Of course, in order to put this general idea to work, one needs algorithms to obtain a factor base, to compute relations as well as an algorithm to perform the linear algebra.

In subsection 3.2.3 below we give a framework for the computation of discrete logarithms in degree 0 class groups of curves over finite fields. We give a general algorithm which relies subroutines for

a) computation of the order of the degree 0 class group

b) integer factorization

c) construction of a factor base and for precomputation

d) relation generation.

e) sparse linear algebra.

Specifications of the inputs and outputs of these subroutines are given in subsection subsection 3.2.3. Then later in this chapter, specific subroutines are discussed for specific classes of curves. In fact, one obtains very different algorithms depending on which subroutines one chooses. The algorithms for integer factorization as well we for linear algebra are however the same in all the following algorithms, and can be discussed already now.

For integer factorization we use the algorithm by H.W. Lenstra and C. Pomerance ([LP92]), which gives rise to the following result.

Proposition 3.4 *An integer N can with a randomized algorithm be factored in an expected time of $L_N[\frac{1}{2}, 1 + o(1)]$.*

3.2.2 Sparse linear algebra

After the relation collection, we will have obtained a matrix M in sparse representation over $\mathbb{Z}/N\mathbb{N}$, where $N := \mathrm{Cl}^0(\mathcal{C})$, and we want to compute some row vector $\underline{\gamma}$ over $\mathbb{Z}/N\mathbb{Z}$ with $\underline{\gamma}M = 0$ and $[\underline{\gamma}]_\ell \neq 0$ for all prime divisors ℓ of N.

Let $N = \prod_{i=1}^{v} \ell_i^{e_i}$ be the factorization of N, where ℓ_i are pairwise distinct prime numbers and e_i natural numbers. If we then have found row vectors $\underline{\gamma}^{(i)}$ over $\mathbb{Z}/\ell^{e_i}\mathbb{Z}$ with $\underline{\gamma}\,[M]_{\ell_i^{e_i}} = 0$ and $\underline{\gamma}^{(i)} \neq 0$, we can use the Chinese Remainder Theorem to obtain a solution $\underline{\gamma}$ as desired.

It remains to determine such vectors $\underline{\gamma}^{(i)}$. This computation will be based on a computation over the finite field $\mathbb{F}_{\ell_i}$ (discussed below) and an obvious "lifting algorithm" which is described in the following lemma.

In the rest of this subsubsection, as usual, we describe all linear algebra computations on column vectors.

As outlined above, in the index calculus algorithms, we will however use operations on row vectors.

Lemma 3.5 *Let ℓ be a prime number, and let $A \in (\mathbb{Z}/\ell^e\mathbb{Z})^{m\times n}$ for some $e, m, n \in \mathbb{Z}$. Let now $\underline{v} \in (\mathbb{Z}/\ell^e\mathbb{Z})^n$ with $[A\underline{v}]_{\ell^{e-1}} = 0$; let $\underline{b} \in (\mathbb{Z}/\ell\mathbb{Z})^m$ with $A\underline{v} = \ell^{e-1}\underline{b}$. If now there exists a $\underline{u} \in (\mathbb{Z}/\ell\mathbb{Z})^n$ with $A\underline{u} = \underline{b}$, then we have $A(\underline{v} - \ell^{e-1}\underline{u}) = 0$. Moreover, every $\underline{w} \in (\mathbb{Z}/\ell\mathbb{Z})^n$ with $A\underline{w} = 0$ can be obtained in this way.*

Note that the lifting algorithm is guaranteed to succeed — starting from any row vector over $\mathbb{F}_\ell$ — if $[A]_\ell$ has *full row rank*; otherwise it might fail. Because of this, our goal in the relation generation — where we use the corresponding result for row vectors — is to produce a matrix which has *full column rank* modulo all prime divisors of the group order.

It remains to give an efficient algorithm for solving sparse linear systems over finite fields and for computing non-trivial solutions of homogeneous systems, provided that the matrix has full row rank. For this, one can use various variants of Wiedemann's and Lanczos' algorithms. An algorithm as we require it can in fact already be found in the original work by Wiedemann. Indeed, from [Wie86] one obtains (with fast multiplication and division in finite fields):

Proposition 3.6 *There is a randomized algorithm such that the following holds: Upon input of a matrix $A \in \mathbb{F}_q^{m\times n}$ in sparse representation and a vector $\underline{b} \in \mathbb{F}_q^m$, the algorithm outputs either "failure" or a vector $\underline{v} \in \mathbb{F}_q^n$ with $A\underline{v} = \underline{b}$. Moreover, if the matrix A has full row rank, the algorithm outputs a solution $\underline{v}$ with a probability of at least $\frac{1}{2}$. The running time of*

the algorithm is in $\tilde{\mathcal{O}}(n \cdot (n+\omega) \cdot \log(q))$, where ω is the number of non-zero entries of A.

Alternatively, one might also use the variant of Wiedemann's algorithm described in [KS91] or the variant of Lanczos' algorithm from [EK97] together with the remarks on necessary field extensions in [EG02, Section 4]. One can then substitute the condition that A has full column rank by the condition that $\underline{b}$ lies in the column space of A.

It is well-known that one can obtain an algorithm as in the proposition with the additional property that if the algorithm does not fail, the output is uniformly distributed over the kernel of A.

For this, one first chooses $\underline{w} \in \mathbb{F}_q^n$ uniformly randomly, and then one applies the algorithm to A and $\underline{b} - A\underline{w}$ (for $\underline{b}$). If $\underline{v}$ is the random vector computed by the algorithm, then we have $A(\underline{v} + \underline{w}) = \underline{b} - A\underline{w} + A\underline{w} = \underline{b}$.

Let us recall why the result $\underline{v}+\underline{w}$ is uniformly distributed over the kernel of A:

Let $\underline{y} := A\underline{w}$. Let us fix any value $\underline{y}^{(0)} \in \mathbb{F}_q^m$ of $\underline{y}$ (occurring with non-trivial probability). Then conditionally to $\underline{y} = \underline{y}^{(0)}$, the following holds: $\underline{w}$ is uniformly distributed in the preimage of $\underline{y}^{(0)}$ upon multiplication by A. Moreover, all random choices in the algorithm are stochastically independent of $\underline{w}$. Therefore, the output $\underline{v}$ is stochastically independent of $\underline{w}$. Thus conditionally to any value $\underline{v}^{(0)}$ of $\underline{v}$ (which occurs with non-trivial probability) (and still conditionally to $\underline{y} = \underline{y}^{(0)}$), $\underline{w}$ is still uniformly distributed in the preimage of $\underline{y}^{(0)}$, and then $\underline{v}^{(0)} + \underline{w}$ is uniformly distributed in the preimage of $\underline{b}$.

But if the result holds conditionally to any fixed value of $\underline{y}$ and $\underline{v}$, it also holds unconditionally.

Note that if we apply the algorithm just described to a matrix A of full column rank with non-trivial kernel and the vector $\underline{0}$, then with a probability of $\geq \frac{1}{2} \cdot (1 - \frac{1}{q}) \geq \frac{1}{4}$ it outputs a non-trivial vector in the kernel of A. By applying the algorithm up to three times we obtain:

Proposition 3.7 *There is a randomized algorithm such that the following holds: Upon input of a matrix $A \in \mathbb{F}_q^{m \times n}$ in sparse representation with non-trivial kernel, the algorithm outputs either "failure" or a* non-trivial *vector $\underline{v} \in \mathbb{F}_q^m$ with $A\underline{v} = 0$. Moreover, if the matrix A has full row rank, the algorithm outputs a vector $\underline{v}$ with a probability of at least $\frac{1}{2}$. The running time of the algorithm is in $\tilde{\mathcal{O}}(n \cdot (n+\omega) \cdot \log(q))$, where ω is the number of non-zero entries of A.*

Altogether we have:

Proposition 3.8 *There is a randomized algorithm such that the following holds: Upon input of a matrix $A \in (\mathbb{Z}/N\mathbb{Z})^{m\times n}$ in sparse representation and the factorization of N, the algorithm determines some $\underline{v} \in (\mathbb{Z}N\mathbb{Z})^m$ with $A\underline{v} = 0$ and $[\underline{v}]_\ell \neq 0$ for every prime divisor ℓ of N or it reports "failure". Moreover, if for all prime divisors ℓ of N the matrix $[A]_\ell$ has full row rank, the probability that it reports "failure" is $\leq \frac{1}{2}$. The running time of the algorithm is in $\tilde{\mathcal{O}}(n \cdot (n+\omega) \cdot \log(N)^2)$, where ω is the number of non-trivial entries in the matrix.*

Proof. As above, let $N = \prod_{i=1}^{v} \ell_i^{e_i}$ be the factorization of N. Note that $\sum_i e_i \log_2(\ell_i) = \log_2(N)$, and therefore $\sum_i e_i \leq \log_2(N)$.

As outlined above, we proceed for each prime power individually and then use the Chinese Remainder Theorem to obtain a solution over $\mathbb{Z}/N\mathbb{Z}$. For each prime, we start off with an algorithm satisfying Proposition 3.7 and then we apply the lifting algorithm above with an algorithm satisfying Proposition 3.6. In each of these cases, we repeat the computation up to $\lceil \log_2 \log_2(N) + 1 \rceil$ times if the any algorithm reports "failure". Like this, we obtain: If the matrix has full column rank, then the total success probability is $\geq (1 - (\frac{1}{2})^{\lceil \log_2 \log_2(N)+1 \rceil})^{\log_2(N)} \geq (1 - \frac{1}{2\log_2(N)})^{\log_2(N)} \geq 1 - \frac{1}{2} = \frac{1}{2}$. □

3.2.3 The general algorithm

As already noted, in this subsection we give a framework for computations of discrete logarithms in degree 0 class groups of curves over finite fields. We outline a general algorithm relying on various subroutines. This general algorithm is closely related to the framework for computations of discrete logarithms in groups with known group order given in [EG02] (see subsection 3.2.5 for further remarks on the relationship).

The setting is as follows: We fix a particular full subcategory of the large groupoid of instances $(\mathcal{C}/\mathbb{F}_q, a, b)$ of the discrete logarithm problem in degree 0 class groups of curves over finite fields (cf. Example 1.22). The inputs to the algorithm will consist of these instances together the group order and divisor classes $c_1, \ldots, c_u$ of $\mathrm{Cl}^0(\mathcal{C})$.[3] Here the curve and the divisor classes are represented as described above, but additionally we allow that certain restrictions on the allowed representations imposed. For example, in the applications of the general framework we will always assume that the degree of the plane model is in $\mathcal{O}(g)$, where g is the genus of the curve.

Roughly, in the algorithm, we first compute the group order and factor it. Then we set up the ideal representation with respect to the covering

[3] For the moment we put no restriction on the elements $c_1, \ldots, c_u$. However, in applications of the "general algorithm" we will only be able to derive upper bounds on expected running times if we assume that $c_1, \ldots, c_u$ is a generating system.

$x_{\mathcal{C}} : \mathcal{C} \longrightarrow \mathbb{P}^1_{\mathbb{F}_q}$ or the covering $y_{|\mathcal{C}} : \mathcal{C} \longrightarrow \mathbb{P}^1_{\mathbb{F}_q}$, depending on whether the former covering is separable; all further computations take place in ideal representation with respect to this covering. Then we apply a procedure for construction of the factor base and precomputation which outputs an enumerated factor base and a divisor D_1 of degree 1. After this we generate relations between the factor base elements, D_1, the input elements a, b and the further degree 0 divisor classes $c_1, \ldots c_u$ using a procedure for relation generation applied to divisor classes of the form $\alpha a + \beta b$. Finally, we perform a linear algebra computation to generate a relation between a and b.

For this, we assume that we have fixed the following five procedures (all of which can be randomized).

If the covering $x_{|\mathcal{C}} : \mathcal{C} \longrightarrow \mathbb{P}^1_{\mathbb{F}_q}$ is separable, the following divisors are represented in ideal representation with respect this covering, otherwise they are represented in ideal representation with respect to the covering $y_{|\mathcal{C}} : \mathcal{C} \longrightarrow \mathbb{P}^1_{\mathbb{F}_q}$.

a) An algorithm for computing orders of degree 0 class groups of curves $\mathcal{C}/\mathbb{F}_q$ as above.

b) An algorithm for integer factorization.

c) A procedure for *generation of the factor base and for precomputation.* We require the following: If the procedure terminates, a well-defined part of its final state (given by pointers) contains:

   - a divisor D_1 on $\mathcal{C}$ of degree 1
   - a description of an *enumerated factor base* $\mathcal{F} = \{F_1, \ldots, F_k\}$ (by this we mean that given such a result as well as a prime divisor P of $\mathcal{C}$, one can compute whether P is contained in $\mathcal{F}$, and if this is the case the number i with $P = F_i$)
   - representations of the divisor classes $a, b, c_1, \ldots, c_u$ by along D_1 reduced divisors
   - possibly some additional information for later use.

   We call all this the "result" of the computation.

d) A procedure for *relation generation.* Let us assume that we have a result of the previous procedure, as indicated above. Now let $c \in \mathrm{Cl}^0(\mathcal{C})$ be some degree 0 divisor class, represented by an along D_1 reduced divisor. Then, if the relation generation procedure terminates, it outputs vectors $(r_j)_{j=1,\ldots,k} \in (\mathbb{Z}/N\mathbb{Z})^k$ and $(s_j)_{j=1,\ldots,u} \in (\mathbb{Z}/N\mathbb{Z})^u$ defining a relation

$$\sum_j r_j [F_j] - (\sum_j r_j \deg(F_j)) \cdot [D_1] = \sum_j s_j c_j + c \,. \tag{3.4}$$

Here D_1 is the divisor of degree 1 which is computed in the previous procedure. (Usually, in the applications of the "general algorithm" D_1 is equal to D_0, which is part of the input, but in the algorithm in Section 3.3 we allow for another divisor D_1.) An important condition is that the distribution of the output is (for a fixed result of the previous procedure) independent of the *representation* of the input c. (Note that one element $c \in \mathrm{Cl}^0(\mathcal{C})$ might be represented by different bit-strings.)

e) An algorithm for *sparse linear algebra* satisfying Proposition 3.8.

The algorithm is as follows.

General index calculus algorithm

Input: An instance $(\mathcal{C}/\mathbb{F}_q, a, b)$ of the discrete logarithm problem in degree 0 class groups of curves over finite fields and some elements $c_1, c_2, \ldots, c_u$ of $\mathrm{Cl}^0(\mathcal{C})$, satisfying certain conditions and appropriately represented. Maybe some additional information.

1. Apply the algorithm to compute the group order and then the algorithm for integer factorization.
   (*Let $N = \#\,\mathrm{Cl}^0(\mathcal{C})$ and $N = \prod_{i=1}^{v} \ell_i^{e_i}$ with $e_i \in \mathbb{N}$ and pairwise distinct prime elements ℓ_i.)*
2. If the covering $x_{|\mathcal{C}} : \mathcal{C} \longrightarrow \mathbb{P}^1_{\mathbb{F}_q}$ is separable: Set up the ideal-theoretic description of divisors with respect to this covering (see subsubsection 2.5.4.2). Compute free or joint ideal representations of the input divisors. If the covering is not separable, proceed analogously with y instead of x.
   (If the input is already given in this form, this step can be omitted.)
3. Apply the procedure for generation of a factor base and for precomputation.
   (Let $\mathcal{F} = \{F_1, P_2, \ldots, F_k\} \subseteq \mathcal{C}$ be the chosen enumerated factor base, and let D_1 be the computed divisor of degree 1.)
4. 4.1. Let $t \longleftarrow \lceil \log_2(k+u) + \log_2(2\log_2(N)) \rceil + 1$
   4.2. Construct matrices $R \in (\mathbb{Z}/N\mathbb{Z})^{(2(k+u)\cdot t)\times k}$ and $S \in (\mathbb{Z}/N\mathbb{Z})^{(2(k+u)\cdot t)\times u}$ in sparse representation as well as vectors $\underline{\alpha}, \underline{\beta} \in (\mathbb{Z}/N\mathbb{Z})^{2(k+u)\cdot t}$ as follows:
      4.2.1. For $i = 1, \ldots, (k+u)\cdot t$ do
      Choose uniformly and independently randomly $\alpha_i, \beta_i \in \mathbb{Z}/N\mathbb{Z}$ and apply the algorithm to generate relations over the factor base to $\alpha_i a + \beta_i b$.

Let

$$\sum_j r_{i,j}[F_j] - r_i \cdot [D_1] = \sum_j s_{i,j} c_j + \alpha_i a + \beta_i b$$

with $r_i := \sum_j r_{i,j} \cdot \deg(F_j)$ be the relation generated.

4.2.2. For $i_1 = 1, \ldots, k$
For $i_2 = 1, \ldots, t$
Let $i \longleftarrow (i_1 - 1) \cdot t + i_2 + (k+u) \cdot t$.
Choose uniformly and independently randomly $\alpha_i, \beta_i \in \mathbb{Z}/N\mathbb{Z}$ and apply the relation generation algorithm to $[P_{i_1}] - [D_1] + \alpha_i a + \beta_i b$.
Let

$$\sum_j r_{i,j}[F_j] - r_i[D_1] = \sum_j s_j c_j + \alpha_i a + \beta_i b$$

be the relation generated.

4.2.3. For $i_1 = 1, \ldots, u$
For $i_2 = 1, \ldots, t$
Let $i \longleftarrow (i_1 - 1) \cdot t + i_2 + (2k+u) \cdot t$.
Choose uniformly and independently randomly $\alpha_i, \beta_i \in \mathbb{Z}/N\mathbb{Z}$ and apply the relation generation algorithm to $c_i + \alpha_i a + \beta_i b$.
Let

$$\sum_j r_{i,j}[F_j] - r_i[D_1] = \sum_j s_j c_j + \alpha_i a + \beta_i b$$

be the relation generated.

5. Try to compute a row vector $\underline{\gamma} \in (\mathbb{Z}/N\mathbb{Z})^{1 \times (2(k+u) \cdot t)}$ with $\underline{\gamma}(R|S) = 0$ and $[\gamma]_\ell \neq 0$ for all prime divisors ℓ of $\#\operatorname{Cl}^0(\mathcal{C})$ with the algorithm for linear algebra. If this fails, go back to Step 4.

6. If $\sum_i \gamma_i \beta_i \in (\mathbb{Z}/N\mathbb{Z})^*$, let $\xi := -\dfrac{\sum_i \gamma_i \alpha_i}{\sum_i \gamma_i \beta_i}$, otherwise go back to Step 4.

7. Compute $\operatorname{ord}(a)$, using the factorization of N.
Output the unique non-negative number $x \in \{0, \ldots, \operatorname{ord}(a) - 1\}$ with $[x]_{\operatorname{ord}(a)} = [\xi]_{\operatorname{ord}(a)}$.

Note here that the computation of $\operatorname{ord}(a)$ can be performed efficiently (in polynomial time in $\log(N)$) along the following lines:

As in the algorithm, let $N = \prod_{i=1}^{v} \ell_i^{e_i}$ with $e_i \in \mathbb{N}$ and pairwise distinct prime numbers ℓ_i. Now let $L_i := \frac{N}{\ell_i^{e_i}}$, and let $o_i := \min\{j \in 0, \ldots, e_i \mid \ell_i^j L_i \cdot a = 0\}$ for $i = 1, \ldots, v$. Then $\prod_{i=1}^{v} \ell_i^{o_i}$ is the order of a.

It is again obvious that the algorithm is correct, that is, if the algorithm terminates, x is the discrete logarithm of a with respect to b.

3.2.4 Analysis of general algorithm

We now analyze the algorithm for a fixed input and a fixed output of the factor base generation and precomputation algorithm (Step 3 in the "general algorithm").

As remarked in subsection 1.4.3, the execution of a randomized algorithm on a particular input is given by a time-uniform Markov chain. In the present analysis, we not only fix a particular input but also a particular result of Step 3. We then again obtain a time-uniform Markov chain.

We now analyze the (values of the) variables at the time Step 6 is reached. In this way we obtain random variables which we denote in the same way as the corresponding variables in the algorithm.

We have:

Proposition 3.9 *Conditionally to any result of Step 3, at the time Step 6 is reached, the random element $\sum_i \gamma_i \beta_i$ is uniformly randomly distributed in $\mathbb{Z}/N\mathbb{Z}$. In particular, the probability that $\sum_i \gamma_i \beta_i \in (\mathbb{Z}/N\mathbb{Z})^*$ is $\frac{\varphi(N)}{N}$.*

The proof relies on the following lemma.

Lemma 3.10 *Let N be a natural number, and let $\underline{\gamma} \in (\mathbb{Z}/N\mathbb{Z})^m$ with $[\underline{\gamma}]_\ell \neq \underline{0}$ for all prime divisors ℓ of N. Furthermore, let $\underline{w}$ be a uniformly distributed random element in $(\mathbb{Z}/N\mathbb{Z})^m$. Then $\sum_i \gamma_i w_i$ is uniformly distributed in $\mathbb{Z}/N\mathbb{Z}$.*

Proof. Let us first consider the case that N is a prime power. Then at least one entry of $\underline{\gamma}$ is invertible. This implies the statement. The general case follows then easily with the Chinese Remainder Theorem. □

Sketch of a proof of Proposition 3.9.

All random elements $\alpha_i a + \beta_i b$ and β_i are stochastically independent. Moreover, for each i β_i is uniformly distributed in $(\mathbb{Z}/N\mathbb{Z})^{k+1}$.

The first statement implies that all random vectors $(r_{i,j})_j$ and all elements β_i are stochastically independent, and this implies in particular that the random matrix R and the random vector $\underline{\beta}$ are stochastically independent. This in turn implies that $\underline{\gamma}$ and $\underline{\beta}$ are stochastically independent.

Now conditionally to $\underline{\gamma} = \underline{\gamma}^{(0)}$ for any fixed $\underline{\gamma}^{(0)} \in (\mathbb{Z}/N\mathbb{Z})^{k+1}$ with $[\gamma]_\ell \neq 0$ for all prime divisors ℓ of N, $\underline{\beta}$ is still uniformly distributed. With the lemma this implies that – again conditionally to $\underline{\gamma} = \underline{\gamma}^{(0)}$ –, $\sum_i \gamma_i\beta_i$ is uniformly randomly distributed in $\mathbb{Z}/N\mathbb{Z}$. But then the random element $\sum_i \gamma_i\beta_i$ is uniformly distributed too. □

Proposition 3.11 *Conditionally to any result of Step 3, the probability that the linear algebra computation fails is* $\leq \frac{1}{2}$.

This proposition relies on the following lemma. For a uniform distribution this lemma appears in [Pom87]. We note however that the proof contains quite a gap. It is also stated that the lemma below holds, but no argument is given. A generalization of the result in [Pom87] was proven by F. Heß in [Heß05]. The lemma below follows from part (i) of the proof of [Heß05, Lemma 64].

Lemma 3.12 *Let V be a vector space over any field with $n := \dim(V) < \infty$, and let $b_1, \ldots, b_n$ be a basis of V. Let $t \in \mathbb{N}$, and let $v_1, \ldots, v_{nt}$ and $v'_1, \ldots, v'_{nt}$ be identically distributed independent random vectors with values in a finite subset of V. Let for $i = 1, \ldots, n$ and $j = 1, \ldots, t$ $v_{nt+(i-1)\cdot t+j} := b_i + v'_{nt+(i-1)\cdot t+j}$.*

Let V' be the random subspace of V defined by $V' := \langle v_1, \ldots, v_{2kn}\rangle$. Then with a probability of at least $1 - \frac{n}{2^{t-1}}$ we have $V' = V$.

Proof of Proposition 3.11

With $n := k + u$ and t as in the algorithm, the lemma applies to the rows of the matrix $[(R|S)]_\ell$ for each prime divisor ℓ of N.

Let us fix a prime ℓ which divides N. By the lemma, the probability that the matrix $[(R|S)]_\ell$ does not have full column rank is $\leq \frac{k+u}{2^{t-1}}$. By the definition of t, this is $\leq \frac{1}{2\log_2(N)}$.

The number of primes dividing N is bounded by $\log_2(N)$. This implies that the probability that the linear algebra computation fails is $\leq \frac{1}{2}$. □

A brief discussion of the running time

We conclude this analysis with a very brief discussion of the running time. Of course, the actual running time of an application of the general algorithm depends crucially on the running times of the subroutines for generation of the factor base and precomputation and for relation generation.

We cannot say anything about the running time of Steps 1 and 3 at the present time. We merely want to caution the reader here that Step 3 might not in any way be of "negligible" complexity in applications. For example, Step 3 might consist of the choice of a factor base and a subsequent

construction of a so-called tree of large prime relations (cf. subsubsection 3.2.6.2 below).

Step 2 can be performed in a time which is polynomially bounded in d, the degree of the plane model, and $\log(q)$ by the results of subsubsections 2.5.4.2 and 2.5.4.6 as well as subsection 2.5.5. If one allows only plane models of degree $\mathcal{O}(g)$, this is of course polynomially bounded in g and $\log(q)$.

In Step 4, clearly, we need $2(k+u)$ calls to the relation generation procedure. Again nothing can be said about the actual running time.

Step 5 can be performed with $\tilde{\mathcal{O}}((k+u)\cdot(k+u+\omega)\cdot\log(N))$ bit operations according to Proposition 3.8 and the definition of t, where ω is the number of non-zero elements of the matrix. We note again that the success probability is $\geq \frac{1}{2}$ conditionally to any outcome of Step 3 by Proposition 3.11.

The success probability of Step 6 is $\frac{\varphi(N)}{N}$ by Proposition 3.9. Note that $\frac{\varphi(N)}{N} \in \Omega(\frac{1}{\log\log(N)})$ (cf. [RS62, Formula 3.41]).

This means that conditionally to any outcome of Step 3, with a probability of $\Omega(\frac{1}{\log\log(N)})$, the algorithm terminates without going back to Step 4. Or with other words: Conditionally to any outcome of Step 3, the expected number of iterations is in $\mathcal{O}(\log\log(N))$.

3.2.5 Some historical remarks

We make some historical remarks on the index calculus method and the origin of the phrase "index calculus", and we put the general algorithm given above into historical perspective.

As mentioned above, a classical term for "discrete logarithm" in $\mathbb{F}_p^*$ with respect to a generating element is "index". Various authors have produced tables of indices (analogous to usual logarithm tables). For example, C.G. Jacobi gave such a table for all prime powers less than 1000 in his "Canon Aritheticus" ([Jac39]). In his book "Théorie des nombres" ([Kra22]) from 1922 M. Kraitchik showed how one can compute indices of residue classes of small primes via relation generation and linear algebra without having to compute the indices of all elements of $\mathbb{F}_p^*$. He stated that with this method he has computed all indices of primes up to 100 in all prime fields $\mathbb{F}_p^*$ with $p \leq 10000$. A book of tables was also produced by A. Western and J. Miller in 1968 ([WM68]) with a relation-based method. In [Mil75] J. Miller attributed the method to A. Western, but as we have just remarked it was in fact already used by Kraitchik. To our knowledge, the first time that a relation generation and linear algebra based approach to index computation was called "index calculus" was in A. Odlyzko's work "Discrete logarithms in finite fields and their cryptographic significance" ([Odl84]) from 1984.

Around 1978-79 the method was independently studied L. Adleman ([Adl79]); it was also mentioned by J. Pollard [Pol78]).

An outline for an analysis of an index calculus algorithm leading to an expected subexponential running time in the case of the multiplicative groups finite prime fields was given by Adleman in [Adl79]. A rigorous result for computations in finite fields of characteristic 2 was given by M. Hellman and J. Reyneri in [HR83]. After that a rigorous result for both finite fields in characteristic 2 and finite prime fields was given by C. Pomerance in [Pom87]. Pomerance obtained an expected running time of $L_q[\frac{1}{2}, \sqrt{2}+o(1)]$, where q is the size of the finite field and $L[\alpha, c]$ is the subexponentiality function with parameters α and c which was already defined in the introduction:

$$L_N[\alpha, c] := e^{c\cdot(\log(N))^{\alpha}\cdot(\log\log(N))^{1-\alpha}}$$

This is still the best proven running time for discrete logarithms in prime fields or finite fields of a fixed characteristic.

A. Enge and P. Gaudry gave in [EG02] a general framework for subexponential discrete logarithm algorithms under the assumption that the group order is known. Their setting is as follows: They consider groups G such that there exists a free abelian monoid M and a computable surjective morphism $p : M \longrightarrow G$ with a computable section or "lifting" $s : G \longrightarrow M$ and a degree-function on M. To define the factor base, they fix a "smoothness bound" S and define the factor base as the set of basis elements of M whose degree is $\leq S$.

As examples of this general setting they discussed the discrete logarithm problems in finite prime fields, in finite fields of a fixed characteristic, in class groups of number fields and in degree 0 class groups of hyperelliptic curves in imaginary quadratic representation. They showed in particular that in this framework discrete logarithms in degree 0 class groups of hyperelliptic curves in imaginary quadratic representation can be computed in an expected running time of $L_{q^g}[\frac{1}{2}, \sqrt{2} + o(1)]$, provided that one restricts oneself to instances with $\frac{g}{\log(q)} \longrightarrow \infty$ and the group is cyclic or a basis of the group is known. Here g is the genus and $\mathbb{F}_q$ the ground field of a curve, and by a basis of the group a set of elements is meant such that every element has a unique coordinate vector with respect to the set.[4]

Even though the work by Enge and Gaudry is related to the work by Pomerance, there is a noticeable difference: Pomerance's work is based on the traditional way to compute indices, going back to Kraitchik as well as Western and Miller: Let $g \in \mathbb{F}_p^*$ be a generating element, and let $\mathcal{F}$ be a

[4]In fact, the analysis in [EG02] shows that one only needs that $g \geq \vartheta \cdot \log(q)$ for some constant $\vartheta > 0$. Explicitly, in Example 3, the condition $k \geq \vartheta \log p$ implies that $p \leq e^{(\frac{\log N}{\vartheta})^{1/2}} \in L_N(o(1))$, and then one obtains a total running time of $L_N(\sqrt{2} + o(1))$. An analogous observation holds for Example 5.

factor base. Let us now assume that g^i splits over the factor base. Then one obtains a relation between i on the one hand and the (unknown) indices on the other hand. Using Lemma 3.12 Pomerance shows that one can with high probability obtain a linear system with a unique solution; the entries of the solution are then the indices. In a second step one then relates the input of the discrete logarithm problem to the factor base.

In contrast, Enge and Gaudry use a direct approach to eliminate the factor base elements. Our general index calculus algorithm presented above is closely related to the general algorithm in [EG02]; we only depart from the algorithm in [EG02] in the following four points.

- We make use of an arbitrary generating system whereas for non-cyclic groups in [EG02] a basis of the group is required. In order to deal with the arbitrary generating system, our algorithm is slightly different from the one in [EG02].

- We do not use a free abelian monoid M in the way Enge and Gaudry do. Of course, we have the canonical surjective map $\mathrm{Div}(\mathcal{C}) \longrightarrow \mathrm{Cl}(\mathcal{C})$, but only if the divisor D_1 is a prime P of degree 1, we have the map $\mathrm{Div}(\mathcal{C} - \{P\}) \longrightarrow \mathrm{Cl}^0(\mathcal{C}), D \mapsto [D] - \deg(D) \cdot [P]$ which naturally generalizes the setting for hyperelliptic curves in imaginary quadratic representation in [EG02]. In particular we do not fix a lifting from divisor classes to divisors in the way it is done in [EG02].

- We drop the condition of the factor base being defined by a smoothness bound.

- We not only choose a factor base but also allow of a "precomputation" afterwards. This will become important when we describe the double large prime variation algorithms for curves of small genus. The fact that the analysis in [EG02] still holds true after a precomputation was already pointed out in [GTTD07] (concerning the computation of a graph of large prime relations).

3.2.6 Large prime variation and double large prime variation

"Large prime variation" and "double large prime variation" are add-ons to the index calculus method which can be used to improve the running time. We describe the ideas of these approaches and then show how slightly modified approaches can be integrated into the general algorithm given above.

3.2.6.1 Large prime variation

We give a general description of "large prime variation" in the case of groups with arbitrary (not necessarily cyclic) group structure.

As above, let $c_1, \ldots, c_u$ be a generating set of $\mathrm{Cl}^0(\mathcal{C})$. Now additionally to the factor base $\mathcal{F}$ one fixes a so-called *set of large primes* $\mathcal{L}$ consisting of prime divisors of $\mathcal{C}$ which are not contained in $\mathcal{F}$.

Now one not only tries to find relations of the form

$$\sum_j r_j[F_j] - r[D_1] = s_1 c_1 + \cdots + s_u c_u + \alpha a + \beta b \tag{3.5}$$

but in fact relations which involve up to one element from $\mathcal{L}$, that is, relations as (3.5) and also relations of the form

$$\sum_j r_j[F_j] + r_P[P] - r[D_1] = s_1 c_1 + \cdots + s_u c_u + \alpha a + \beta b \tag{3.6}$$

with $P \in \mathcal{L}$ and $r_P \neq 0$. (How these relations are generated depends on the application and is not discussed here.) The original idea of large prime variation is now to proceed as follows: Whenever one encounters two such relations with the same large prime, one produces a relation between factor base elements, generating elements and a, b by eliminating the large prime.

A different method which fits well into our general algorithm above is as follows: One proceeds as above, but if one encounters a relation without a large prime, one discards it. Now each relation generated links a particular large prime to the factor base elements and the generating set. If one has linked enough large primes to the factor base and the generating set, one stops the process; let $\mathcal{L}' \subseteq \mathcal{L}$ be the set of large primes linked to the factor base in this way. Then one generates relations as in Step 4 of the general algorithm, but with $\mathcal{F} \cup \mathcal{L}'$ in place of $\mathcal{F}$. Whenever one has obtained a relation between factor base elements, large primes, the generating system and the input elements a, b one can eliminate the large primes in these relations, thus obtaining a relation between factor base elements and the generating elements. In this way one proceeds as in the general algorithm above.

Note here that this approach can be described as follows: The first part of the method can be seen as a "precomputation". After this, we again have a procedure to obtain relations over the factor base, just a different one than the previous one. Therefore, the approach still fits our general algorithm, and in particular Propositions 3.9 and 3.11 remain valid.

Finally, we would like to caution the reader not to take the terminology "large primes" too literally: The idea of large prime variation was first used in the context of integer factorization, and there the large primes are really

larger than the factor base elements. However, in our applications below, both the factor base elements and the large primes have degree 1.

3.2.6.2 Double large prime variation

"Double large prime variation" can be seen as a further development of large prime variation.

We fix a set of "large primes" $\mathcal{L}$ as above. Now one uses relations with up to two large primes, that is relations of the form (3.5), (3.6) as well as

$$\sum_j r_j[F_j] + r_P[P] + r_Q[Q] - r[D_1] = s_1 c_1 + \cdots s_u c_u + \alpha a + \beta b \qquad (3.7)$$

with $P, Q \in \mathcal{L}$, $P \neq Q$ and $r_P, r_Q \neq 0$. Let us fix the following terminology, following for example [GTTD07].

Terminology 3.13 A relation of the form (3.5) is called a *Full relation*, a relation of the form (3.6) is called an *FP relation*, and a relation of the form (3.7) is called a *PP relation*.

The general idea to use all these relations is now as follows: One constructs a so called *graph of large prime relations*, which is a labeled graph on the set $\mathcal{L} \dot\cup \{*\}$: A PP relation involving large primes P and Q leads to an edge from P to Q, labeled with the relation. A FP relation involving a large prime P leads to an edge from P to $*$. Now cycles in the graph allow that large primes are canceled. A cycle involving $*$ always leads to a Full relation, and other cycles lead to a Full or an FP relation.

There are several variants of this idea. Below we discuss the variants which are relevant in the context of the present work (see also Remark 3.15 below). In all these variants, we represent the graph of large prime relations by *adjacency lists*.

Construction of an acyclic graph In this variant, first of all, Full relations are saved in the relation matrix. If one has an FP or a PP relation, one proceeds with a case distinction. One inserts a corresponding edge provided that it does not lead to a cycle. If however one would obtain a cycle in doing so, one does not insert the corresponding edge – the graph thus stays acyclic. Additionally, via the cycle one then tries to cancel large primes so that one obtains a Full relation or an FP relation. With these "combined" relations one proceeds as already indicated. Note here that one always obtains a Full relation if the cycle contains $*$ but otherwise one might obtain a Full or an FP relation. One terminates if one has obtained enough Full relations.

This variant is referred to as "full algorithm" in [GTTD07].

Note however that this approach does not fit into our general algorithm given above. Indeed, for an asymptotic analysis, generally speaking, it seems to be preferable to first construct a suitable *tree* with root $*$ and then to use this tree to obtain Full relations as follows:

Let us assume that a tree T has been constructed. Now one again generates relations. If one has a relation which only involves elements of the factor base or vertices from T, one can eliminate the vertices from T with the help of the tree to obtain a relation over the factor base.

Note that this approach fits into our general algorithm. It remains however the question how to construct such a tree, which we call a *tree of large prime relations*. We are aware of the following three possibilities.

Direct computation of a tree Here during the construction of the graph of large prime relations, one discards all relations which would lead to a component which is not connected to $*$, and one discards all relations which would lead to cycles. This means that at all times during the construction, the graph of large prime relations is already a tree.

An algorithm based on such a direct computation of a tree is given in [GTTD07]. It is called *simplified algorithm*, and it is used to obtain a proven result on the complexity of the computation of discrete logarithms in cyclic degree 0 class groups of hyperelliptic curves in imaginary quadratic representation.

Stage-wise computation of a tree In our applications, we want to control the depth of the tree, because we want to keep the relation matrix sparse. One possibility to do so is the following variant of the direct computation: One proceeds in "stages around $*$". In the first stage, one considers only relations involving elements of the factor base and one large prime, that is, FP relations. This leads to a tree of depth one. After having found suitably many FP relations, one moves to the second stage. Here one considers only relations which contain one elements of the factor base, vertices of the first stage and one large prime. This leads to a tree of depth ≤ 2. Again, if the tree has enough vertices, one moves to the next stage. One continues in this manner, such that after the k^{th} step the depth of the tree is $\leq k$ and the tree has a predefined number of vertices.

Section 3.3, where we extend the results of [GTTD07] to arbitrary curves of a fixed genus, is based on this approach.

Computation of a tree from a graph Here one first constructs a suitably large graph. Then inside this graph one constructs a so-called *shortest path tree* inside this graph. This approach is for example taken in the work

[Die06] by the author.

Remark 3.14 Above we suggested to construct a tree labeled with relations. Note that given an edge of the tree one can easily obtain a relation between the edge, factor base elements and the generating system. However, if one always immediately generates these relations, one does not need the tree at all; rather one just needs a list of relations involving one large prime each. A variant of the stage-wise computation following this approach was pursued by K. Nagao in [Nag04] (see also [AT06, subsubsection 3.2 c)] where a slightly different method is called "concentric circles method").

Remark 3.15 Double large prime variation was originally suggested by A.K. Lenstra and M.S. Manasse as a practical improvement for factorization algorithms in [LM91] and [LM94]. (As noted in [LM94] the idea might however have been older.) Since then various works on practical aspects of double large prime variation, including for computations of discrete logarithms, appeared. In practical computations one faces severe storage problems if one tries to store all generated FP and PP relations in the RAM of a computer. Because of this, one usually proceeds as follows: First one stores a list of FP and PP relations and stores them on a hard drive. Then in several "filtering steps" (and using Hash-tables) one constructs a graph. For example, in [LM94] it was suggested to construct directly a breadth-first tree in the graph of large prime relations by reading the list of FP and PP relations over and over.

It was then realized by N. Thériault that for the solution of the discrete logarithm problem in hyperelliptic curves in imaginary quadratic representation of a *fixed genus*, one can obtain a decrease in the running time which is *superpolynomial* in the cardinality of the ground field (see [Thé03]). (Note however that the main results in [Thé03] are not proven but only argued heuristically.) Later, it was proven by P. Gaudry, E. Thomé, N. Thériault and the author that one can obtain a further superpolynomial decrease in the running time by using a double large prime variation, provided that the degree 0 class group of the curve is cyclic.

It is the superpolynomial decrease in the running time for discrete logarithm computation in degree 0 class groups of a fixed genus which is our motivation for investigation into this method. As we study in this work algorithms from an asymptotic point of view in formalized models, practical considerations as mentioned above are not relevant for the present work.

3.3 Index calculus with double large prime variation for curves of fixed genus

3.3.1 Introduction and results

In this section we present an index calculus algorithm with double large prime variation for curves of a fixed genus g. The input of the algorithm consists of a tuple $(\mathcal{C}/\mathbb{F}_q, a, b)$, where $\mathcal{C}$ is a curve of genus g over $\mathbb{F}_q$, $a, b \in \mathrm{Cl}^0(\mathcal{C})$ and $b \in \langle a \rangle$. The representation of the input follows Chapter 2: The curve is represented by a plane model, a divisor D_0 of degree 1 is fixed and a, b are represented by along D_0 reduced divisors which in turn is given by any of the possibilities described in Chapter 2.

Additionally, we fix some number $d_{\max}$ and demand that the curve $\mathcal{C}$ is represented by a plane model of degree $\leq d_{\max}$. Note that by Proposition 2.6 such a number $d_{\max}$ exists. We also fix an upper bound on the height of the divisor D_0.

On the basis of our algorithm, we obtain the following result.

Theorem 1 *Let some natural number $g \geq 2$ be fixed. Then the discrete logarithm problem in the degree 0 class groups of curves of genus g over finite fields can with a randomized algorithm be solved in an expected time of*

$$\tilde{\mathcal{O}}(q^{2-\frac{2}{g}}) ,$$

where $\mathbb{F}_q$ is the ground field of the curve.

We remark that the algorithm has storage requirements of $\tilde{\mathcal{O}}(q^{1-\frac{1}{g}+\frac{1}{g^2}})$. More concretely, although the algorithm is randomized, there exists a function in $\tilde{\mathcal{O}}(q^{1-\frac{1}{g}+\frac{1}{g^2}})$ such that the storage requirements are bounded by this function for every run.

For hyperelliptic curves in imaginary quadratic representation such that the degree 0 class group is cyclic or a basis of the group is known, the result was already obtained in [GTTD07]. As already mentioned in subsubsection 3.2.6.2 the result in [GTTD07] is proven with an algorithm which follows the strategy of "direct computation of a tree". With minor modifications, this algorithm also applies to arbitrary curves, and *on a heuristic basis*, the "simplified algorithm" leads to the result in Theorem 1.

On the proof of Theorem 1

We now give an outline of the proof of Theorem 1.

First we give an index calculus algorithm with double large prime variation which gives rise to the following proposition.

Proposition 3.16 *Let us fix some $g \geq 2$. Then there exists a randomized algorithm such that the following holds: Upon input of a curve $\mathcal{C}/\mathbb{F}_q$, elements $a, b \in \mathrm{Cl}^0(\mathcal{C})$ with $b \in \langle a \rangle$ and a generating system $c_1, \ldots, c_u$ whose size is polynomially bounded in $\log(q)$, the algorithm outputs the discrete logarithm of b to with respect to a. The expected running time of the algorithm is in $\tilde{\mathcal{O}}(q^{2-\frac{2}{g}})$. Moreover, the algorithm has storage requirements of $\tilde{\mathcal{O}}(q^{1-\frac{1}{g}+\frac{1}{g^2}})$. More precisely, the algorithm uses only the first $\tilde{\mathcal{O}}(q^{1-\frac{1}{g}+\frac{1}{g^2}})$ registers for* LOAD *and* STORE *operations, and each register always contains elements whose bit-length is polynomially bounded in $\log(q)$.*

The algorithm is given in the next subsection and the analysis in subsection 3.3.3.

Then we show the that there is an efficient algorithm which outputs a small system of divisor classes of degree 0 which generates the degree 0 class group with high probability (see subsection 3.3.5):

Proposition 3.17 *Let us fix some $g \geq 1$. Then there exists a randomized algorithm such that the following holds: Upon input of a curve $\mathcal{C}/\mathbb{F}_q$ (as always represented by a plane model of degree $\leq d_{\max}$) and a divisor D_0 of degree 1 and height polynomially bounded in $\log(q)$, the algorithm computes a system of random elements $c_1, \ldots, c_u$ of $\mathrm{Cl}^0(\mathcal{C})$, represented by along D_0 reduced divisors, where $u := e\ell$ with $e := \lceil \log_2(\#\,\mathrm{Cl}^0(\mathcal{C})) \rceil$ and $\ell := \lceil \log_2(e) + 1 \rceil$. Moreover, expected running time is polynomially bounded in d and $\log(q)$, and with a probability $\geq \frac{1}{2}$, the system $c_1, \ldots, c_u$ is a generating system of $\mathrm{Cl}^0(\mathcal{C})$.*

A problem is however that we do not know how one can in a sufficiently efficient way certify that the output is indeed a generating system. As a work-around, we proceed as follows: We assume that we have a generating system and apply the index calculus algorithm. We stop the computation if it has not terminated within a predefined time.

An obvious consequence of Proposition 3.16 is:

Proposition 3.18 *Let us fix some $g \geq 2$. Then there exists a randomized algorithm such that the following holds: Upon input of a curve $\mathcal{C}/\mathbb{F}_q$, elements $a, b \in \mathrm{Cl}^0(\mathcal{C})$ with $b \in \langle a \rangle$ and a system $c_1, \ldots, c_u$ of elements in $\mathrm{Cl}^0(\mathcal{C})$ with u polynomially bounded in $\log(q)$, the algorithm either fails or outputs the discrete logarithm of b with respect to a. The running time of the algorithm is in $\tilde{\mathcal{O}}(q^{2-\frac{2}{g}})$, and if $c_1, \ldots, c_u$ is a generating system, the probability of failure is $\leq \frac{1}{2}$. The algorithm has storage requirements of $\tilde{\mathcal{O}}(q^{1-\frac{1}{g}+\frac{1}{g^2}})$.*

Indeed, let us fix a bit-RAM Π satisfying the claim in Proposition 3.16, and let $\delta > 0$ such that the bit-RAM terminates in an expected time of $q^{2-\frac{2}{g}} \cdot \log(q)^\delta$ if applied to an instance as in Proposition 3.16. We apply this bit-RAM with the input of Proposition 3.18 and terminate the execution if a time of $\lceil 2 \cdot q^{2-\frac{2}{g}} \cdot \log(q)^\delta \rceil$ is exceeded – provided the algorithm has not terminated at this point in time. (For this we "mix" the Turing machines for the commands with "control units" which measure the running times and terminate if necessary, and we mix Π with similar "control units".) By Markov's bound we have: If $c_1, \ldots, c_u$ is a generating system, with a probability of $\geq \frac{1}{2}$, the algorithm outputs the discrete logarithm of b with respect to a.

Note here that just from the statement on the running time we cannot conclude that we indeed the running time is in $\tilde{\mathcal{O}}(q^{2-\frac{2}{g}})$. It has to be ruled out that very expensive LOAD and STORE operations are performed, commands whose running time is not in $\tilde{\mathcal{O}}(q^{2-\frac{2}{g}})$. However, as stated in 3.16 every LOAD and STORE command is executed in a time which is polynomially bounded in $\log(q)$. □

The *proof of Theorem 1* is now easy:

So let $g \geq 2$ be fixed. Now given an instance $(\mathcal{C}, a, b)$ with $g(\mathcal{C}) = g$ as described above, we proceed as follows: First we apply an algorithm for which Proposition 3.17 holds; let the result be $c_1, \ldots, c_u$. Then we apply an algorithm for which Proposition 3.18 holds to $\mathcal{C}, a, b$ and the system $c_1, \ldots, c_u$. The expected running time is then in $\mathcal{O}(q^{2-\frac{2}{g}})$, and moreover, the probability of failure is $\leq \frac{3}{4}$. This implies Theorem 1. □

3.3.2 The index calculus algorithm

Let $g \geq 2$ be fixed. We now describe the index calculus algorithm which leads to Proposition 3.16. As already stated, the algorithm uses double large prime variation. We do so by computing a tree of large prime relations. The main challenge resides in controlling the growth of the tree of large prime relations as well as its depth, that is, the maximal distance of any vertex to the root. This task is more difficult than for hyperelliptic curves in imaginary quadratic representation, where one has a concrete description of the effective divisors which are reduced along the point at infinity, and one knows that the growth process is very regular.

In the algorithm below we employ the strategy of "stage-wise computation of a tree" in subsubsection 3.2.6.2. The reason why we adopt this strategy is that it is then immediate that the depth of the tree (not only the expected value of the depth) lies in $\mathcal{O}(\log(q))$.

The algorithm is a realization of the "general algorithm" of the previous

section. It therefore relies on subroutines for a) computation of the group order, b) factorization, c) generation of a factor base and precomputation as well as d) relation generation, e) linear algebra. Algorithms for factorization and for linear algebra where already discussed above.

We now outline procedures for a), c) and d) which lead to Proposition 3.16.

a) Computation of the group order

The L-polynomial of a curve over $\mathbb{F}_q$ given by a plane model of bounded degree can be computed (with a deterministic algorithm) in a time which is polynomially bounded in $\log(q)$. (This follows from [Pil91, Theorem H] which in turn relies on Pila's extension of the point counting algorithm by Schoof ([Sch85]) to abelian varieties ([Pil90]).) This means in particular that the order of the degree 0 class group can be computed in polynomial time in $\log(q)$.

In the following description as well in the analysis of subroutine c) and d) in the next subsection, we implicitly assume that certain lower bounds on q are satisfied. In the analysis in the next subsection, we then state further lower bounds which have to be satisfied in order that the analysis holds. If these bounds are not satisfied, the algorithm might fail (it might stop without a result or not terminate in finite expected time). Note that the number of isomorphism classes of curves for which the algorithm fails is finite. As usual, one obtains an algorithm which always terminates in a finite expected time by running the algorithm "in parallel" with a brute-force computation.

c) Construction of the factor base and the graph of large prime relations

Let an instance as described in Proposition 3.16 be given: Let $\mathcal{C}$ be a curve of genus g over $\mathbb{F}_q$, let $a, b \in \mathrm{Cl}^0(\mathcal{C})$, and let $c_1, \ldots, c_u$ be a generating system of $\mathrm{Cl}^0(\mathcal{C})$. Moreover, let $N := \#\,\mathrm{Cl}^0(\mathcal{C})$. As usual, let $\mathcal{C}$ be given by a plane model which in turn is given by a homogeneous polynomial $F(X, Y, Z) \in \mathbb{F}_q[X, Y, Z]$.

In the procedure, a point $P_0 \in \mathcal{C}(\mathbb{F}_q)$ is fixed and divisor classes are represented by along P_0 reduced divisors. The procedure uses a factor base $\mathcal{F} = \{F_1, F_2, \ldots\} \subseteq \mathcal{C}(\mathbb{F}_q) - \{P_0\}$ of size $\lceil q^{1-\frac{1}{g}} \rceil$, and the set of large primes is $\mathcal{L} := \mathcal{C}(\mathbb{F}_q) - (\mathcal{F} \cup \{P_0\})$.

Construction of the factor base We first enumerate the points in $\mathcal{C}(\mathbb{F}_q)$. Then we fix a point P_0 (if such a point exists) and an enumerated factor

base $\mathcal{F} \subseteq \mathcal{C}(\mathbb{F}_q) - \{P_0\}$ of size $\lceil q^{1-\frac{1}{g}} \rceil$ (if this is possible).

The tree of large prime relations We construct a tree of large prime relations whose vertex set is contained in $\mathcal{L} \dot\cup \{*\}$.

In the following, we always represent divisor classes by along P_0 reduced divisors in free ideal representation. We therefore change the representation of the divisor classes $a, b, c_1, \ldots, c_u$ from along D_0 to along P_0 reduced divisors.

Then we proceed as follows: We repeatedly choose uniformly randomly $s_1, \ldots, s_u \in \mathbb{Z}/N\mathbb{Z}$ and compute the along P_0 reduced effective divisor D in free representation with

$$[D] - \deg(D) \cdot [P_0] = \sum_{j=1}^{u} s_j \, c_j \ , \tag{3.8}$$

where $P_0 \in \mathcal{C}(\mathbb{F}_q)$ is the point computed above. (As stated, P_0 takes the role of D_1 in the "general algorithm".)[5,6] Note that by Proposition 2.117 and the representation of the curve by a plane model of bounded degree, the computation of D is possible in an expected time which is polynomially bounded in $\log(q)$.

As already mentioned, we construct the tree in *stages*, and during each stage we only attach edges to the tree which are connected to vertices constructed *in the previous stage*. In Stage 1, we attach $\lceil q^{1-1/g} \rceil$ edges coming from FP relations to the root $*$. Thereafter, we terminate Stage s and start Stage $s+1$ whenever the tree has $2^{s-1} \cdot \lceil q^{1-1/g} \rceil$ edges.

The construction of the tree is abandoned if a predefined number of edges $N_{\max}$ is reached. We could for example set $N_{\max} := \lceil q/4 \rceil$. We will however argue in the analysis of the algorithm in the next subsection that $N_{\max} := \lceil q^{1-1/g+1/g^2} \rceil$ suffices. This smaller value of $N_{\max}$ only lowers the time for the construction of the tree by a constant factor but decreases the storage requirements substantially. This is analogous to the situation in [GTTD07].

Let us fix this notation.

Notation 3.19 The set of vertices of a tree T is also denoted by T.

Altogether, we have the following procedure for construction of a suitable tree of large prime relations. In the procedure we construct a labeled tree

[5] Note here that by the fixed representation of divisor classes, to compute the divisor D is the same as computing the sum $\sum_{j=1}^{u} s_j \, c_j$ in the degree 0 class group.

[6] As usual, we apply the command RAND to choose the s_i. (Such that these elements are chosen independently of each other.)

called T. The edges of the tree are labeled with the corresponding relations; the vertices are labeled too, namely with the stage at which they inserted into the tree. We denote the subtree of T which has been constructed until (including) stage s by T_s. With other words: A vertex of T occurs in T_s if and only if its label is $\leq s$.

Procedure: Construction of the tree of large prime relations

Construct a tree T with vertex set contained in $\mathcal{L} \dot\cup \{*\}$ as follows:
Let T consist only of the root $*$.
Let $N_{\max} \longleftarrow \lceil q^{1-1/g+1/g^2} \rceil$
Let $s \longleftarrow 1$.
Repeat
  Repeat
    Choose $s_1, \ldots, s_u \in \mathbb{Z}/N\mathbb{Z}$ uniformly and independently at random.
    Compute the along P_0 reduced divisor D in free representation with $[D] - \deg(D) \cdot [P_0] = \sum_j s_j c_j$.
    If D splits as $D = \sum_j r_j F_j + c_P P + c_Q Q$ where $c_P > 0, c_Q > 0$
      and $P \in \mathcal{F} \cup T_{s-1}$ and $Q \in \mathcal{L} - (\mathcal{F} \cup T)$,
      if $P \in \mathcal{F}$ (i.e. if we have an FP relation),
        insert Q and an edge from $*$ to Q into T
      if $P \in T_{s-1}$ (i.e. if we have a PP relation),
        insert Q and an edge from P to Q into T.
      In both cases label Q with s and the edge with $(r_j)_j$ (in sparse representation).
  Until T contains $\min\{2^{s-1} \cdot \lceil q^{1-1/g} \rceil, N_{\max}\}$ edges.
  If the number of edges equals $N_{\max}$, STOP.
  Let $s \longleftarrow s + 1$.

This construction of the tree guarantees that the depth of the tree is always in $\mathcal{O}(\log(q))$ (see also inequality (3.15) in the next subsection). The main difficulty of the analysis of the procedure resides in proving that a tree of sufficient size can be constructed in an expected time of $\tilde{\mathcal{O}}(q^{2-2/g})$. This is verified in the next subsection.

d) Relation generation

We come to the relation generation procedure. Recall that we have the following specification of the relation generation procedure: If applied to a result of the procedure for generation of the factor base and precomputation as well as an element $g \in \mathrm{Cl}^0(\mathcal{C})$, it should output vectors $(r_j)_j$ and $(s_j)_j$

over $\mathbb{Z}/N\mathbb{Z}$ defining a relation

$$\sum_j r_j[F_j] - (\sum_j r_j \deg(F_j)) \cdot [P_0] = \sum_j s_j c_j + c\,. \tag{3.9}$$

So let us assume that the factor base has been chosen and a tree of large prime relations T as above has been constructed, and let $g \in \mathrm{Cl}^0(\mathcal{C})$. Then we repeatedly choose $s_1, \ldots, s_u$ uniformly at random and compute the along P_0 reduced divisor D in free representation such that

$$[D] - \deg(D) \cdot [P_0] = \sum_{j=1}^{u} s_i\, c_j + c\,, \tag{3.10}$$

until D splits over $\mathcal{F} \cup T$. If this is the case, we use the tree to substitute the large primes involved. Like this we obtain a Full relation. We output this relation.

3.3.3 Analysis of the index calculus algorithm

We now show that with the subroutines described above, one can with the "general algorithm" given in subsection 3.2.3 compute a solution to the discrete logarithm problem in an expected time of $\tilde{\mathcal{O}}(q^{2-2/g})$ (as always for fixed genus $g \geq 2$, and moreover if the size of the generating system is polynomially bounded in $\log(q)$).

Steps 1 and 2 – computation of the group order and factorization, setting up of the ideal representation

We have already argued that one can perform these steps in an expected time of $L_q[\frac{1}{2}, 1 + o(1)]$. Therefore, these steps are not time critical.

Step 3 – construction of the factor base and the tree of large prime relations

Construction of the factor base One can iterate over all points of $\mathcal{C}(\mathbb{F}_q)$ in an expected time of $\tilde{\mathcal{O}}(q)$ as follows: Let us assume wlog. that the covering $x : \mathcal{C} \longrightarrow \mathbb{P}^1_{\mathbb{F}_q}$ is separable. (Thus in Step 2 of the "general algorithm", this covering is considered.) Then using the ideal representation, we proceed as follows: We iterate over all points of $\mathbb{P}^1(\mathbb{F}_q)$, and for each such point P we consider its preimage $x^{-1}(P)$ in $\mathcal{C}$ (a trivial operation in joint ideal representation). We compute the free representation of $x^{-1}(P)$, which is possible in an expected time which is polynomially bounded in $\log(q)$ by Proposition 2.86.

In particular, we can find a point P_0 (if such a point exists) and an appropriate factor base in an expected time of $\tilde{\mathcal{O}}(q)$.

Construction of the tree of large prime relations We come to the analysis of the growth of the tree of large prime relations. The analysis relies crucially on the following proposition.

Proposition 3.20 *For curves of fixed genus g over finite fields $\mathbb{F}_q$, the number of special effective divisors of degree g is in $\mathcal{O}(q^{g-1})$.*

Recall that an effective divisor D is called *special* if the linear system $|K-D|$ is non-empty, where K is a canonical divisor. Note that by the Riemann-Roch theorem, an effective divisor of degree g is non-special if and only if it is the only if the linear system $|D|$ merely contains D itself.

Note that map $D \mapsto [D]$ gives an injection from the set of non-special divisors of degree g into the set of divisor classes of degree g, and therefore, the map $D \mapsto [D - P_0]$ gives an injection from the set of non-special divisors of degree g into the degree 0 divisor class group. Then we can apply the bijection between the degree 0 class group and the set of along P_0 reduced divisors.

Explicitly, let D be a non-special effective divisor of degree g, and let D' be the unique effective divisor of minimal degree with $D' + (\deg(D) - \deg(D')) \cdot P_0 = D$. Then D' is reduced along P_0, and it is the along P_0 reduced divisor which represents the class $[D]$.

We assume that Proposition 3.20 is well known to many experts in curves and function fields. For the lack of a suitable reference we give a proof in the next subsection. Note that a straightforward application of the Hasse-Weil bound merely gives that the number in question is in $O(q^{g-1/2})$.

This proposition makes it possible to discard all special divisors in the analysis of the construction of the tree of large prime relations.

Let $C > 0$ be such that for all curves of genus g over any finite fields $\mathbb{F}_q$ the number of special divisors of degree g is $\leq C \cdot q^{g-1}$.

As in the previous subsection, let $N_{\max} := \lceil q^{1-1/g+1/g^2} \rceil$ be the number of edges (that is, the number of vertices different from $*$) at which the construction of the tree is stopped.

The conditions

$$N_{\max} + \#\mathcal{F} \leq q/4 \qquad \#(\mathcal{C}(\mathbb{F}_q) - \{P_0\}) \in [\max\{q^{1-\frac{1}{g}}, q/2\}, 2q]$$
$$\#\operatorname{Cl}^0(\mathcal{C}) \leq 2q^g \qquad q \geq (4 \cdot g! \cdot C)^g$$

hold for $q \gg 0$, that is, for q large enough; we assume that they are satisfied.

Note that by our assumption that $c_1, \ldots, c_u$ generate $\operatorname{Cl}^0(\mathcal{C})$, if $s_1, \ldots, s_u$ are uniformly distributed random elements from $\mathbb{Z}/N\mathbb{Z}$, $\sum_i s_i c_i$ is uniformly distributed in $\operatorname{Cl}^0(\mathcal{C})$. This means that the divisor D in (3.8) is uniformly distributed in the set of all effective divisors which are reduced along P_0.

By our assumptions on q, we always have

$$\#(\mathcal{C}(\mathbb{F}_q) - (T \cup \mathcal{F} \cup \{P_0\}) \geq q/2 - q/4 = q/4 \,. \tag{3.11}$$

Let $\mathrm{Div}^g(\mathcal{C})$ be the set of effective divisors of degree g on $\mathcal{C}$, and let $\mathrm{Div}^{g,ns}(\mathcal{C})$ (resp. $\mathrm{Div}^{g,\mathrm{s}}(\mathcal{C})$) be the subset of non-special (resp. special) effective divisors of degree g.

Let us first assume that we are still in Stage 1, that is, only relations with one large prime (not yet in the tree) are considered.

Let us thus assume we are given the tree T with $< \lceil q^{1-1/g} \rceil$ edges. We want to bound the expected number of relations (3.8) needed until a new edge is inserted into the tree.

Let

$$\mathcal{D} := \big\{ P_1 + \cdots + P_g \in \mathrm{Div}^g(\mathcal{C}) \mid \forall i = 1, \ldots, g-1 : P_i \in \mathcal{F}, \\ P_g \in \mathcal{C}(\mathbb{F}_q) - (T \cup \mathcal{F} \cup \{P_0\}) \big\} \,,$$

$$\mathcal{D}^{ns} := \mathcal{D} \cap \mathrm{Div}^{g,ns}(\mathcal{C}) \,.$$

Note that any divisor $D \in \mathcal{D}^{ns}$ is reduced along P_0 (because P_0 is not contained in the support of D and the linear system $|D|$ consists merely of D). If a divisor $D = P_1 + \cdots + P_g$ as in the set $\mathcal{D}^{ns}$ appears in a relation (3.8), a new edge is inserted into the tree. (Other divisors might also lead to new edges: We ignore FP relations which involve a larger multiple of the large prime, we ignore non-special divisors, and we ignore divisors of degree $< g$.)

We have

$$\begin{aligned} \#\mathcal{D} &= \tbinom{\#\mathcal{F}+g-2}{g-1} \cdot \#\big(\mathcal{C}(\mathbb{F}_q) - (T \cup \mathcal{F} \cup \{P_0\})\big) \geq \tfrac{\#\mathcal{F}^{g-1}}{(g-1)!} \cdot q/4 \qquad \text{by (3.11)} \\ &\geq \tfrac{1}{4(g-1)!} \cdot q^{\frac{(g-1)^2}{g}} \cdot q = \tfrac{1}{4(g-1)!} \cdot q^{\frac{g^2-g+1}{g}} = \tfrac{1}{4(g-1)!} \cdot q^{g-1+\frac{1}{g}} \,. \end{aligned} \tag{3.12}$$

By our assumption that $q \geq (4 \cdot g! \cdot C)^g$, we have

$$\#\,\mathrm{Div}^{g,\mathrm{s}}(\mathcal{C}) \leq C q^{g-1} \leq \frac{1}{4g!} \cdot q^{g-1+\frac{1}{g}} \leq \frac{1}{8(g-1)!} \cdot q^{g-1+\frac{1}{g}} \,. \tag{3.13}$$

Inequalities (3.12) and (3.13) imply

$$\#\mathcal{D}^{ns} \geq \frac{1}{8(g-1)!} \cdot q^{g-1+\frac{1}{g}} \,.$$

Together with our assumption that $\#\,\mathrm{Cl}^0(\mathcal{C}) \leq 2q^g$, this implies that the probability that a relation (3.8) enlarges the tree is

$$\geq \frac{\#\mathcal{D}^{ns}}{\#\,\mathrm{Cl}^0(\mathcal{C})} \geq \frac{1}{16(g-1)!} \cdot q^{-(1-\frac{1}{g})} \,.$$

The expected number of relations (3.8) which have to be considered until the tree is enlarged is thus

$$\leq 16(g-1)! \cdot q^{1-\frac{1}{g}} .$$

This implies that the expected number of tries until the tree has $\lceil q^{1-\frac{1}{g}} \rceil$ edges is

$$\leq 16(g-1)! \cdot q^{1-\frac{1}{g}} \cdot \lceil q^{1-\frac{1}{g}} \rceil \leq 16(g-1)! \cdot (q+1)^{2-\frac{2}{g}} .$$

We now assume that $s \geq 2$ and a tree T with $< 2^{s-1} \cdot \lceil q^{1-1/g} \rceil$ edges containing a subtree T_{s-1} with $2^{s-2} \cdot \lceil q^{1-1/g} \rceil$ edges, has already been constructed. The task is again to derive a bound on the expected number of relations (3.8) needed until the tree is enlarged.

Similarly to above, let

$$\begin{aligned} \mathcal{D} := \big\{ P_1 + \cdots + P_g \in \mathrm{Div}^g(\mathcal{C}) |\ \forall i = 1, \ldots, g-2 : P_i \in \mathcal{F}, \\ P_{g-1} \in \mathcal{F} \cup T_{s-1},\ P_g \in \mathcal{C}(\mathbb{F}_q) - (T \cup \mathcal{F} \cup \{P_0\}) \big\} , \end{aligned}$$

$$\mathcal{D}^{ns} := \mathcal{D} \cap \mathrm{Div}^{g,ns}(\mathcal{C}) .$$

We now have

$$\begin{aligned} \#\mathcal{D} &= \big(\tbinom{\#\mathcal{F}+g-2}{g-1} + \tbinom{\#\mathcal{F}+g-3}{g-2} \cdot \#(T_{s-1} - \{*\}) \big) \cdot \#\big(\mathcal{C}(\mathbb{F}_q) - (T \cup \mathcal{F} \cup \{P_0\}) \big) \\ &\geq \big(\tfrac{\#\mathcal{F}^{g-1}}{(g-1)!} + \tfrac{\#\mathcal{F}^{g-2}}{(g-2)!} \cdot 2^{s-2} \cdot q^{1-1/g} \big) \cdot q/4 \\ &\geq \big(\tfrac{1}{(g-1)!} \cdot q^{\frac{(g-1)^2}{g}} + \tfrac{1}{(g-2)!} \cdot 2^{s-2} \cdot q^{\frac{(g-1)^2}{g}} \big) \cdot q/4 \\ &= \big(\tfrac{1}{4(g-1)!} + \tfrac{1}{4(g-2)!} \cdot 2^{s-2} \big) \cdot q^{g-1+\frac{1}{g}} . \end{aligned} \tag{3.14}$$

Together with (3.13), this implies

$$\#\mathcal{D}^{ns} \geq \frac{1}{4(g-2)!} \cdot 2^{s-2} \cdot q^{g-1+\frac{1}{g}} .$$

This implies that the probability that a relation (3.8) enlarges the tree is

$$\geq \frac{1}{8(g-2)!} \cdot 2^{s-2} \cdot q^{-(1-\frac{1}{g})} .$$

The expected number of relations (3.8) which have to be considered until the tree is enlarged is thus

$$\leq 8(g-2)! \cdot \frac{1}{2^{s-2}} \cdot q^{1-\frac{1}{g}} .$$

This implies that given any tree T_{s-1} with $2^{s-2} \cdot \lceil q^{1-\frac{1}{g}} \rceil$ edges, the expected number of tries until a tree T with $\min\{2^{s-1} \cdot \lceil q^{1-\frac{1}{g}} \rceil, N_{\max}\}$ edges is constructed is

$$\leq 16(g-2)! \cdot (q+1)^{2-\frac{2}{g}} .$$

We have $s \in \mathcal{O}(\log(q))$ as can be easily be seen: During the execution of the procedure we always have for $s \geq 2$

$$2q \geq \#(T - \{*\}) \geq \#(T_{s-1} - \{*\}) = 2^{s-2} \cdot \lceil q^{1-\frac{1}{g}} \rceil ,$$

i.e.

$$s \leq \log_2(q^{\frac{1}{g}}) + 3 = \frac{1}{\log(2) \cdot g} \cdot \log(q) + 3 \in \mathcal{O}(\log(q)) . \tag{3.15}$$

It follows that in total an expected number of $\mathcal{O}(\log(q) \cdot q^{2-\frac{2}{g}})$ relations (3.8) have to be considered until the tree has $N_{\max}$ edges. As each of these relations can be obtained in an expected time which is polynomially bounded in $\log(q)$, we conclude that a tree with $N_{\max}$ edges can be constructed in an expected time of

$$\tilde{\mathcal{O}}(q^{2-\frac{2}{g}}) .$$

Note that the depth of the tree is always bounded by s. In particular, as $s \in \mathcal{O}(\log(q))$, the depth of the tree is also in $\mathcal{O}(\log(q))$.

Step 4 – relation generation

We now assume we have constructed a tree T with $N_{\max} = \lceil q^{1-\frac{1}{g}+\frac{1}{g^2}} \rceil$ edges.

Similarly to above let

$$\mathcal{D} := \left\{ P_1 + \cdots + P_g \in \mathrm{Div}^g(\mathcal{C}) |\ \forall i = 1, \ldots, g : P_i \in \mathcal{F} \cup (T - \{*\}) \right\} ,$$

$$\mathcal{D}^{ns} := \mathcal{D} \cap D^{g,ns} .$$

Then $\mathcal{D}$ contains $\geq \frac{1}{g!} \cdot (\#\mathcal{F} + \#(T - \{*\}))^g \geq \frac{1}{g!} \cdot q^{g-1+\frac{1}{g}}$ elements. By the first two inequalities of (3.13), $\mathcal{D}^{ns}$ contains at least $\frac{3}{4g!} \cdot q^{g-1+\frac{1}{g}}$ elements. This means that the probability that the divisor D in relation (3.8) splits into elements of the factor base or vertices of the tree is

$$\geq \frac{3}{8g!} \cdot q^{-(1-\frac{1}{g})} .$$

The expected number of relations (3.10) which have to be considered in each call to the relation generation procedure is therefore in $\mathcal{O}(q^{1-\frac{1}{g}})$. As each relation (3.10) can be obtained in an expected time which is polynomially bounded in $\log(q^g)$, this means that the expected running time of the relation generation procedure is in $\tilde{\mathcal{O}}(q^{1-\frac{1}{g}})$.

In the relation generation part of the algorithm (Step 4 of the general algorithm), we have to generate $\tilde{\mathcal{O}}(\#\mathcal{F}) = \tilde{\mathcal{O}}(q^{1-\frac{1}{g}})$ "combined" Full relations. This means that the total running time of this step of the algorithm is in $\tilde{\mathcal{O}}(q^{2-\frac{2}{g}})$.

Step 5 – linear algebra

The linear algebra takes place on a sparse matrix with $\tilde{\mathcal{O}}(q^{1-1/g})$ rows and $\mathcal{O}(q^{1-1/g})$ columns.

As the tree has depth $\mathcal{O}(\log(q))$ and the size of the generating system is by assumption polynomially bounded in $\log(q)$, every row of the matrix $(R|S)$ contains only $\mathcal{P}oly(\log(q))$ non-zero entries. By Proposition 3.8, the computation can then be performed in an expected time of $\tilde{\mathcal{O}}(q^{2-2/g})$.

Final result

We have seen that Steps 1 – 5 of the algorithm all have an expected running time of $\tilde{\mathcal{O}}(q^{2-\frac{2}{g}})$. Moreover, we have argued in subsection 3.2.4 that after an expected number of $\mathcal{O}(\log\log(N)) \subseteq \mathcal{O}(\log(q))$ restarts of the computation of the matrix $(R|S)$, the linear algebra computation leads the solution to the discrete logarithm problem. This means that the total running time is in

$$\tilde{\mathcal{O}}(q^{2-\frac{2}{g}}) \ ,$$

in accordance with the statement in Proposition 3.16.

Storage requirements

Clearly there exists a function in $\tilde{\mathcal{O}}(q^{1-\frac{1}{g}+\frac{1}{g^2}})$ such that the storage requirements for the tree are bounded by this function for every run of the algorithm.

The storage requirements for the matrix are (for every run of the algorithm) bounded by a function in $\tilde{\mathcal{O}}(q^{1-\frac{1}{g}})$. Note again that this is the case because we restart the construction of the matrix every time the linear algebra computation fails instead of inserting a new row.

3.3.4 On the number of special divisors

The purpose of this subsection is to prove Proposition 3.20.

We consider curves of a fixed genus g over finite fields.

Let $\mathcal{C}$ be such a curve over $\mathbb{F}_q$. Let $\mathrm{Div}^g(\mathcal{C})$ be the set of effective divisors of degree g on $\mathcal{C}$, and let D_0 be an divisor of degree g on $\mathcal{C}$. We have the surjective map $\mathrm{Div}^g(\mathcal{C}) \longrightarrow \mathrm{Cl}^0(\mathcal{C}), D \mapsto [D] - [D_0]$. Note that the set of

special divisors of degree g is exactly the subset of $\mathrm{Div}^g(\mathcal{C})$ where the map to $\mathrm{Cl}^0(\mathcal{C})$ is not injective.

The number of special divisors is therefore bounded from above by $2\,(\#\,\mathrm{Div}^g(\mathcal{C}) - \#\,\mathrm{Cl}^0(\mathcal{C}))$, and it suffices to prove that $\#\,\mathrm{Div}^g(\mathcal{C}) - \#\,\mathrm{Cl}^0(\mathcal{C}) \in \mathcal{O}(q^{g-1})$.

We follow the exposition to the zeta-function in [Sti93]. Note however that we use different symbols for the indices.

Let $L = \prod_{i=1}^{2g}(1 - \alpha_i t) \in \mathbb{C}(t)$ be the L-polynomial of $\mathcal{C}$, let A_n be the number of effective divisors of degree n, let B_n be the number of prime divisors of degree n on $\mathcal{C}$, and let[7]

$$S_n := \sum_{i=1}^{2g} \alpha_i^n \ .$$

As the α_i can be arranged such that $\alpha_i \alpha_{g+i} = q$ for all $i = 1, \ldots, g$, we have

$$\#\,\mathrm{Cl}^0(\mathcal{C}) = L(1) \in q^g - S_1 \cdot q^{g-1} + \mathcal{O}(q^{g-1}) \ .$$

We thus have to show that

$$A_g \in q^g - S_1 \cdot q^{g-1} + \mathcal{O}(q^{g-1}) \ .$$

We will in fact show the more general statement

$$A_n \in q^n - S_1 \cdot q^{n-1} + \mathcal{O}(q^{n-1}) \tag{3.16}$$

for any fixed $n \in \mathbb{N}$.

Let us fix the following definition.

Definition 3.21 Let D be an effective divisor of degree n on $\mathcal{C}$ such that $D = \sum_{\ell=1}^{n} D_\ell$, where D_ℓ is a sum of e_ℓ prime divisors of degree ℓ. Then the vector $\underline{e} = (e_\ell)_\ell \in \mathbb{N}_0^n$ (with $\sum_\ell \ell e_\ell = n$) is called the *decomposition type* of D.

By sorting effective divisors of degree n by decomposition types, we obtain

$$A_n = \sum_{\underline{e}} \prod_{\ell} \binom{B_\ell + e_\ell - 1}{e_\ell} \ , \tag{3.17}$$

where the sum runs over all $\underline{e} \in \mathbb{N}_0^n$ with $\sum_\ell e_\ell \ell = n$ and the products run over $\ell \in \{1, \ldots n\}$. We have

$$A_1 = B_1 = q + 1 - S_1 \ ,$$

[7] The definition of S_n follows Equation (2.25) in [Sti93]. In [Sti93, Corollary V.1.17] an analogous definition is made with opposite sign.

which establishes the claim for $n = 1$. So let $n \geq 2$. By [Sti93, Proposition V.2.9], we have

$$\begin{aligned} B_\ell = \frac{1}{\ell} \cdot \sum_{m|\ell} \mu\big(\frac{\ell}{m}\big)(q^m - S_m) \quad &\in \frac{1}{\ell} \cdot q^\ell + \mathcal{O}(q^{\ell/2}) \\ &\subseteq \frac{1}{\ell} \cdot q^\ell + \mathcal{O}(q^{\ell-1}) \end{aligned} \tag{3.18}$$

for $\ell \geq 2$.

This implies that

$$\begin{aligned} A_n \quad &\in \sum_{\underline{e}} \frac{1}{e_1!}(q - S_1)^{e_1} \cdot \prod_{\ell \geq 2} \frac{1}{e_\ell!} \cdot \frac{1}{\ell^{e_\ell}} \cdot q^{\ell \cdot e_\ell} + \mathcal{O}(q^{n-1}) \\ &\subseteq \sum_{\underline{e}} \big(\prod_{\ell} \frac{1}{e_\ell!} \cdot \frac{1}{\ell^{e_\ell}}\big) \cdot (q^n - e_1 \cdot S_1 \cdot q^{n-1}) + \mathcal{O}(q^{n-1}) \; . \end{aligned}$$

In order to derive (3.16) it remains to be shown that

$$\sum_{\underline{e}} \prod_{\ell} \frac{1}{e_\ell!} \cdot \frac{1}{\ell^{e_\ell}} = 1 \tag{3.19}$$

and

$$\sum_{\underline{e}} e_1 \cdot \prod_{\ell} \frac{1}{e_\ell!} \cdot \frac{1}{\ell^{e_\ell}} = 1 \; . \tag{3.20}$$

Equation (3.19) is equivalent to

$$\sum_{\underline{e}} \prod_{\ell} \frac{n!}{e_\ell!} \cdot \frac{1}{\ell^{e_\ell}} = n! \; . \tag{3.21}$$

This is true because for any $\underline{e} \in \mathbb{N}_0^n$ with $\sum_\ell e_\ell \ell = n$, the set of permutations on n elements having exactly e_ℓ ℓ-cycles (for $\ell = 1, \ldots, n$) has $\prod_\ell \frac{n!}{e_\ell!} \cdot \frac{1}{\ell^{e_\ell}}$ elements.

We come to Equation (3.20). Note that we have a bijection

$$\begin{array}{ccc} \{\underline{e} \in \mathbb{N}_0^n |\ \sum_\ell e_\ell \ell = n,\ e_1 \neq 0\} & \longrightarrow & \{\underline{e}' \in \mathbb{N}_0^{n-1} |\ \sum_\ell e'_\ell \ell = n-1\} \; , \\ \underline{e} & \mapsto & \underline{e}' \end{array}$$

with $e'_1 = e_1 - 1$ and $e'_i = e_i$ for all $i = 1, \ldots, n-1$.

Equation (3.20) is then equivalent to

$$\sum_{\underline{e}'} \prod_{\ell} \frac{1}{e'_\ell!} \cdot \frac{1}{\ell^{e'_\ell}} = 1 \; , \tag{3.22}$$

where the sum runs over all $\underline{e}' \in \mathbb{N}_0^{n-1}$ with $\sum_\ell e'_\ell \ell = n-1$ and the products run over $\ell \in \{1, \ldots, n-1\}$. We already know that this equation is true.

3.3.5 Finding a generating system

The main purpose of this subsection is to show Proposition 3.17. Note that in Proposition 3.17 we only consider curves of a fixed genus, represented by plane models of bounded degree. In this subsection, we consider arbitrary curves over finite fields. We show below how one can efficiently compute a small system of degree 0 divisor classes which with a probability $\geq \frac{1}{2}$ generates the degree 0 class group, provided the L-polynomial is known (see Proposition 3.29). Proposition 3.17 follows from this statement and the fact – already mentioned in subsection 3.3.2 –, that one can then compute the L-polynomial in a time which is polynomially bounded in $\log(q)$ ([Pil90], [Pil91]). On our way to prove Proposition 3.29, we also show how one can efficiently compute uniformly randomly generated divisors of a specific degree, provided one knows the L-polynomial, a result which might be of independent interest (see Proposition 3.28).

As usual, curves are represented by plane models, and divisors are given in ideal representation.

Proposition 3.22 *Given a curve $\mathcal{C}$ over a finite field $\mathbb{F}_q$, represented by a plane model of degree d, and a natural number n one can with a randomized algorithm*

- *decide if $\mathcal{C}$ has an $\mathbb{F}_q$-rational point*
- *if this is the case compute such a point which is uniformly randomly distributed in $\mathcal{C}(\mathbb{F}_q)$*

in an expected time which is polynomially bounded in d and $\log(q)$.

Proof. As in subsection 2.5.4, we assume that the covering $x_{|\mathcal{C}} : \mathcal{C} \longrightarrow \mathbb{P}^1_{\mathbb{F}_q}$ is separable. As usual we set $r := \deg(x_{|\mathcal{C}})$.

By the Hasse-Weil bound, we have $\#\mathcal{C}(\mathbb{F}_q) \geq q + 1 - 2gq^{1/2}$. This means that for $q \geq 4g^2$, $\mathcal{C}(\mathbb{F}_q)$ is non-empty. The algorithm depends on a case distinction:

If $q < d^4$ we compute a list of the elements in $\mathcal{C}(\mathbb{F}_q)$ by iterating over all elements P of $\mathbb{P}^1(\mathbb{F}_q)$ and computing for each such P the divisor $x^{-1}_{|\mathcal{C}}(P)$ in free representation. If it turns out that $\mathcal{C}(\mathbb{F}_q)$ is empty, we output that this is the case, otherwise we choose one of the points in $\mathcal{C}(\mathbb{F}_q)$ uniformly at random. We have already argued in subsection 3.3.3 that this computation can be performed in an expected time which is polynomially bounded in d and q, that is, in an expected time which is polynomially bounded in d as q is also polynomially bounded in d by assumption.

If $q \geq d^4$ (such that $q > 4g^2$ and therefore $\mathcal{C}(\mathbb{F}) \neq \emptyset$), we proceed with the following algorithm.

Algorithm for computation of a uniformly randomly distributed rational point on a curve over a finite field

Input: A curve $\mathcal{C}$, represented by a plane model.

1. Choose a point $P \in \mathbb{P}^1(\mathbb{F}_q)$ uniformly at random.
2. Compute $x^{-1}(P)$ in free representation.
3. Let $P_1, \ldots, P_a$ be the distinct $\mathbb{F}_q$-rational points occurring in $x^{-1}(P)$.
   (The other prime divisors in the support of $x^{-1}(P)$ are ignored, and $a = 0$ is possible.)
4. Choose a number i in $\{1, \ldots, r\}$ uniformly at random.
5. If $i \leq a$, output P_i, otherwise go back to Step 1.

Without any assumption on q and g, this algorithm computes a uniformly randomly distributed point in $\mathcal{C}(\mathbb{F}_q)$ provided that this set is non-empty, as we show now.

The computation of Steps 1 – 5 can be performed in an expected time which is polynomially bounded in d and $\log(q)$ (see in particular Proposition 2.86).

Let us analyze the algorithm: After Step 4 the following always holds: The random variable (P, i) is uniformly distributed on the set $\mathbb{P}^1(\mathbb{F}_q) \times \{1, \ldots, r\}$, which has $(q+1) \cdot r$ elements. This means that in every iteration of the algorithm, the probability that the algorithm terminates in Step 5 is always $\frac{\#\mathcal{C}(\mathbb{F}_q)}{(q+1)\cdot r}$, and if this is the case, every point in $\mathcal{C}(\mathbb{F}_q)$ is chosen with the same probability of $\frac{1}{\#\mathcal{C}(\mathbb{F}_q)}$.

It follows that the output of the algorithm is a uniformly randomly distributed element in $\mathcal{C}(\mathbb{F}_q)$. Moreover, the expected number of iterations is $\frac{(q+1)\cdot r}{\#\mathcal{C}(\mathbb{F}_q)}$. In order to prove the proposition, we therefore have to show that the quantity $\frac{q+1}{\#\mathcal{C}(\mathbb{F}_q)}$ is polynomially bounded in d and $\log(q)$; we show that it is polynomially bounded in g (which in turn is polynomially bounded in d).

We have $\#\mathcal{C}(\mathbb{F}_q) \geq q+1-2gq^{1/2}$ by the Hasse-Weil bound. For $q \geq 16g^2$ we have $\#\mathcal{C}(\mathbb{F}_q) \geq \frac{q}{2} + 1$, and therefore $\frac{q+1}{\#\mathcal{C}(\mathbb{F}_q)} \leq 2$. On the other hand, for $q < 16g^2$ the quantity $\frac{q+1}{\#\mathcal{C}(\mathbb{F}_q)}$ is clearly polynomially bounded in g. □

Proposition 3.23 *Given a curve $\mathcal{C}$ over a finite field $\mathbb{F}_q$, represented by a plane model of degree d and a natural number n, one can with a randomized algorithm*

- *decide if $\mathcal{C}$ has a prime divisor of degree n*

- *if this is the case compute such a prime divisor which is uniformly randomly distributed in the set of all prime divisors of $\mathcal{C}$ of degree n*

in an expected time which is polynomially bounded in d, $\log(q)$ and n.

Proof. As in subsection 3.3.4, let B_n be the number of prime divisors of $\mathcal{C}$ of degree n. Then we have (cf. [Sti93, Corollary V.2.10.]):

$$B_n \geq \frac{q^n}{n} - (2+7g) \cdot \frac{q^{n/2}}{n}$$

For $q > (2+7g)^{2/n}$ we therefore have $B_n \geq 1$.

Similarly to the algorithm for the previous proposition, we have a case distinction according to $q > (2+7d^2)^2$.

In both cases we consider $\mathbb{F}_{q^n}$-rational points of $\mathcal{C}_{\mathbb{F}_{q^n}}$, which we represent as described in the beginning of subsubsection 2.5.4.5. Note that given such a point, one can compute the associated prime divisor (=closed point) by computing the intersection of the corresponding prime ideal with the function field $\mathbb{F}_q(\mathcal{C})$ as described in subsubsection 2.5.4.5 in a time which is polynomially bounded in d, n and $\log(q)$.

If $q \leq (2+7d^2)^2$, we iterate over all $\mathbb{F}_{q^n}$-rational points of $\mathcal{C}_{\mathbb{F}_{q^n}}$ (as described at the beginning of the proof of Proposition 3.22). For each such point we compute the associated prime divisor (=closed point) of $\mathcal{C}$, and check if this is a prime divisor of degree n. Like this we check if a prime divisor of degree n on $\mathcal{C}$ exists, and if this is the case, we uniformly randomly choose one.

So let now $q > (2+7d^2)^2$. Then in particular $q > (2+7d^2)^{2/n}$, and therefore there is a prime divisor of degree n on $\mathcal{C}$. Then the algorithm is also easy: We first choose an $\mathbb{F}_{q^n}$-rational point of $\mathcal{C}_{\mathbb{F}_{q^n}}$ uniformly at random compute the associated prime divisor (=closed point) of $\mathcal{C}$. If the prime divisor has degree n, we output it, otherwise we repeat this process.

Here we compute the $\mathbb{F}_{q^n}$-rational point using an algorithm for Proposition 3.22 with the modifications described in subsubsection 2.5.4.5. The expected time of one iteration is polynomially bounded in d, $\log(q)$ and n. We therefore have to show that the number of iterations is polynomially bounded in d, $\log(q)$ and n.

The number of points in $\mathcal{C}(\mathbb{F}_{q^n})$ such that the associated prime divisor has degree n is $n \cdot B_n$. Therefore the probability that a uniformly distributed point in $\mathcal{C}(\mathbb{F}_{q^n})$ does *not* give rise to a prime divisor of degree n is $\frac{\#\mathcal{C}(\mathbb{F}_{q^n}) - nB_n}{\#\mathcal{C}(\mathbb{F}_{q^n})}$ which is $\leq \frac{(2+9g)\cdot q^{n/2}}{q^n - 2gq^{n/2}}$. For $q \geq (4+20g)^{n/2}$ this is $\leq \frac{1}{2}$. □

We aim at giving an efficient algorithm to compute a random effective divisor which is uniformly randomly distributed in the set of all effective divisors of a specific degree. We want to use the fact that we can efficiently

compute prime divisors which are uniformly randomly distributed among all prime divisors of a specific degree. Let us for this first fix the following usual definition and the following notations which are inspired by the notations in [Heß05]. Let us for this fix a curve $\mathcal{C}$ over a finite field.

Definition 3.24 Let $n \in \mathbb{N}_0$ and $m \in \mathbb{N}$. Then an effective divisor on $\mathcal{C}$ of degree n which is the sum of prime divisors of degree at most m is called *m-smooth* or (n, m)-smooth.

Notation 3.25 Let $n \in \mathbb{N}_0$ and $m \in \mathbb{N}$. Then the number of effective divisors on $\mathcal{C}$ of degree n which split into prime divisors of degree $\leq m$ (resp. $= m$, resp. $\geq m$) is denoted by $\psi_{\leq}(n, m)$ (resp. $\psi_{=}(n, m)$, resp. $\psi_{\geq}(n, m)$). We also set $\psi(n, m) := \psi_{\leq}(n, m)$, the number of (n, m)-smooth divisors.

Note that in the notation of subsection 3.3.4, $\psi(n, n) = A_n$.

Lemma 3.26 *Given the L-polynomial of a curve of genus g over a finite field $\mathbb{F}_q$ and two natural number $m \leq n$, one can compute the numbers $\psi_{\leq}(n, m), \psi_{=}(n, m)$ and $\psi_{\geq}(n, m)$ in a time which is polynomially bounded in n, g and $\log(q)$.*

The *proof* of this lemma is inspired by the product formula for the zeta-function. As in subsection 3.3.4 let A_n be the number of effective divisors of degree n, and let B_n be the number of prime divisors of degree n. Then the zeta-function is

$$\sum_{i \in \mathbb{N}} A_i t^i = \sum_{D \text{ an eff. divisor}} t^{\deg D} =$$

$$\prod_{P \text{ a prime divisor}} (1 - t^{\deg(P)})^{-1} = \prod_{\ell \in \mathbb{N}} (1 - t^{\ell})^{-B_\ell} .$$

Similarly, for all $m \in \mathbb{N}$,

$$\sum_{i \in \mathbb{N}} \psi_{\leq}(i, m) t^i = \prod_{\ell \leq m} (1 - t^{\ell})^{-B_\ell} , \tag{3.23}$$

$$\sum_{i \in \mathbb{N}} \psi_{=}(i, m) t^i = (1 - t^{m})^{-B_m} , \tag{3.24}$$

and

$$\sum_{i \in \mathbb{N}} \psi_{\geq}(i, m) t^i = \prod_{\ell \geq m} (1 - t^{\ell})^{-B_\ell} . \tag{3.25}$$

The algorithm to compute $\psi_{\leq}(n, m)$ is as folllows:

Let the an L-polynomial $L(t)$ be given. We first compute $S_1, \ldots, S_m$ via Newton's identities (that is, via the equation $L'(t) = -L(t) \cdot (\sum_{i=1}^{\infty} S_i t^i)$)

from the coefficients of the L-polynomial. From these we compute $B_1, \ldots, B_m$ using (3.18). Then we compute $\prod_{\ell \le m}(1-t^\ell)^{B_\ell}$ and the inverse of its residue class modulo t^{n+1}, which is $\sum_{i=1}^{n} \overline{\psi}_{\le}(i,m)\,[t]^i_{(t^{n+1})}$.

The other algorithms operate similarly. □

Lemma 3.27 *Given the L-polynomial of a curve g over a finite field $\mathbb{F}_q$ and natural numbers n, m with $m \le n$, one can with a randomized algorithm compute in an expected time which is polynomially bounded in g, n and $\log(q)$ a random tuple $\underline{e} \in \mathbb{N}_0^n$ with $\sum_\ell e_\ell \ell = n$ whose distribution is equal to the distribution of the decomposition type of a random effective divisor on $\mathcal{C}$ which is uniformly distributed among all (n,m)-smooth divisors.*

Proof. The algorithm operates in a recursive way and is based on partitioning the set of (n,m)-smooth divisors into subsets, according to how many prime divisors (with multiplicities) of degree m occur in an (n,m)-smooth divisor.

If $m \ge 2$ and $\ell \in \{0, \ldots, \lfloor \frac{n}{m} \rfloor\}$, there are

$$\psi_=(\ell m, m) \cdot \psi(n - \ell m, m-1)$$

(n,m)-smooth divisors of the form $\sum_{i=1}^{\ell} P_i + D'$, where the P_i are prime divisors of degree m and D' is an $m-1$-smooth divisor.

Algorithm to compute a random tuple reflecting the distribution type of a random smooth divisor

Input: L, the L-polynomial of a curve over a finite field and two natural numbers n, m. (The algorithm is called by $A(L,n,m)$).

If $m = 1$, output $(n, 0, \ldots, 0) \in \mathbb{N}_0^n$. Otherwise:

1. Compute $\psi(n,m)$ and the numbers $a_\ell \longleftarrow \psi_=(\ell m, m) \cdot \psi(n - \ell m, m-1)$ for $\ell = 0, \ldots, \lfloor \frac{n}{m} \rfloor$.
2. Let $b_\ell \longleftarrow \sum_{i=0}^{\ell} a_i$ for $i = 0, \ldots, \lfloor \frac{n}{m} \rfloor$; let $b_{-1} \longleftarrow 0$.
3. Choose a natural number $x \le \psi(n,m)$ uniformly at random.
4. Determine ℓ such that $x \in [b_{\ell-1} + 1, b_\ell]$.
5. Output

$$(0, \ldots, 0, \ell, 0, \ldots, 0) + (A(L, n - \ell m, m-1) \,|\, \underline{0}) \in \mathbb{N}_0^n \,,$$

where the non-trivial entry in the first tuple is at index m and $(A(L, n - \ell m, m-1)\ \underline{0})$ is the concatenation of the output of the algorithm applied to $L, n - \ell m, m-1$ and the zero-tuple of length ℓm.

By the remarks above the algorithm, the distribution of the random variable ℓ in the algorithm is equal to the distribution of the number of (n,m)-smooth divisors being the sum of ℓ prime divisors of degree m and an $m-1$-smooth divisor.

It follows by induction on m that the algorithm operates correctly. Moreover, the running time is as claimed by Lemma 3.26. □

We now easily obtain:

Proposition 3.28 *Given a curve $\mathcal{C}$ over a finite field $\mathbb{F}_q$, represented by a plane model of degree d as well as its L-polynomial, one can with a randomized algorithm compute a random effective divisor of degree n which is uniformly randomly distributed in the set of all effective divisors of degree n on $\mathcal{C}$ in an expected time which is polynomially bounded in d, $\log(q)$ and n.*

Proof. Given the previous statements, the algorithm for this proposition is straightforward: We compute a random tuple $\underline{e} \in \mathbb{N}_0^n$ with $\sum_\ell e_\ell \ell = n$ whose distribution is equal to the distribution of the decomposition type of a random effective divisor on $\mathcal{C}$ which is uniformly distributed among all effective divisors of degree n.

Then for each $\ell = 1, \ldots, n$, we compute e_ℓ prime divisors $P_{\ell,i}$ which are uniformly distributed among the set of all prime divisors of degree ℓ.

We output the divisor $\sum_\ell \sum_i P_{\ell,i}$.

By Proposition 3.23 and Lemma 3.27 these computations can be performed in the claimed expected running time. □

Proposition 3.29 *Given a curve $\mathcal{C}$ over a finite field $\mathbb{F}_q$, represented by a plane model of degree d as well as its L-polynomial and a divisor D_0 of degree 1 whose height is polynomially bounded in d and $\log(q)$, one can with a randomized algorithm compute a uniformly randomly distributed element of $\mathrm{Cl}^0(\mathcal{C})$, represented by an along D_0 reduced divisor, in an expected time which is polynomially bounded in d and $\log(q)$.*

Proof. Every divisor of degree $\geq 2g-1$ is non-special. Therefore, if D is an effective divisor which is uniformly distributed among all divisors of some degree $n \geq 2g-1$, $[D - D_0]$ is uniformly distributed in $\mathrm{Cl}^0(\mathcal{C})$. Moreover, $2g-1 \leq (d-1)(d-2)-1$.

In order to compute the desired uniformly distributed divisor class, we first compute a uniformly distributed effective divisor D of degree $(d-1)(d-2)-1$, and then we compute its reduction along D_0.

The computations can be performed in the claimed expected running time by the previous proposition and Proposition 2.110. □

We now prove:

Proposition 3.30 *There exists a randomized algorithm such that the following holds: Given a curve $\mathcal{C}$ over a finite field $\mathbb{F}_q$, represented by a plane model of degree d, as well as its L-polynomial and a divisor D_0 of degree 1 whose height is polynomially bounded in d and $\log(q)$, the algorithm computes a system of elements $c_1, \ldots, c_u \in \mathrm{Cl}^0(\mathcal{C})$, represented by along D_0 reduced divisors, where $u := e\ell$ with $e := \lceil \log_2(\#\mathrm{Cl}^0(\mathcal{C}))\rceil$ and $\ell := \lceil \log_2(e) + 1\rceil$. Moreover, the expected running time is polynomially bounded in d and $\log(q)$, and with a probability $\geq \frac{1}{2}$, the system $c_1, \ldots, c_u$ is a generating system of $\mathrm{Cl}^0(\mathcal{C})$.*

The proposition immediately follows from the preceding proposition and the following lemma (cf. the proof of [Heß05, Lemma 50]).

Lemma 3.31 *Let G be a finite abelian group with N elements, let $e := \lceil \log_2(N)\rceil, \ell := \lceil \log_2(e) + 1\rceil, u := e\ell$, and let $g_1, \ldots, g_u$ be uniformly distributed random elements of G. Then with a probability of $\geq \frac{1}{2}$, $g_1, \ldots, g_u$ generate G.*

Proof. Let first H be a proper subgroup of G, and let $g_1, \ldots, g_a$ be uniformly randomly distributed elements from G. Then as $\frac{\#H}{\#G} \leq \frac{1}{2}$, with a probability $\leq \frac{1}{2^a}$, all g_i lie in H, that is, with a probability $\geq 1 - \frac{1}{2^a}$, $H \subsetneq \langle H, g_1, \ldots, g_a\rangle$.

It follows by induction on b: Let $a, b \in \mathbb{N}$, and let $g_1, \ldots, g_{ab}$ be uniformly randomly distributed elements from G. Then with a probability $\geq (1 - \frac{1}{2^a})^b \geq 1 - \frac{b}{2^a}$, $\langle g_1, \ldots, g_a\rangle$ contains at least $\min\{N, 2^b\}$ elements.

With $b := e$ and $a := \ell$, we have $2^b \geq N$ and $\frac{b}{2^a} \leq \frac{1}{2}$. The lemma thus follows. □

Proposition 3.17 follows from Proposition 3.30 and the fact – already mentioned in subsection 3.3.2 – that for curves of a bounded degree over finite fields, one can compute the L-polynomial in polynomially bounded time in $\log(q)$, where q is the base field.

3.4 Index calculus for curves of lower-bounded genus

3.4.1 Introduction and results

Previous works on computing discrete logarithms in class groups of curves over finite fields have been directed in two distinct directions: In one class of works, curves of a fixed genus are considered (cf. [Gau00], [Thé03], [GTTD07] and also [Die06]). In the other class of works, only curves whose genus is is a certain sense large against the bit-length of the cardinality of the ground field are considered (cf. e.g. [Cou01], [Eng02], [EG02], [Heß05]). (Often more

conditions are imposed, for example that the curves are hyperelliptic and given in imaginary quadratic representation).

The algorithms in the first kind of works give rise to expected running times which are exponential but not subexponential in q^g whereas in the second kind of works expected running times which are subexponential in q^g are established.

In this section, we break with this tradition by proving the following result.

Theorem 2 *Let some natural number $g_0 \geq 2$ be fixed. Then the discrete logarithm problem in the degree 0 class groups of curves of genus $\geq g_0$ over finite fields can with a randomized algorithm be solved in an expected time of*

$$\tilde{\mathcal{O}}((q^g)^{\frac{2}{g_0}(1-\frac{1}{g_0})}) ,$$

where $\mathbb{F}_q$ is the ground field and g the genus of the curve.

Here as well as in the rest of this section, we assume that the curve is represented by a plane model of degree $\mathcal{O}(g)$ (which is possible by Proposition 2.6). As in Proposition 2.117, the divisor classes are represented by an along a divisor D_0 reduced divisors, where the height of D_0 is polynomially bounded in d, that is, polynomially bounded in g.

Note that this theorem clearly generalizes Theorem 1. Of course, the expected running time is exponential and not subexponential in q^g because we cannot even for any fixed genus prove a subexponential expected running time.

It is natural to bound the running time in terms of the number of elements in the degree 0 class group instead of q^g. We show that the corresponding statement then holds too:

Theorem 3 *Let some natural number $g_0 \geq 2$ be fixed. Then the discrete logarithm problem in the degree 0 class groups of curves $\mathcal{C}/\mathbb{F}_q$ of genus $\geq g_0$ over finite fields can with a randomized algorithm be solved in an expected time of*

$$\tilde{\mathcal{O}}((\#\operatorname{Cl}^0(\mathcal{C}))^{\frac{2}{g_0}(1-\frac{1}{g_0})}) .$$

By applying Theorem 3 with $g_0 = 3$ we obtain that one can solve the discrete logarithm problem in degree 0 class groups of curves of genus ≥ 3 in an expected time of

$$\tilde{\mathcal{O}}(\#\operatorname{Cl}^0(\mathcal{C}))^{\frac{4}{9}}) = \tilde{\mathcal{O}}((\#\operatorname{Cl}^0(\mathcal{C}))^{\frac{1}{2}})^{\frac{8}{9}}) . \tag{3.26}$$

Outline of the proof

We give an outline of the proof of Theorems 2 and 3.

Below we prove the following result.

Proposition 3.32 *Let $C > 0$ be fixed. Then there exists some $g_1 \in \mathbb{N}$ such that one can solve the discrete logarithm problem in the degree 0 class groups of curves of genus $\geq g_1$ in an expected time of $\mathcal{O}((q^g)^C)$, where q is the cardinality of the ground field and g is the genus.*

We show how one can obtain Theorems 2 and 3 via this result and Theorem 1. We first consider the statement in Theorem 2.

Let g_0 be a natural number ≥ 2, and let $C := \frac{2}{g_0}(1 - \frac{1}{g_0})$. Let g_1 be as in the proposition for this constant C.

Then given an instance of the discrete logarithm problem with a curve of genus $g \geq g_0$, we first compute its genus g (this is possible in poly-logarithmic time by Proposition 2.96). Then we proceed with a case distinction. If the genus is $\geq g_1$ we apply an algorithm which satisfies Proposition 3.32. If the genus is $< g_1$, we apply an algorithm which satisfies Theorem 1 for genus g curves.

We now come to Theorem 3.

Note first that $\#\operatorname{Cl}^0(\mathcal{C}) \in [(\sqrt{q}-1)^{2g}, (\sqrt{q}+1)^{2g}]$. This implies that

$$\#\operatorname{Cl}^0(\mathcal{C}) \geq (\sqrt{q}-1)^{2g} = (q^g)^{\frac{\log((\sqrt{q}-1)^2)}{\log(q)}} \geq (q^g)^{\frac{\log((\sqrt{2}-1)^2)}{\log(2)}} .$$

Let now again g_0 be a natural number ≥ 2, and this time let $C := \frac{2}{g_0}(1 - \frac{1}{g_0}) \cdot \frac{\log((\sqrt{2}-1)^2)}{\log(2)}$, such that $\#\operatorname{Cl}^0(\mathcal{C})^{\frac{2}{g_0}(1-\frac{1}{g_0})} \geq (q^g)^C$. Let g_1 again be as in Proposition 3.32. Now again for curves of genus $\geq g_1$ we apply an algorithm satisfying Proposition 3.32 and for curves of genus $g < g_1$ (and of course $g \geq g_0$) we apply an algorithm satisfying Theorem 1 for genus g curves.

3.4.2 The algorithm

We now outline an algorithm for Proposition 3.32, accompanied by statements on running times. The algorithm is substantially easier to state than the index calculus algorithm in Section 3.3: It is a "basic" index calculus algorithm, and there is no precomputation like for example a construction of a graph of large prime relations. The algorithm is closely related to the algorithms in [Heß05], and we make use of various results from [Heß05].

We again wish to follow the "general algorithm" in subsection 3.2.3. Recall that the input to the "general algorithm" not only consists of the

curve and the two group elements a, b but also of a system of elements $c_1, \ldots, c_u$ of the degree 0 class group. We could now proceed as in Section 3.3: Using a very small system which with probability $\geq \frac{1}{2}$ is a generating system and terminating the algorithm if a predefined time bound is reached. However, in the situation we are concerned with here, there is an easier solution: It is shown in [Heß05] that one can compute a generating system of a size which is polynomially bounded in q and g in an expected time which is also polynomially bounded in q and g (see [Heß05, Theorem 34 and Algorithm 35]).

Note that generally if any subroutine of our algorithm has an expected running time which is polynomially bounded in q and g, then this subroutine in particular has an expected running time of $\mathcal{O}(q^{Cg})$ for g large enough. Thus if such a subroutine is executed once or even a number of times which is polynomially bounded in q and g, then the running time of this subroutine is not critical for the establishment of the desired result.

We now describe the various subroutines for the "general algorithm"; we also discuss the expected running times of the routines.

Obviously, we only have to establish Proposition 3.32 for constants C of the form $C = \frac{1}{k}$ for $k \in \mathbb{N}$. So let $k \in \mathbb{N}$ be fixed, and let $C := \frac{1}{k}$.

As usual, let us fix an instance consisting of a curve $\mathcal{C}/\mathbb{F}_q, a, b \in \mathrm{Cl}^0(\mathcal{C})$ and a generating system $c_1, \ldots, c_u$.

a) Computation of the group order

A. Lauder and D. Wan have shown in [LW] that one compute the order of the degree 0 class group of a curve over $\mathbb{F}_q$ given by a plane model of degree d in a time which is polynomially bounded in q and d. (In fact, if $q = p^e$ with p prime and $n \in \mathbb{N}$, the running time is polynomially bounded in p, e and d.) Note that as by our assumption $d \in \mathcal{O}(g)$, the running time is in particular polynomially bounded in q and g.

c) Construction the factor base

We first compute the genus g. We fix a "smoothness bound" $m := \lceil \frac{g}{8k} \rceil$, and let the factor base $\mathcal{F}$ be the set of prime divisors of degree $\leq m$. Let us assume (wlog.) that in Step 2 of the "general algorithm", the ideal representation is set up with respect to the covering $x_{|\mathcal{C}} : \mathcal{C} \longrightarrow \mathbb{P}^1_{\mathbb{F}_q}$. Then we construct the set of prime divisors by iterating over all prime divisors of degree $\leq m$ on $\mathbb{P}^1_{\mathbb{F}_q}$ and considering the preimages under the covering $x_{|\mathcal{C}} : \mathcal{C} \longrightarrow \mathbb{P}^1_{\mathbb{F}_q}$.

The factor base has $\leq r \cdot (q+1)^m$ elements, where $r := \deg(x_{|\mathcal{C}})$. For $g \geq 8k$, we have $m \leq \frac{g}{4k}$. As further $r \leq d \in \mathcal{O}(g)$, the size of the factor

base is then (for $g \geq 8k$) in $\tilde{\mathcal{O}}(q^{\frac{g}{4k}})$, and the expected running time is also in $\tilde{\mathcal{O}}(q^{\frac{g}{4k}})$.

d) Relation generation

Let $c_1, \ldots, c_u \in \mathrm{Cl}^0(\mathcal{C})$ be a generating system, and let $c \in \mathrm{Cl}^0(\mathcal{C})$. Then we proceed as follows: We choose $s_1, \ldots, s_u$ uniformly at random and then choose an effective divisor D uniformly at random in the class $s_1c_1 + \cdots + s_uc_u + c + (2g-1) \cdot [D_0]$, which has degree $2g-1$. We compute the free representation of D. If D is m-smooth, we have obtained a relation. Otherwise we repeat the process.

Each iteration can be performed in an expected time of $\mathcal{P}oly(g \log(q)) \cdot u$. Note also that as the random divisor class $s_1c_1 + \cdots + s_uc_u + c + (2g-1)[D_0]$ is uniformly distributed in the set of divisor classes of degree $2g-1$ and divisors of degree $\geq 2g-1$ are non-special, the random divisor D is uniformly distributed in the set of all effective divisors of degree $2g-1$.

In order to bound the expected running time of this relation generation procedure, we need a lower bound on the probability that a uniformly distributed random divisor of degree $2g-1$ is m-smooth. For this we can use [Heß05, Theorem 8] which gives a much more precise statement than the one we need. In fact, just from the fact that $m \in \Omega(2g-1)$, we learn from [Heß05, Theorem 8] that the probability in question is in $g^{\Omega(1)}$.

This establishes that the expected running time of the relation generation procedure is in $\mathcal{P}oly(g \cdot \log(q)) \cdot u$.

Analysis of the algorithm

It is now easy to establish that the algorithm gives rise to Proposition 3.32.

First, in Step 1 the expected running time is polynomially bounded in q and d for the computation of the group order. The expected running time for the factorization is in $L_{\#\,\mathrm{Cl}^0(\mathcal{C})}[\frac{1}{2}, \mathcal{O}(1)] = L_q[\frac{1}{2}, \mathcal{O}(1)]$.

Let $g_2 \in \mathbb{N}$ be a constant such that for $g \geq g_2$ the size of the generating systems in $\mathcal{O}(q^{\frac{g}{4k}})$ and expected running time of the procedure to compute a generating system is in $\mathcal{O}(q^{\frac{g}{4k}})$ too. Let us restrict to instances with $g \geq g_2$. Let us – as already indicated in subroutine c) above – restrict to instances which additionally satisfy $g \geq 8k$, such that the size of the factor base is in $\tilde{\mathcal{O}}(q^{\frac{g}{4k}})$ and the expected running time of the subroutine for generation of the factor base too. The expected time for the generation of the relation matrix is in $\mathcal{P}oly(g \cdot \log(q)) \cdot u \cdot \tilde{\mathcal{O}}(\#\mathcal{F}) \subseteq \tilde{\mathcal{O}}(q^{\frac{g}{2k}})$. The size of the relation matrix is in $\tilde{\mathcal{O}}(q^{\frac{g}{4k}})$, and the number of non-zero entries is in $\tilde{\mathcal{O}}(q^{\frac{g}{2k}})$. Therefore, the expected time for the linear algebra is in $\tilde{\mathcal{O}}(q^{\frac{3g}{4k}})$.

In total, for g large enough, the expected running time is in $\mathcal{O}(q^{\frac{g}{k}})$. This establishes the result.

3.5 Index calculus for elliptic curves over extension fields

3.5.1 Introduction and results

In this section we show that the index calculus technique can also be successfully applied to the discrete logarithm problem in elliptic curves over finite non-prime fields.

We establish the following theorem.

Theorem 4 *Let $\epsilon > 0$. Then the discrete logarithm problem in the groups of rational points of elliptic curves over finite fields $\mathbb{F}_{q^n}$ with $(2+\epsilon)\cdot n^2 \leq \log_2(q)$ can with a randomized algorithm be solved in an expected time which is polynomially bounded in q.*

Here and in the following, q is a prime power and n a natural number. The theorem leads to the following corollary.

Corollary 3.33 *Let again $\epsilon > 0$, and let $a > 2 + \epsilon$. Then the discrete logarithm problem in the groups of rational points of elliptic curves over finite fields $\mathbb{F}_{q^n}$ with $(2 + \epsilon) \cdot n^2 \leq \log_2(q) \leq a \cdot n^2$ can be solved in an expected time of*

$$e^{\mathcal{O}(1)\cdot(\log(q^n))^{2/3}} .$$

This corollary follows easily from the theorem. Indeed, restricted to instances as in the corollary the elliptic curve discrete logarithm problem can be solved in an expected time which is polynomially bounded in

$$q = 2^{\log_2(q)} = 2^{(\log_2(q))^{(1+1/2)\cdot 2/3}} \leq 2^{(\sqrt{a}\cdot n \log_2(q))^{2/3}} .$$

We note that the corollary establishes for the first time that there exists a sequence of finite fields of increasing size such that the elliptic curve discrete logarithm problem over these fields can be an expected time which is subexponential in the field size (or the group order or the input length).

When fixing the extension degree, we obtain the following result, which is analogous to Theorem 1 in Section 3.3.

Theorem 5 *Let some natural number $n \geq 2$ be fixed. Then the discrete logarithm problem in the groups of rational points of elliptic curves over finite fields $\mathbb{F}_{q^n}$ can with a randomized algorithm be solved in an expected time of*

$$\tilde{\mathcal{O}}(q^{2-\frac{2}{n}}) .$$

Similarly to the algorithm for Theorem 1 the algorithm has storage requirements of $\tilde{\mathcal{O}}(q^{1-\frac{1}{n}+\frac{1}{n^2}})$.

On the proofs

We give here a very brief overview of the algorithms leading to the above theorems.

As already mentioned, the algorithms again follow the index calculus method. From a macroscopic point of view we proceed as in the previous index calculus algorithms in this work: We first compute a "potential generating system", and then we follow the "general algorithm" in subsection 3.2.3.

Note that in Section 3.2 we consider the discrete logarithm problem in degree 0 class groups of curves, but the general approach outlined there also applies to elliptic curves if the consider the canonical isomorphism $E(K) \longrightarrow \mathrm{Cl}^0(E), P \mapsto [P - O]$ for an elliptic curve E/K. Note here also that for computational purposes we represent degree 0 divisor classes on any curve by divisors which are reduced along a fixed divisor of degree 1. Thus if we fix the divisor O on an elliptic curve, the class $[P - O]$ is represented by P (for $P \neq O$). Thus to say that we compute in the degree 0 class group $\mathrm{Cl}^0(E)$ means by definition that we compute in $E(K)$. In the following, we use the corresponding capital letters to denote points corresponding to divisor classes in the general algorithm (which are denoted there by small letters). Divisors (apart from points in $E(K)$ of course) are not used in the following.

The most challenging aspects in the proofs of the above results lie (as might be expected) in finding appropriate factor bases and efficient relation generation procedures as well as in analyzing the procedures.

For the definition of the factor base and the relation generation we depart from the previous algorithms in this work: In contrast to the usual approach in index calculus, we define the factor base in an *algebraic* rather than an *arithmetic* way, and for the relation generation we solve systems of multivariate polynomial equations.

Let us discuss the approach for Theorem 4:

Let $E/\mathbb{F}_{q^n}$ be an elliptic curve. Then we compute a covering $\varphi : E \longrightarrow \mathbb{P}^1_k$ of degree 2 which satisfies $\varphi \circ [-1] = \varphi$ as well as certain additional

conditions. The factor base is then given by

$$\{P \in E(\mathbb{F}_{q^n}) \mid \varphi(P) \in \mathbb{F}_q\}\,, \tag{3.27}$$

where as always we identify $\mathbb{F}_q = \mathbb{A}^1(\mathbb{F}_q)$ with $\mathbb{P}^1(\mathbb{F}_q) - \{0\}$.

The relation generation relies on an algorithm which we call *decomposition algorithm*. Given an elliptic curve $E/\mathbb{F}_{q^n}$ the extension degree n, a covering φ as above and some point $P \in E(\mathbb{F}_{q^n})$, this algorithm either fails or outputs a tuple $(P_1, \ldots, P_n) \in E(\mathbb{F}_{q^n})^n$ with $\varphi(P_i) \in \mathbb{F}_q$ or $P_i = O$ for $i = 1, \ldots, n$ such that

$$P_1 + \cdots + P_n = P\,.$$

The decomposition algorithm is based on solving multivariate systems of polynomial equations over $\mathbb{F}_q$. Of course it fails if there is no such tuple $(P_1, \ldots, P_n)$. But it also fails if the solution set of the associated multivariate system is not *zero-dimensional*. We remark here that the most difficult part of the proof is to show that for a uniformly distributed point $P \in E(\mathbb{F}_{q^n})$ with a sufficiently high probability the solution sets to these systems are indeed zero-dimensional. In order to prove this result, we pass to higher-dimensional schemes over $\mathbb{F}_q$ by using so-called Weil restrictions. The proof then relies crucially on intersection theory in $(\mathbb{P}^1_{\mathbb{F}_{q^n}})^n$.

For Theorem 5 we proceed similarly. But now we let the factor base be an appropriate subset of the set defined in (3.27). We then proceed with a double large prime variation, similarly to the algorithm for curves of a fixed genus in subsection 3.3.2.

Some historical comments

In Feb. 2004 I. Semaev put a preprint on the server of the International Association for Cryptographic Research (IACR) in which he discussed the possibility of index calculus in the groups of rational points on elliptic curves over prime fields ([Sem04]). In his work, Semaev defined the factor base via an upper bound on the x-coordinates of points.

He also introduced so-called *summation polynomials*: Let E be an elliptic curve over a field K, given by a Weierstraß model, and let $m \in \mathbb{N}$, $m \geq 2$. Then the m-th summation polynomial as defined by Semaev is an irreducible polynomial $f \in K[x_1, \ldots, x_m]$ such that for the following holds: Given $P_1, \ldots, P_m \in E(\overline{K}) - \{O\}$, we have

$$f(x(P_1), \ldots, x(P_m)) = 0 \longleftrightarrow \exists \epsilon_1, \ldots, \epsilon_m \in \{1, -1\} \,:\, \epsilon_1 P_1 + \cdots + \epsilon_m P_m = O\,,$$

where we identify $\mathbb{A}^1(\overline{K}) = \mathbb{P}^1(\overline{K}) - \{\infty\}$ with $\overline{K}$. These summation polynomials have degree 2^{m-2} in each variable.

Now, any algorithm to determine solutions with "small coordinates" for multivariate equations of high degree would give rise to an algorithm for relation generation. However, no efficient algorithm for this task is known (except for very special equations), and therefore, Semaev's approach does (currently) not lead to an algorithm which is faster than generic algorithms to solve discrete logarithm problems.

Semaev's work lead however both P. Gaudry and the author to reflect on the question whether a similar approach over extension fields might not give algorithms which (at least asymptotically) are faster than generic algorithms.

Both had previous work on G. Frey's "Weil descent" idea in mind, which is based on the so-called *Weil restriction*: Given any separable finite field extension $K|k$ and an abelian variety A over K, the *Weil restriction* of A with respect to $K|k$ is an abelian variety $\mathrm{Res}_k^K(A)$ such that for any k-scheme Z one has $\mathrm{Res}_k^K(A)(Z) \simeq A(Z)$ in a functorial way, thus in particular $\mathrm{Res}_k^K(A)(k) \simeq A(K)$. The Weil restriction of A with respect to $K|k$ has dimension $[K : k] \cdot \dim(A)$ and is attached to A in a functorial way. Very generally speaking, Frey's idea was as follows: Let K and k be finite fields, and let E be an elliptic curve over k. Now maybe one can use the fact that $\mathrm{Res}_k^K(E)(k)$ is an abelian variety of higher dimension over a smaller field to attack the discrete logarithm problem in $E(K)$. More concretely, his idea was to find curves on $\mathrm{Res}_k^K(E)(k)$ of low genus and to "pull-back" the discrete logarithm problem in $E(k)$ to the Jacobians of such curves, where one then might use index calculus methods. These ideas lead for example to the so-called *GHS-attack*[8] (see for example [GHS02], [Heß03], [Die03]) and more generally to what the author calls *covering attacks* (see for example the appendix of [Die03]). Now both Gaudry and the author wanted to use the Weil restriction directly, without a transfer to any further curve.

Already in March 2004 Gaudry put a first version of his work on the archive of the IACR ([Gau04a]). This work in fact not only focuses on elliptic curves but includes also a discussion on the discrete logarithm problem in any abelian variety, in particular including Jacobians of hyperelliptic curves.

For elliptic curves, Gaudry fixes the extension degree n and lets q go to infinity. Using a single-large prime variation Gaudry originally obtained, on a heuristic basis, an expected running time of

$$\tilde{\mathcal{O}}(q^{2-\frac{2}{n-1/2}}) .$$

Later, using a double large prime variation, he improved this to a heuristic expected running time of

$$\tilde{\mathcal{O}}(q^{2-\frac{2}{n}}) .$$

[8] GHS stands for Gaudry, Heß, Smart.

The author on the other hand tried if a common variation of n and q would lead to a sequence of finite fields such that the elliptic curve discrete logarithm problem over these fields would become subexponential.

Gaudry then presented his results at the Elliptic Curve Cryptography Workshop (ECC) 2004 in Bochum. In his presentation he also mentioned the results by the author, and this lead to an ad-hoc presentation by the author on the next day. The author then in particular argued – again on a heuristic basis – that restricted to instances over finite fields $\mathbb{F}_{q^n}$ with $n \in \Theta(\log(q))$, one can solve the elliptic curve discrete logarithm problem in an expected time of

$$e^{\mathcal{O}(1)\cdot(\log(q^n))^{3/4}}.$$

In the meantime, Gaudry's work has been accepted for publication in the Journal of Symbolic Computation. The newest publically available version of Gaudry's work is from Oct. 2004 ([Gau04b]). We would like to make some comments on this version: This work contains in particular two statements which are called "Theorems". Theorem 1 is concerned with index calculus on abelian varieties of a fixed dimension, Theorem 2 is Theorem 5 above. However, in the setting of [Gau04b] Theorem 1 is incorrect and Theorem 2 is correct as we show below but in [Gau04b]) this result in not established but only *heuristically argued to be true*.

We first explain why in the context of [Gau04b] [Gau04b, Theorem 1] is incorrect: It is claimed that for fixed n one can solve the discrete logarithm problem in abelian varieties of dimension n in an expected time of $\tilde{O}(q^{2-\frac{2}{n}})$, where as usual $\mathbb{F}_q$ is the ground field. We conjecture that this statement is correct *provided one represents the abelian varieties appropriately*. However, the statement is clearly not correct if one follows the information on the representation given in [Gau04b], and additionally, during the description of the representation a severe mathematical mistake is made:

On page 3 of [Gau04b] it is stated that an abelian variety A over $\mathbb{F}_q$ is represented as follows: An open part U of A is embedded in $\mathbb{A}^{n+m}_{\mathbb{F}_q}$ such that the projection to the first n coordinates $U \longrightarrow \mathbb{A}^n_{\mathbb{F}_q}$ is quasi-finite. Then the author states that "a coordinate system with these properties is called Noether normalization of the variety". With other words: It is concluded that the map $U \longrightarrow \mathbb{A}^n_{\mathbb{F}_q}$ is finite, but this is incorrect.

However, even if we restrict ourselves to a representation as above with a finite map, the alleged running time cannot be accurate for a very simple reason: No bound in m is given, thus the description can be arbitrarily complex. We note here that it would however not suffice to fix a bound on m. One should for example also fix some upper bound on the number of defining equations and their degrees. But even if one has done so, one should still also consider the group law.

We are now coming to the shortcomings in the establishment of [Gau04b, Theorem 2].

- The factor base is a subset of the set $\{P \in E(\mathbb{F}_{q^n}) \mid x(P) \in \mathbb{F}_q\}$. Under the bijection $E(\mathbb{F}_{q^n}) \simeq \mathrm{Res}_{\mathbb{F}_q}^{\mathbb{F}_{q^n}}(E)(\mathbb{F}_q)$ this set corresponds to the set of rational points in a particular 1-dimensional subscheme of the Weil restriction. In [Gau04b] it is written "For simplicity we assume that [this subscheme] is irreducible." But what if it is not irreducible? For example, it might be that it only contains one irreducibility component but is geometrically reducible. In this case the factor base might be very small.

- Even if we assume that the factor base is defined by a curve, to establish that it contains about q elements, one should establish a bound on the genus of the curve, but this is not done.

- In subsection 3.6 the average number of "decompositions" generated by solving one system is discussed. There is however an important shortcoming: In the algorithm only systems which give rise to zero-dimensional solution sets are solved. Therefore, when computing the average number of decompositions, solutions in systems with higher-dimensional solution sets should be discarded. It is implicitly assumed that the number of solutions of the systems with higher-dimensional solution sets is negligible, but this is not established.

- The algorithm uses a double large prime variation. However, it is not at all discussed why this leads to the desired expected running time. *Heuristically*, this is the case, but one has to establish that the constructed graph of large prime relations contains (with sufficiently high probability) sufficiently many points and has a sufficiently small diameter.

- Even if all the previous points are addressed, the algorithm in [Gau04b] only gives the desired result if the group is cyclic.

We also make some comments on the relationship between Theorem 4 and our heuristic result presented at ECC 2004: For the result at ECC 2004 we enlarged the factor base – it is not anymore the set of rational points of a 1-dimensional subscheme but of a higher dimensional subscheme. We have tried hard to prove that with a sufficiently high probability the generated systems of polynomial equations have a zero-dimensional solution set. However, we can only prove this important statement if we go back to 1-dimensional subschemes and to rather small extension degrees.

An outline

Let us give an outline of the rest of this section:

In the next subsection, we give the algorithms for the two theorems stated above. For this we start off with an overview of the "decomposition algorithm". Then we give the two algorithms. The subalgorithms for the computation of a suitable covering φ are given in the following subsection. In subsection 3.5.4 we review some well known facts on multihomogeneous polynomials. We in particular discuss intersection theory in $(\mathbb{P}^1_k)^n$ and resultants, including computational aspects. Then we introduce the homogeneous summation polynomials via an abstract approach. Finally, the last subsection contains the analysis of the algorithm.

An interesting aspect about the algorithm and its analysis is that the algorithm can be stated with substantially less theoretical background than the analysis, and our exposition reflects this fact. In particular, it is not necessary to speak about Weil restriction when describing the algorithm, and consequently we only introduce the Weil restriction at the beginning of the analysis. One might however argue that the Weil restrictions are implicitly present in the definition of the factor base and the construction of the multivariate system in the decomposition algorithm.

3.5.2 The algorithms

In this subsection we outline the algorithms for theorems 4 and 5.

As already mentioned, the relation generation procedure of both algorithms relies on a *decomposition algorithm*. Before we come to the two algorithms, we give an overview over this procedure. All computational results in this subsection are over finite fields and on a bit-RAM.

3.5.2.1 The decomposition algorithm

The decomposition algorithm relies on "homogeneous summation polynomials". These polynomials can be obtained by homogenizing the summation polynomials introduced by Semaev in [Sem04] in an appropriate way. A more systematic point of view is however to regard Semaev's summation polynomials as being obtained by dehomogenization of the homogeneous summation polynomials. The homogeneous summation polynomials are studied in detail in subsection 3.5.5; here we merely merely mention the key results which are needed to describe the decomposition algorithm.

Let us fix some notation and terminology.

Notation and Terminology 3.34 We identify $(\mathbb{P}^1)^n$ componentwise with $\mathrm{Proj}(\mathbb{Z}[\mathrm{X}_1, \mathrm{Y}_1]) \times \cdots \times \mathrm{Proj}(\mathbb{Z}[\mathrm{X}_n, \mathrm{Y}_n])$. Therefore we have bases $\mathrm{X}_i, \mathrm{Y}_i \in$

$\Gamma((\mathbb{P}^1)^n, \mathcal{O}(0,\ldots,0,1,0,\ldots,0))$, where the 1 is at the i^{th} position. For any commutative ring A we have the multigraded homogeneous coordinate ring $A[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ of $(\mathbb{P}^1_A)^n$. In the following by a *multihomogeneous* polynomial in $A[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ we mean a polynomial which is homogeneous with respect to the multigrading. A *multihomogeneous* ideal in $A[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ is then an ideal in $A[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ which is generated by multihomogeneous polynomials. Now for some multihomogeneous ideal I, we denote the *subscheme* defined by I in $(\mathbb{P}^1_k)^n$ by $V(I)$. Moreover, we set $\mathrm{x}_i := \frac{\mathrm{X}_i}{\mathrm{Y}_i}$ and $\mathbb{A}^n := \mathrm{Spec}(\mathbb{Z}[\mathrm{x}_1, \ldots, \mathrm{x}_n])$.

In subsection 3.5.5 we show the following two propositions.

Proposition 3.35 *Let E be an elliptic curve over a field k, and let us fix a covering $\varphi : E \longrightarrow \mathbb{P}^1_k$ of degree 2 with $\varphi \circ [-1] = \varphi$. Let $m \in \mathbb{N}$ with $m \geq 2$. Then there exists an up to multiplication by a non-trivial constant unique irreducible multihomogeneous polynomial $S_{\varphi,m} \in k[\mathrm{X}_1, \mathrm{Y}_1, \mathrm{X}_2, \mathrm{Y}_2, \ldots, \mathrm{X}_m, \mathrm{Y}_m]$ such that for all $P_1, \ldots, P_m \in E(\overline{k})$ we have $S_{\varphi,m}(\varphi(P_1), \ldots, \varphi(P_m)) = 0 \longleftrightarrow \exists \epsilon_1, \ldots, \epsilon_m \in \{1, -1\}$ such that $\epsilon_1 P_1 + \cdots \epsilon_m P_m = O$. The polynomial $S_{\varphi,m}$ has multidegree $(2^{m-2}, \ldots, 2^{m-2})$.*

Definition 3.36 We call a *multihomogeneous* polynomial $S_{\varphi,m}$ as in the proposition an m^{th} *summation polynomial of E with respect to φ.*

Proposition 3.37 *Given an elliptic curve in Weierstraß form over a finite field $\mathbb{F}_q$ $m \in \mathbb{N}$ with $m \geq 2$ and $\varphi : E \longrightarrow \mathbb{P}^1_k$ of degree 2 with $\varphi \circ [-1] = \varphi$, the m^{th} summation polynomial with respect to the covering $\varphi : E \longrightarrow \mathbb{P}^1_{\mathbb{F}_q}$ can be computed with a randomized algorithm in an expected time of $\mathcal{P}oly(e^{m^2} \cdot \log(q))$.*

Now let $K|k$ be a field extension of degree n with basis $b_1, \ldots, b_n$, let E be an elliptic curve over K (rather than over k!), and let $\varphi : E \longrightarrow \mathbb{P}^1_K$ be a covering of degree 2 with $\varphi \circ [-1] = \varphi$.

Now let $P \in E(K)$. Let $S_{\varphi,n+1}(\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n, \varphi(P))$ be a polynomial obtained by inserting the coordinates of $\varphi(P)$ for the variables X_{n+1}, Y_{n+1} in an $(n+1)^{\text{th}}$ summation polynomial of E with respect to φ; note that this polynomial is unique up to multiplication with a non-trivial constant.

Let $S^{(1)}, \ldots, S^{(n)} \in k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ be defined by

$$\sum_{j=1}^{n} b_j S^{(j)} = S_{\varphi,n+1}(\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n, \varphi(P)) \; . \tag{3.28}$$

Clearly, if $S^{(j)}$ is non-zero, just as $S_{\varphi,n+1}$ it is multigraded of multidegree $(2^{n-1}, \ldots, 2^{n-1})$. Note also that a different basis of $K|k$ would give rise

to a system of polynomials over k which generate the same k-vector space. The same holds if the summation polynomial is multiplied by a non-trivial constant or if the coordinates of $\varphi(P)$ are simultaneously multiplied by a non-trivial constant. In particular, the subscheme $V(S^{(1)}, \ldots, S^{(n)})$ of $(\mathbb{P}^1_k)^n$ does not depend on these choices.

For $Q_1, \ldots, Q_n \in \mathbb{P}^1(k)$, the following conditions are equivalent:

- There exist $P_1, \ldots, P_n \in E(\overline{K})$ such that $P_1 + \cdots + P_n = P$ and $x(P_i) = Q_i$ for all $i = 1, \ldots, n$.
- $S_{\varphi,n+1}(Q_1, \ldots, Q_n, \varphi(P)) = 0$.
- For all $j = 1, \ldots, n$, $S^{(j)}(Q_1, \ldots, Q_n) = 0$, that is, $(Q_1, \ldots, Q_n)$ is a k-rational point of $V(S^{(1)}, \ldots, S^{(n)})$.

By a "decomposition algorithm" we mean an algorithm for the following computational problem: Given a prime power q, $n \in \mathbb{N}$, an $\mathbb{F}_q$-basis $b_1, \ldots, b_n$ of $\mathbb{F}_{q^n}|\mathbb{F}_q$, an elliptic curve E over $\mathbb{F}_{q^n}$ (given by a Weierstraß model), $\varphi : E \longrightarrow \mathbb{P}^1_k$ as well as $P \in E(\mathbb{F}_{q^n})$, determine if the subscheme $V(S^{(1)}, \ldots, S^{(n)})$ of $(\mathbb{P}^1_{\mathbb{F}_q})^n$ defined by $S^{(1)}, \ldots, S^{(n)}$ is zero-dimensional, and if this is the case, determine all tuples $(P_1, \ldots, P_n) \in E(\mathbb{F}_{q^n})^n$ with $\varphi(P_i) \in \mathbb{P}^1(\mathbb{F}_q)$ for all $i = 1, \ldots, n$ and $P_1 + \cdots + P_n = P$!

In subsection 3.5.4 we show the following proposition (see subsubsection 3.5.4.1 and Proposition 3.65 in subsubsection 3.5.4.3).

Proposition 3.38

a) Let k be a field, and let $F_1, \ldots, F_n \in k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ be multi-graded polynomials of multidegree $(d, d, \ldots, d)$ for some $d \in \mathbb{N}$. If then $V(F_1, \ldots, F_n)$ is zero-dimensional, its degree is $n! \cdot d^n$ (that is, the number of solutions over $\overline{k}$ "with multiplicities" is $n! \cdot d^n$).

b) Given a system of multihomogeneous polynomials $F_1, \ldots, F_n \in \mathbb{F}_q[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ of multidegree $(d, d, \ldots, d)$ for some $d \in \mathbb{N}$, one can determine if the system has a zero-dimensional solution set and if this is the case compute all solutions over $\mathbb{F}_q$ in an expected time of $\mathcal{P}oly(n! \cdot d^n \cdot \log(q))$.

Based on the previously mentioned computational results, we have the following decomposition algorithm.

We have already remarked that one can compute the polynomial $S_{\varphi,n+1}$ in an expected time of $\mathcal{P}oly(e^{n^2} \cdot \log(q))$. Thus one can also determine the polynomials $S^{(1)}, \ldots, S^{(n)}$ in an expected time of $\mathcal{P}oly(e^{n^2} \cdot \log(q))$. By the previous proposition one can then determine if the subscheme

$V(S^{(1)},\dots,S^{(n)})$ of $(\mathbb{P}^1_{\mathbb{F}_q})^n$ is zero-dimensional and if this is the case compute all its $\mathbb{F}_q$-rational points in an expected time of $\mathcal{P}oly(n!\cdot 2^{n^2}\cdot\log(q)) = \mathcal{P}oly(e^{n^2}\cdot\log(q))$.

Assume now that the scheme is indeed zero-dimensional, and that all $\mathbb{F}_q$-rational points have been computed. We now want to find all tuples $(P_1,\dots,P_n)\in E(\mathbb{F}_{q^n})^n$ with $\varphi(P_i)\in\mathbb{P}^1(\mathbb{F}_q)$ for all $i=1,\dots,n$ and $P_1+\dots+P_n=P$.

For this we iterate over all $\mathbb{F}_q$-rational points of $V(S^{(1)},\dots,S^{(n)})$. For each $(Q_1,\dots,Q_n)\in V(S^{(1)},\dots,S^{(n)})(\mathbb{F}_q)$ we consider all possibles tuples $(P_1,\dots,P_n)\in E(\mathbb{F}_{q^n})^n$ with $x(P_i)=Q_i$ for $i=1,\dots,n$ and check if $P_1+\dots+P_n=P$. We output all tuples $(P_1,\dots,P_n)$ for which this is the case.

Now for each tuple $(P_1,\dots,P_n)\in V(S^{(1)},\dots,S^{(n)})(\mathbb{F}_q)$ we need $\tilde{\mathcal{O}}(2^n)\cdot\mathcal{P}oly(\log(q))$ bit operations, and we have $\mathcal{P}oly(e^{n^2})$ such tuples $(P_1,\dots,P_n)$. The expected total running time is then still in $\mathcal{P}oly(e^{n^2}\cdot\log(q))$.

We obtain:

Proposition 3.39 *Given q, $n\in\mathbb{N}$, E, φ and P as above, one can determine if the subscheme $V(S^{(1)},\dots,S^{(n)})$ of $(\mathbb{P}^1_{\mathbb{F}_q})^n$ is zero-dimensional, and if this is the case determine all tuples $(P_1,\dots,P_n)\in E(\mathbb{F}_{q^n})^n$ with $\varphi(P_i)\in\mathbb{P}^1(\mathbb{F}_q)$ for $i=1,\dots,n$ in an expected time of $\mathcal{P}oly(e^{n^2}\cdot\log(q))$.*

Terminology 3.40 We say that "the decomposition algorithm succeeds" if applied to an instance as described above if the scheme $V(F_1,\dots,F_n)$ is zero-dimensional and there exists a tuple $(P_1,\dots,P_n)\in E(\mathbb{F}_{q^n})^n$ with $\varphi(P_i)\in\mathbb{P}^1(\mathbb{F}_q)$ and $P_1+\dots+P_n=P$. Otherwise we say that "the decomposition algorithm fails".

In order to analyze the index calculus algorithms we need a lower bound in the probability that the decomposition algorithm succeeds. Let us mention the key result for the analysis of the algorithm for Theorem 4, which we prove in subsubsection 3.5.6.5 (Proposition 3.98):

Proposition 3.41 *Let $\epsilon>0$. Then for n large enough*[9] *and $(2+\epsilon)\cdot n^2\le\log_2(q)$ the following holds: Let $E/\mathbb{F}_{q^n}$ be an elliptic curve, and let $\varphi: E\longrightarrow\mathbb{P}^1_K$ be a covering of degree 2 with $\varphi\circ[-1]=\varphi$ such that the following condition holds:*

There exists a point $P\in\mathbb{P}^1(\overline{k})$ which is a ramification point of φ such that the points $P,\sigma(P),\dots,\sigma^{n-1}(P)$ are all distinct and φ is not ramified at $\sigma(P),\dots,\sigma^{n-1}(P)$.

Then the probability that the decomposition algorithm succeeds if applied to a uniformly randomly distributed element in $E(\mathbb{F}_{q^n})$ is $\ge q^{-\frac{1}{2}}$.

[9] As usual, by the phrase "for n large enough" we mean that there exists a constant $C>0$ such that the statement holds for $n\ge C$.

3.5.2.2 The algorithm for Theorem 4

Below we give an algorithm which leads to the following result.

Proposition 3.42 *Let $\epsilon > 0$. Then there exists a randomized algorithm such that the following holds: Given a prime power q, a natural number n with $(2+\epsilon) \cdot 2^n \leq \log_2(q)$, an elliptic curve over $\mathbb{F}_{q^n}$ (in Weierstraß form) and two points $A, B \in E(\mathbb{F}_{q^n})$ with $B \in \langle A \rangle$ as well as a generating system of $E(\mathbb{F}_{q^n})$ whose size is polynomially bounded in $\log(q^n)$, the algorithm outputs the discrete logarithm of B with respect to A. Moreover, the expected running time is polynomially bounded in q, and each LOAD and STORE operation operates in a time which is polynomially bounded in $\log(q)$.*

Let us see how one can with this proposition obtain Theorem 4:

First, Proposition 3.42 implies (see the proof of Proposition 3.18 from Proposition 3.16):

Proposition 3.43 *Let $\epsilon > 0$. Then there exists a randomized algorithm such that the following holds: Given a prime power q, a natural number n with $(2+\epsilon) \cdot n^2 \leq \log_2(q)$, an elliptic curve over $\mathbb{F}_{q^n}$ (in Weierstraß form) and two points $A, B \in E(\mathbb{F}_{q^n})$ with $B \in \langle A \rangle$ as well as a generating system of $E(\mathbb{F}_{q^n})$ whose size is polynomially bounded in $\log(q^n)$, the algorithm outputs the discrete logarithm of B with respect to A or fails. Moreover, the running time of the algorithm is polynomially bounded in q, and the probability of failure is $\leq \frac{1}{2}$.*

If we repeat the two steps

1. choose a "potential generating system" of size $\mathcal{P}oly(\log(q^n))$ with an algorithm satisfying Proposition 3.17
2. apply an algorithm satisfying Proposition 3.43

until we have obtained the discrete logarithm, we obtain:

Proposition 3.44 *Let $\epsilon > 0$. Then there exists a randomized algorithm such that the following holds: Given a prime power q, a natural number n with $(2+\epsilon) \cdot n^2 \leq \log_2(q)$, an elliptic curve over $\mathbb{F}_{q^n}$ (in Weierstraß form) and two points $A, B \in E(\mathbb{F}_{q^n})$ with $B \in \langle A \rangle$, the algorithm outputs the discrete logarithm of B in an expected time which is polynomially bounded in q.*

Note now that in Proposition 3.43 in particular the extension degree n is part of the input whereas in Theorem 4 this is not the case. To obtain

Theorem 4 we apply an algorithm for Proposition 3.43 with all possible extension degrees "in parallel".

We are now coming to the algorithm for Proposition 3.42. For this we outline – as already mentioned – an index calculus algorithm which follows the "general algorithm" in subsection 3.2.3. The input of the algorithm consists of q, n, E, A, B as described in Proposition 3.42 as well as a system of elements $C_1, \ldots, C_u$ in $E(\mathbb{F}_{q^n})$.

We give the subroutines together with an analysis of their complexity. The subroutines and thus also the complete algorithm apply to all instances without any bound on n, but they might not always terminate. Indeed, we prove the following statement:

Let $\epsilon > 0$. Then given a prime power q, a *large enough* natural number n with $(2+\epsilon)\cdot 2^n \leq \log_2(q)$, an elliptic curve over $\mathbb{F}_{q^n}$ (in Weierstraß form) and two points $A, B \in E(\mathbb{F}_{q^n})$ with $B \in \langle A \rangle$ as well as a generating system of $E(\mathbb{F}_{q^n})$ whose size is polynomially bounded in $\log(q^n)$, the algorithm outputs the discrete logarithm of B with respect to A. Moreover, the expected running time is then polynomially bounded in q.

Proposition 3.42 can then be obtained by applying this algorithm "in parallel" with a brute force computation.

a) Computation of the group order

We use Schoof's algorithm to compute the L-polynomial ([Sch85]), which operates in a time which is polynomially bounded in $\log(q^n)$.

c) Construction of the factor base

We have already mentioned that the factor base is a set

$$\{P \in E(\mathbb{F}_{q^n}) \mid \varphi(P) \in \mathbb{F}_q\} .$$

for a suitable covering of degree 2 $\varphi : E \longrightarrow \mathbb{P}^1_{\mathbb{F}_{q^n}}$ with $\varphi \circ [-1] = \varphi$.

Let σ be the relative Frobenius automorphism of $\overline{\mathbb{F}}_q|\mathbb{F}_q$. Now the condition on φ we impose is the following condition, already mentioned in Proposition 3.41.

Condition 3.45 There exists a point $P \in \mathbb{P}^1(\overline{\mathbb{F}}_q)$ which is a ramification point of φ such that the points $P, \sigma(P), \ldots, \sigma^{n-1}(P)$ are all distinct and φ is not ramified at $\sigma(P), \ldots, \sigma^{n-1}(P)$.

This condition might seem strange for the moment. The reasons for this condition will be discussed in subsubsection 3.5.6.3. Very briefly, the factor

base is in a certain sense defined by a 1-dimensional $\mathbb{F}_q$-scheme, and the condition ensures that this scheme is birational to a curve over $\mathbb{F}_q$.

In subsection 3.5.3 we prove:

Proposition 3.46 *Given a prime power q, $n \in \mathbb{N}$ and an elliptic curve over $\mathbb{F}_{q^n}$ in Weierstraß form such that $(q,n) \neq (3,2)$ one can compute a covering $\varphi : E \longrightarrow \mathbb{P}^1_{\mathbb{F}_{q^n}}$ of degree 2 with $\varphi \circ [-1] = \varphi$ satisfying Condition 3.45 in an expected time which is polynomially bounded in $n \cdot \log(q)$.*

Thus clearly the factor base can be constructed in an expected time which is polynomially bounded in $\log(q^n)$.

d) Relation generation

Let $C_1, \ldots, C_u$ be the system of elements in $E(\mathbb{F}_{q^n})$ which is part of the input, and let $C \in E(\mathbb{F}_{q^n})$ be given.

We choose $s_1, \ldots, s_u \in \{0, \ldots, \#E(\mathbb{F}_{q^n}) - 1\}$ uniformly at random and compute $\sum_j s_j C_j + C$. Then we apply the decomposition algorithm as described in the previous subsubsection to this element and the covering φ. If the procedure does not fail, we have obtained at least one relation between factor base elements, $C_1, \ldots, C_u$ and the input elements A, B. We output such a relation. (It does not matter which one we output as long as the distribution of the output only depends on the element $\sum_j s_j C_j + C$ (and not on the further internal state of the algorithm).) We repeat this procedure until we have obtained such a relation.

In our applications of the algorithm, u is polynomially bounded in $\log(q^n)$. Then the time to compute $\sum_j s_j C_j + C$ is polynomial in $\log(q^n)$. By Proposition 3.39, the expected running time of one iteration is then in $\mathcal{P}oly(e^{n^2} \cdot \log(q))$. Moreover, by Proposition 3.41 (Proposition 3.98) for instances with $(2+\epsilon) \cdot n^2 \leq \log_2(q)$ and n large enough the expected number of iterations is in $\mathcal{O}(q^{1/2})$.

We conclude that for instances with $(2+\epsilon) \cdot n^2 \leq \log_2(q)$ and n large enough and for u polynomially bounded in $\log(q^n)$, the expected running time of the relation generation procedure is in $\mathcal{P}oly(e^{n^2} \cdot \log(q)) \cdot \mathcal{O}(q^{1/2}) \subseteq \mathcal{P}oly(q)$.

The overall running time

We again restrict ourselves to instances with $(2+\epsilon) \cdot n^2 \leq \log_2(q)$ and u polynomially bounded in $\log(q^n)$. As the factor base has a size of $\mathcal{O}(q)$, it is now clear that for n large enough the expected running time of the whole algorithm is then polynomially bounded in q.

3.5.2.3 The algorithm for Theorem 5

It order to obtain Theorem 5 we modify the previous algorithm with a double large prime variation.

Let us *fix* n and consider the discrete logarithm problem in elliptic curves over finite fields $\mathbb{F}_{q^n}$. Then we follow exactly the algorithm in subsubsection 3.3.2, with the following obvious modifications: Let φ be a covering as above. Then the factor base is a subset of size $\lceil q^{1-\frac{1}{n}} \rceil$ of $\{P \in E(\mathbb{F}_{q^n}) \mid \varphi(P) \in \mathbb{F}_q\}$ which is invariant under application of $[-1]$, for relation generation we rely on the "decomposition algorithm", and generally the genus g is substituted by the extension degree n. An important observation is (cf. Proposition 3.39): *For fixed extension degree n there is a decomposition algorithm whose expected running time is polynomially bounded in* $\log(q)$.

The analysis is also similar to the analysis of the algorithm in subsection 3.3.2. It relies on the following proposition.

Proposition 3.47 *Let n be fixed. Then again for elliptic curves $E/\mathbb{F}_{q^n}$ and coverings φ such that Condition 3.45 holds the following is true:*

a) $\#\{P \in E(\mathbb{F}_{q^n}) \mid \varphi(P) \in \mathbb{P}^1(\mathbb{F}_q)\} \sim q$.

b) There exist constants $C, D > 0$ such that the following holds: Let M be any subset of $\{(P_1, \ldots, P_n) \in E(K)^n \mid \varphi(P_i) \in \mathbb{P}^1(k)$ for all $i = 1, \ldots, n\}$. Then the number of elements $P \in E(K)$ such that the decomposition algorithm succeeds and there exist $P_1, \ldots, P_n \in M$ with $P_1 + \cdots + P_n = \pm P$ is

$$\geq D \cdot \#M - C \cdot q^{n-1} \, .$$

For a) we refer to Remark 3.79 in subsubsection 3.5.6.3. Part b) is a summary of Proposition 3.96 in subsubsection 3.5.6.5 for fixed n.

The subroutines a) and b) are exactly as above. So let us describe subroutines c) and d) in greater detail.

c) Construction of the factor base and the graph of large prime relations

First we again construct a covering $\varphi : E \longrightarrow \mathbb{P}^1_{\mathbb{F}_{q^n}}$ of degree 2 with $\varphi \circ [-1] = \varphi$ satisfying Condition 3.45. As indicated above, this can be done in an expected time which is polynomially bounded in $\log(q^n)$. We enumerate this set and define the factor base $\mathcal{F}$ as any subset $\mathcal{F}$ of cardinality $\lceil q^{1-\frac{1}{n}} \rceil$ or $\lceil q^{1-\frac{1}{n}} \rceil + 1$ of $\{P \in E(\mathbb{F}_{q^n}) \mid \varphi(P) \in \mathbb{F}_q\}$ which is invariant under application of $[-1]$. The set of large primes is then $\mathcal{L} := \{P \in E(\mathbb{F}_{q^n}) \mid \varphi(P) \in \mathbb{F}_q \, , \, P \notin \mathcal{F}\}$.

Clearly these operations can be performed in an expected time of $\tilde{O}(q)$.

Now as already mentioned the construction of the tree of large prime relations is performed as in subsection 3.3.2. A minor difference is that we do not anymore have unique factorization, and we check for each output of the decomposition algorithm if it leads to an FP or a PP relation. The procedure is therefore as follows:

Procedure: Construction of the tree of large prime relations

Construct a tree T with vertex set contained in $\mathcal{L} \dot\cup \{*\}$ as follows:
Let T consist only of the root $*$.
Let $N_{\max} \longleftarrow \lceil q^{1-1/n+1/n^2} \rceil$
Let $s \longleftarrow 1$.
Repeat
  Repeat
    Choose $s_1, \ldots, s_u \in \mathbb{Z}/N\mathbb{Z}$ uniformly and independently at random.
    Apply the decomposition algorithm to $\sum_j s_j C_j$.
    If the decomposition algorithm succeeds,
      check for every tuple $(P_1, \ldots, P_n)$ with $\varphi(P_i) \in \mathbb{P}^1(\mathbb{F}_q)$ and $\sum_j P_j = \sum_j s_j C_j$ if it leads to a FP or a PP relation, that is, to a relation of the form $\sum_j r_j F_j + c_P P + c_Q Q = \sum_j s_j C_j$ where $c_P > 0, c_Q > 0$ and $P \in \mathcal{F} \cup T_{s-1}, Q \in \mathcal{L} - (\mathcal{F} \cup T)$.
      If this is the case,
        fix such a relation.
        if $P \in \mathcal{F}$ (i.e. if we have an FP relation),
          insert Q and an edge from $*$ to Q into T
        if $P \in T_{s-1}$ (i.e. if we have a PP relation),
          insert Q and an edge from P to Q into T.
        In both cases label Q with s and the edge with $(r_j)_j$ (in sparse representation).
    Until T contains $2^{s-1} \cdot \lceil q^{1-1/n} \rceil$ edges or the number of edges equals $N_{\max}$.
    If the number of edges equals $N_{\max}$, STOP.
    Let $s \longleftarrow s + 1$.

Here as in subsection 3.3.2 T_s is the subtree of T consisting of vertices with label $\leq s$, that is, the tree which is constructed until including stage s.

With part b) of Proposition 3.47 the analysis in subsection 3.3.3 carries also easily over to the present setting. Let us first consider Stage 1 of the algorithm, that is, $s = 1$.

We set

$$M := \mathcal{F}^{n-1} \times (\mathcal{L} - T) .$$

For q large enough (independently of T) this set has cardinality $\geq (q^{1-\frac{1}{n}})^{n-1} \cdot \frac{q}{4}$. Therefore the number of elements in $E(\mathbb{F}_{q^n})$ for which we obtain a new edge is $\geq \frac{D}{4} \cdot (q^{1-\frac{1}{n}})^{n-1} \cdot q - C \cdot q^{n-1} = \frac{D}{4} \cdot q^{n-1+\frac{1}{n}} - Cq^{n-1}$. For q large enough this is

$$\geq \frac{D}{8} \cdot q^{n-1+\frac{1}{n}} ,$$

and the probability to obtain an FP relation is therefore

$$\geq \frac{D}{16} \cdot q^{-(1-\frac{1}{n})} .$$

We thus obtain: A tree of large prime relations of size $\lceil q^{1-\frac{1}{n}+\frac{1}{n^2}} \rceil$ is constructed in an expected time of $\tilde{O}(q^{2-\frac{2}{n}})$.

Let us now assume that we are in Stage s with $s \geq 2$. We set

$$M := \mathcal{F}^{n-2} \times (\mathcal{F} \times T_{s-1}) \times (\mathcal{L} - T) .$$

Now for q large enough (and independently of T, in particular independently of s) this set has cardinality $\geq (q^{1-\frac{1}{n}})^{n-2} \cdot 2^{s-2} \cdot q^{g-1+\frac{1}{g}} \cdot \frac{q}{4} = 2^{s-2} \cdot (q^{1-\frac{1}{n}})^{n-1} \cdot \frac{q}{4}$. For q large enough the number of elements in $E(\mathbb{F}_{q^n})$ for which we obtain a new edge is

$$\geq \frac{D}{16} \cdot 2^{s-2} \cdot q^{-(1-\frac{1}{n})} .$$

This implies that for q large enough (independently of s) the following holds: Given any tree T_{s-1} with $2^{s-2} \cdot \lceil q^{1-\frac{1}{g}} \rceil$ edges, the expected number of tries until a tree T with $\min\{2^{s-1} \cdot \lceil q^{1-\frac{1}{g}} \rceil, N_{\max}\}$ edges is constructed is

$$\leq \frac{32}{D} \cdot q^{-(1-\frac{1}{n})} .$$

This completes the analysis.

d) Relation generation

The relation generation is also as in subsection 3.3.2, again with the obvious difference that we use the decomposition algorithm. Again by item b) of Proposition 3.47 it is obvious that the expected running time is in $\tilde{O}(q^{2-\frac{2}{n}})$.

The overall running time and conclusion

Altogether we obtain an expected running time of $\tilde{O}(q^{2-\frac{2}{n}})$, and we have storage requirements of $\tilde{O}(q^{1-\frac{1}{n}+\frac{1}{n^2}})$.

3.5.3 Computing a suitable covering

We discuss how a covering $\varphi : E \longrightarrow \mathbb{P}^1_{\mathbb{F}_{q^n}}$ as required in the construction of the factor base can be computed.

We make some case distinctions. In each case we start off with a specific Weierstraß model and determine some automorphism α of $\mathbb{P}^1_{\mathbb{F}_{q^n}}$. Then we set $\varphi := \alpha \circ x_{|E}$.

3.5.3.1 Even characteristic

Let first $j(E) = 0$. Then E is the projective curve defined by the homogenization of some polynomial of the form

$$y^2 + a_3 y + x^3 + a_4 x + a_6 \ .$$

(see [Sil86, Appendix A]) (with $a_3 \neq 0$). (In similar situations we say from now on that E is "defined" by a polynomial for shortness.) Now $x_{|E}$ is ramified exactly over ∞. We set $\alpha := \frac{ax-1}{\mathrm{x}}$ for some $a \in \mathbb{F}_{q^n}$ which is not contained in any proper subfield of $\mathbb{F}_{q^n}|\mathbb{F}_q$.[10] Then α maps ∞ to a, and thus φ is ramified exactly at a. Clearly the condition is satisfied.

Let now $j(E) \neq 0$. Then E is the projective curve defined by the polynomial

$$y^2 + xy + x^3 + a_2 x^2 + a_6 \ .$$

Then $x_{|E}$ is ramified exactly over 0 and ∞. We set $\alpha := \mathrm{x} + a$ with a as above. Then φ is ramified at a and ∞, and again the condition is satisfied.

3.5.3.2 Odd characteristic

Let E be defined by

$$y^2 - f(x) \ ,$$

where $f(x) \in \mathbb{F}_q[x]$ is monic of degree 3. The conditions which have to be satisfied are now more subtle but the algorithm is very simple:

We choose $\lambda \in \mathbb{F}_{q^n}$ uniformly at random and with $\alpha := \mathrm{x} - \lambda$ we check if the condition is satisfied. We repeat this until the condition is satisfied.

Note here that if $f(\mathrm{x}) = (\mathrm{x} - \lambda_1)(\mathrm{x} - \lambda_2)(\mathrm{x} - \lambda_3)$ (with $\lambda_i \in \mathbb{F}_{q^{6n}}$), then the ramification points of $\varphi = \alpha \circ x_{|E}$ in $\mathbb{P}^1(\overline{\mathbb{F}}_q)$ are $\lambda_i - \lambda$ for $i = 1, 2, 3$. So it is easy to check the condition.

Proposition 3.46 now follows from the following lemma. (Note that we only apply the lemma in the case that q is odd.)

[10] By a "proper subfield" we mean here a subfield of a field extension $K|k$ which is not equal to K.

Lemma 3.48 *There exists a constant $C \in (0,1)$ such that the following holds:*

Let q be a prime power and n a natural number such that $(q,n) \neq \{(2,2),(3,2),(2,3),(2,4)\}$. Now let $\lambda_1, \lambda_2, \lambda_3 \in \overline{\mathbb{F}}_q$, and let λ be a uniformly distributed element in $\mathbb{F}_{q^n}$. Then with a probability $\geq C$ we have

$$(\lambda_1 - \lambda)^{q^i} \notin \{\lambda_1 - \lambda, \lambda_2 - \lambda, \lambda_3 - \lambda\}$$

for $i = 1, \ldots, n-1$.

Proof. Let $\ell = 1,2,3$. We have $(\lambda_1 - \lambda)^{q^i} = \lambda_\ell - \lambda$ if and only if $\lambda^{q^i} - \lambda = \lambda_1^{q^i} - \lambda_\ell$. The map $\lambda \mapsto \lambda^{q^i} - \lambda$ is an $\mathbb{F}_q$-linear map with kernel $\mathbb{F}_{q^{\gcd(i,n)}}$. There are thus either no or $q^{\gcd(i,n)}$ such λ.

We obtain: In total there are at most $3\sum_{i=1}^{n-1} q^{\gcd(i,n)}$ elements λ for which the condition in the lemma is not satisfied.

Now $3\sum_{i=1}^{n-1} q^{\gcd(i,n)} \leq 3(n-1) \cdot q^{n/2}$, and therefore the probability in question is

$$\geq 1 - \frac{3(n-1)}{q^{n/2}} \geq 1 - \frac{3(n-1)}{2^{n/2}} \, .$$

For $n \geq 10$ this is $\geq \frac{5}{32} > 0$.

One also easily sees that for $n \leq 9$ and $(q,n) \neq \{(2,2),(3,2),(2,3),(2,4)\}$ the probability is positive. □

3.5.4 Some results on multihomogeneous polynomials

In this subsection we are concerned with results related to multihomogeneous polynomials and systems of such polynomials. In particular, we give some information on aspects of intersection theory in the special case of $(\mathbb{P}^1_k)^n$, including multigraded resultants, and we discuss computational aspects. Here and in this whole subsection, k is a field.

3.5.4.1 Intersection theory in $(\mathbb{P}^1_k)^n$

In this subsubsection we review some standard material on intersection theory in the special case of $(\mathbb{P}^1_k)^n$.

Lemma 3.49 *Let X be a closed subscheme of $(\mathbb{P}^1_k)^n$ of dimension at least 1, and let $F \in k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ be a multihomogeneous polynomial whose multidegree is componentwise positive. Then $V(F)$ and X intersect non-trivially.*

Proof. The invertible sheaf $\mathcal{O}(\underline{d})$ is very ample, and under the corresponding embedding into projective space F corresponds to a non-trivial linear form. The result thus follows from intersection theory in projective space. □

Definition 3.50 We define the dimension of the empty scheme as -1.

Lemma 3.51 *Let $k \leq n+1$. Let $F_1, \ldots, F_k$ be multihomogeneous polynomials in $k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ such that all multidegrees are componentwise positive. Then $\dim(V(F_1, \ldots, F_k)) \geq n - k$. Moreover, we have equality if and only if for $\ell = 2, \ldots, k$ no irreducibility component of $V(F_1, \ldots, F_{\ell-1})$ is contained in $V(F_\ell)$.*

Proof. Let first $k \geq n$ (such that the first statement is non-trivial). Then by the previous lemma $V(F_1, \ldots, F_k)$ is non-empty. The first statement thus follows with Krull's Hauptidealsatz. The second statement also follows easily with the previous lemma and Krull's Hauptidealsatz. □

Notation 3.52 Let V be a fixed quasi-projective variety, and let X be a closed subscheme of V. Then we denote the class of V in the Chow ring of V by $[X]$. (We do not fix a notation for the cycle corresponding to a closed subscheme as we never perform operations with cycles but only with classes.)

Remark 3.53 Let X be a closed subscheme of $(\mathbb{P}^1_k)^n$ and let $F \in k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ be a multihomogeneous polynomial such that no irreducibility component of X is contained in $V(F)$. Then

$$[X \cap V(F)] = [X] \cdot [V(F)] \,,$$

where $X \cap V(F)$ is the scheme-theoretic intersection. Indeed, this is a special case of Axiom A7 on intersection theory in [Har77, Appendix A]. In particular, if $F_1, \ldots, F_k$ are multihomogeneous polynomials such that for all $\ell = 2, \ldots, k$ no irreducibility component of $V(F_1, \ldots, F_{\ell-1})$ is contained in $V(F_\ell)$, then

$$[V(F_1, \ldots, F_k)] = [V(F_1)] \cdots [V(F_k)] \,.$$

Note that by the previous lemma this is in particular the case if the multidegrees of the polynomials are componentwise positive and $\dim(V(F_1, \ldots, F_k)) = n - k$.

We have the following explicit description of the Chow ring of $(\mathbb{P}^1_k)^n$:

Proposition 3.54 *Let h_i be the class of $V(X_i) \subseteq (\mathbb{P}^1_k)^n$ for $i = 1, \ldots, n$. Then the Chow ring of $(\mathbb{P}^1_k)^n$ is generated by $h_1, \ldots, h_n$, and we have an isomorphism $\mathbb{Z}[H_1, \ldots, H_n]/(H_1^2, \ldots, H_n^2) \longrightarrow \mathrm{CH}((\mathbb{P}^1_k)^n)$, $[H_i] \mapsto h_i$.*

This proposition can easily be derived from a general result on the Chow rings of toric varieties (cf. the proposition on page 106 of [Ful93, Section 5.2]). We remark here that the book [Ful93] is concerned with toric varieties over

the complex number. However, analytic arguments play a minor role in the exposition, and the few such arguments can rather easily be replaced with algebraic arguments. In particular, the proposition just mentioned holds over arbitrary fields.

Example 3.55 The class of an effective Cartier divisor on $(\mathbb{P}^1_k)^n$ of multidegree $(d_1, \ldots, d_n)$ is $d_1h_1 + \cdots + d_nh_n$.

Let us consider the pull-back and push-forward homomorphisms associated with the canonical projections between products of $\mathbb{P}^1_k$'s. The following considerations follow immediately from the axioms of intersection theory in [Har77, Appendix A].

Let for $n_1 > n_2$ $p : (\mathbb{P}^1_k)^{n_1} \longrightarrow (\mathbb{P}^1_k)^{n_2}$ be the projection to the first n_2 components. Let us denote by h_i for $i = 1, \ldots, n_2$ the class of $V(X_i)$ in any of the two Chow groups.

Then the pull-back $p^* : \mathrm{CH}((\mathbb{P}^1_k)^{n_2}) \longrightarrow \mathrm{CH}(\mathbb{P}^1_k)^{n_1})$, which is a ring homomorphism, is given by the homomorphism which corresponds to the obvious inclusion under the isomorphism in Proposition 3.54. This means that it is given by $p^*(h_i) = h_i$.

The push-forward, which is a group homomorphism, is given as follows:

Lemma 3.56 *Let $\underline{e} \in \{0,1\}^{n_1}$. Then $p_*(h_1^{e_1} \cdots h_{n_1}^{e_{n_1}}) = 1$ if $e_{n_2+1} = \cdots = e_{n_1} = 1$ and $p_*(h_1^{e_1} \cdots h_{n_1}^{e_{n_1}}) = 0$ otherwise.*

Let now $F_1, \ldots, F_n$ be multihomogeneous polynomials whose multidegree is componentwise positive. Let the multidegree of F_i be $(d_{i,1}, \ldots, d_{i,n})$, and let $D := ((d_{i,j}))_{i,j}$. If now the scheme $V(F_1, \ldots, F_n)$ is zero-dimensional we conclude with the above proposition and Remark 3.53 that the class of $V(F_1, \ldots, F_n)$ in the Chow group is a cycle of degree $\mathrm{Perm}(D)$, the permanent of D. With other words: If the scheme $V(F_1, \ldots, F_n)$ is zero-dimensional, it has degree $\mathrm{Perm}(D)$. In particular, if additionally the multidegree of each F_i is $(d, \ldots, d)$ for a common $d \in \mathbb{N}$, then the degree of $V(F_1, \ldots, F_n)$ is $n! \cdot d^n$.

3.5.4.2 Multigraded resultants

We will make repeated use of resultants for systems of multihomogeneous polynomials in $k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$. Let us recall the definition and basic properties:

Let us fix some $n \in \mathbb{N}$. Let for $\underline{d} \in \mathbb{N}$ $M_{\underline{d}}$ be the set of monomials of multidegree $\underline{d}$ in $k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$.

Let for each $i = 1, \ldots, n+1$ some $\underline{d}^{(i)} \in \mathbb{N}^n$ be given. (Note that all coefficients are positive). We want to define the generic resultant for

multihomogeneous polynomials of multidegrees $\underline{d}^{(1)}, \ldots, \underline{d}^{(n+1)}$. For this we consider a "universal coefficient ring", which is a multivariate polynomial ring over the integers which for each pair (i, m) with $m \in M_{\underline{d}^{(i)}}$ has one indeterminate $c_{i,m}$, that is, it is the ring $\mathbb{Z}[\{c_{i,m}\}_{i=1,\ldots,n+1, m \in M_{\underline{d}^{(i)}}}]$. We define the *generic system of $n+1$ multihomogeneous polynomials with multidegrees* $\underline{d}^{(1)}, \ldots, \underline{d}^{(n+1)}$ as $G_1, \ldots, G_{n+1} \in \mathbb{Z}[\{c_{(i,m)}\}_{i,m}][\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ with $G_i = \sum_{m \in M_{\underline{d}^{(i)}}} c_{i,m}\, m$.

The generic resultant under consideration is then an element of $\mathbb{Z}[\{c_{i,m}\}_{i,m}]$, and the resultant of a particular system of multihomogeneous polynomials is obtained by substituting the coefficients of the polynomials for the generic coefficients.

Proposition 3.57

a) There is an irreducible polynomial $\mathrm{Res} \in \mathbb{Z}[\{c_{i,m}\}_{i=1,\ldots,n+1, m \in M_{\underline{d}^{(i)}}}]$ *which for* $i = 1, \ldots, n+1$ *is homogeneous in the coefficients the* i^{th} *generic polynomial and which has the following property: For all fields* k *and all systems of multihomogeneous polynomials* $F_1, \ldots, F_{n+1} \in k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$, *where* F_i *has multidegree* $\underline{d}^{(i)}$, *we have* $\mathrm{Res}(F_1, \ldots, F_{n+1}) = 0$ *if and only if* $V(F_1, \ldots, F_{n+1})$ *is non-empty. Here* $\mathrm{Res}(F_1, \ldots, F_{n+1})$ *is obtained by substituting the coefficients of the polynomials for the generic coefficients.*

b) The polynomial Res *with the above properties unique up to sign.*

c) For every field k, *the induced polynomial in* $k[\{c_{i,m}\}_{i,m}]$ *is irreducible.*

d) For each $i = 1, \ldots, n+1$, Res *has degree* $\mathrm{Perm}(D_i)$ *in the coefficients of the* i^{th} *generic polynomial, where* D_i *is obtained from the matrix* $\begin{pmatrix} \underline{d}^{(1)} \\ \vdots \\ \underline{d}^{(n+1)} \end{pmatrix}$ *by deleting the* i^{th} *row.*

Sketch of a proof. A corresponding result over the complex numbers follows from the general results in [GKZ94]. (The polynomial Res is then unique up to multiplication by a non-trivial complex number.) Even though there are various other works on general resultants, we could not find this "universal" result in the literature. We explain now how it can be derived from the results on "mixed resultants" in [GKZ94, Section 3.3].

For every commutative ring R, the set $\mathrm{Spec}(\mathbb{Z}[\{c_{i,m}\}_{i,m}])(R) \simeq \prod_{i,m} R$ corresponds in an obvious way to the set of systems $F_1, \ldots, F_{n+1} \in R[X_1, Y_1, \ldots, X_n, Y_n]$, where F_i has multidegree $\underline{d}^{(i)}$. For such a system of polynomials over a field such that at least one polynomial is non-trivial,

we denote by $\overline{(F_1, \ldots, F_{n+1})}$ the class of the coefficient vectors in projective space $\mathrm{Proj}(\mathbb{Z}[\{c_{i,m}\}_{i,m}])(k)$.

Let $p : (\mathbb{P}^1)^n \times_{\mathbb{Z}} \mathrm{Proj}(\mathbb{Z}[\{c_{i,m}\}_{i,m}]) \longrightarrow \mathrm{Proj}(\mathbb{Z}[\{c_{i,m}\}_{i,m}])$ be the projection to the second component. Let $V(G_1, \ldots, G_{n+1})$ be the closed subscheme of $(\mathbb{P}^1)^n \times_{\mathbb{Z}} \mathrm{Proj}(\mathbb{Z}[\{c_{i,m}\}_{i,m}])$ defined by the generic multihomogeneous polynomials $G_1, \ldots, G_{n+1}$ introduced above.

We consider $p(V(G_1, \ldots, G_{n+1}))$, which is a closed subscheme of $\mathrm{Proj}(\mathbb{Z}[\{c_{i,m}\}_{i,m}])$, with the induced reduced structure. Note that for a system $F_1, \ldots, F_{n+1}$ of polynomials as above over a field k, the fiber of $V(G_1, \ldots, G_{n+1})$ above $\overline{(F_1, \ldots, F_{n+1})}$ is $V(F_1, \ldots, F_{n+1})$, and thus the fiber of $p(V(G_1, \ldots, G_{n+1}))$ at $\overline{(F_1, \ldots, F_{n+1})}$ is set-theoretically equal to $p_k(V(F_1, \ldots, F_{n+1}))$. In particular, $\overline{(F_1, \ldots, F_{n+1})}$ is contained in $p_k(V(G_1, \ldots, G_{n+1})_k)$ if and only if $V(F_1, \ldots, F_{n+1})$ is non-empty.

Now the results in [GKZ94] immediately generalize to aribitrary algebraically closed fields, and therefore $V(G_1, \ldots, G_{n+1})_{\overline{\mathbb{Q}}}$ as well as $V(G_1, \ldots, G_{n+1}))_{\overline{\mathbb{F}}_p}$ for every prime number p are irreducible of codimension 1. It follows that $V(G_1, \ldots, G_{n+1})$ is irreducible of codimension 1. As $\mathrm{Proj}(\mathbb{Z}[\{c_{i,m}\}_{i,m}])$ is regular, this implies that it is a Cartier divisor and thus given by a section of an invertible sheaf on $\mathrm{Proj}(\mathbb{Z}[\{c_{i,m}\}])$. But every invertible sheaf on a projective space over $\mathbb{Z}$ is isomorphic to $\mathcal{O}(a)$ for some $a \in \mathbb{N}$ (cf. [Mum65, §0, 5 b)]). Therefore $V(G_1, \ldots, G_{n+1})$ is defined by a homogeneous polynomial in $\mathbb{Z}[\{c_{i,m}\}_{i,m}]$; let Res be such a polynomial.

We already know that Res is irreducible. Moreover, for every prime number p, the residue class $[\mathrm{Res}]_{\mathbb{F}_p[\{c_{i,m}\}_{i,m}]}$ is non-trivial, and thus the gcd of the coefficients of Res is 1. It is immediate that Res is homogeneous in the coefficients of each generic polynomial.

We have established a) and b). Result d) follows from the general results in [GKZ94] over the complex numbers.

It remains to prove c), where we can restrict ourself to algebraically closed fields. So let k be such a field. Let Res_k be the induced polynomial obtained from Res. We already know that Res_k is the product of a constant and a power of an irreducible polynomial. Now first, by d), Res_k is a polynomial of degree $\sum_i \mathrm{Perm}(D_i)$. Also, by applying the reasoning in [GKZ94] for k instead of $\mathbb{C}$, one obtains that $p_k(V(G_1, \ldots, G_{n+1})_k)$ with the reduced structure also has degree $\sum_i \mathrm{Perm}(D_i)$. As we already know that set-theoretically $p_k(V(G_1, \ldots, G_{n+1})_k)$ is defined by Res_k, we know now that this is also true scheme-theoretically. We conclude that Res_k is irreducible. □

We call the polynomial Res a *generic mixed multigraded resultant*. If the multidegrees of the generic polynomials are equal, we speak of a *multigraded resultant*. For some commutative ring R and multihomogeneous

$F_1, \ldots, F_{n+1} \in R[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ we call $\mathrm{Res}(F_1, \ldots, F_{n+1})$ (where Res is the generic mixed multihomogeneous resultant for polynomials of appropriate multidegree) the *multigraded resultant* of $F_1, \ldots, F_{n+1}$.

3.5.4.3 Computing resultants and solving systems

In this subsubsection we address computational problems related to multigraded resultants and the solution of zero-dimensional multihomogeneous polynomial systems. In particular, we give an algorithm to determine all solutions of a multihomogeneous polynomial system $F_1, \ldots, F_n$ over a field k, where each F_i has multidegree $(d, d, \ldots, d)$ for some $d \in \mathbb{N}$ and the scheme $V(F_1, \ldots, F_n)$ is zero-dimensional. We have not explicitly found our approach in the literature, but similar algorithms for related computational problems have been proposed.

Let $F_1, \ldots, F_{n+1} \in k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ be non-constant multihomogeneous polynomials of equal multidegree $(d, d, \ldots, d)$, and let $\mathrm{Res}(F_1, \ldots, F_{n+1})$ be the multigraded resultant of these polynomials. Now just as the usual Sylvester resultant, this resultant can be expressed as the determinant of a matrix each of whose entries is 0 or a coefficient of one of the polynomials F_i.

Let us to state this result fix the following definition.

Definition 3.58 Let M be a multigraded $k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$-module, and let $\underline{d} \in \mathbb{Z}^n$. Then the k-vector subspace of M consisting of elements of multidegree $\underline{d}$ is denoted by $M_{\underline{d}}$.

We now consider the linear map

$$\begin{aligned} \Phi \ : \ (k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]_{(d-1, 2d-1, \ldots, nd-1)})^{n+1} &\longrightarrow \\ k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]_{(2d-1, 3d-1, \ldots, (n+1)d-1)} \ , & \quad (3.29) \\ (A_1, \ldots, A_{n+1}) \mapsto \textstyle\sum_{i=1}^{n+1} F_i A_i \ . & \end{aligned}$$

Note that both the domain as well as the codomain have dimension $(n+1)! \cdot d^n$. Now in [SZ94] it is proven:

Proposition 3.59 *Let M be the matrix of Φ with respect to the monomial bases in the domain and the codomain with any ordering. Then*

$$\mathrm{Res}(F_1, \ldots, F_{n+1}) = \pm \det(M) \ .$$

Note here that the resultant is only defined up to a sign, and a change of the ordering of any of the two the bases changes a sign too.

This description of the resultant immediately gives rise to the following result in the generic field RAM model:

Proposition 3.60 *Given $F_1, \ldots, F_{n+1}$ of multidegree $(d, \ldots, d)$ one can compute $\mathrm{Res}(F_1, \ldots, F_{n+1})$ with a number of field and bit operations which is polynomially bounded in $(n+1)! \cdot d^n$.*

Proposition 3.59 states in particular that $V(F_1, \ldots, F_{n+1})$ is empty if and only if Φ is surjective, that is, the ideal $(F_1, \ldots, F_{n+1})_{(2d-1,3d-1,\ldots,(n+1)d-1)}$ is equal to the whole ambient space $k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]_{(2d-1,3d-1,\ldots,(n+1)d-1)}$. This statement can be generalized:

Proposition 3.61 *Let $F_1, \ldots, F_m$ be multihomogeneous polynomials in $k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ of multidegree $(d, d, \ldots, d)$. Then $V(F_1, \ldots, F_m)$ is empty if and only if $(F_1, \ldots, F_m)_{(2d-1,3d-1,\ldots,(n+1)d-1)} = k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]_{(2d-1,3d-1,\ldots,(n+1)d-1)}$.*

Proof. It is obvious that the latter statement implies the former. So let $V(F_1, \ldots, F_m)$ be empty. To show the equality we can perform a base-change. So we consider the field extension $k((c_{i,j})_{i=1,\ldots,n+1,j=1,\ldots,m})|k$, where the $c_{i,j}$ are indeterminates, and let $G_i := \sum_{j=1}^{m} c_{i,j} F_j$ for $i = 1, \ldots, n+1$. By the following lemma, if $g_1, \ldots, g_{n+1}$ are obtained by any dehomogenization from $G_1, \ldots, G_{n+1}$, then $V(g_1, \ldots, g_{n+1})$ is empty. Thus $V(G_1, \ldots, G_{n+1})$ is empty too. Therefore $(G_1, \ldots, G_{n+1})_{(2d-1,3d-1,\ldots,(n+1)d-1)}$ is equal to the ambient space. This clearly implies that $(F_1, \ldots, F_m)_{(2d-1,3d-1,\ldots,(n+1)d-1)}$ is equal to the ambient space too. □

Lemma 3.62 *Let k be a field, and let R be a non-trivial commutative noetherian k-algebra of dimension n. Let $f_1, \ldots, f_m \in R$ with $(f_1, \ldots f_m) = R$. Now let $g_i := \sum_{j=1}^{m} c_{i,j} f_j$ in $R \otimes_k k((c_{i,j})_{i,j})$ for $i = 1, \ldots, n+1$. Then $g_1, \ldots, g_{n+1}$ generate the unit ideal of $R \otimes_k k((c_{i,j})_{i,j})$.*

Proof. This statement follows from the following statement by induction on n:

Let R be a non-trivial commutative noetherian k-algebra, and let $f_1, \ldots, f_m \in R$ with $(f_1, \ldots, f_m) = R$. Then $\dim((R \otimes_k k(c_1, \ldots, c_m))/(c_1 f_1 + \cdots + c_m c_m)) < \dim(R \otimes_k k(c_1, \ldots, c_m)) = \dim(R)$, where we define the dimension of the trivial algebra as -1.

To prove this we have to show that $c_1 f_1 + \cdots + c_m f_m$ is not contained in any minimal prime ideal of $R \otimes_k k(c_1, \ldots, c_m)$.

Now, the minimal prime ideals of $R \otimes_k k(c_1, \ldots, c_m)$ are exactly the ideals of the form $(\mathfrak{p})$, where $\mathfrak{p}$ is a minimal prime ideal of R. (Let first $\mathfrak{p}$ be a prime ideal of R. Then $R/(\mathfrak{p}) \otimes_k k[c_1, \ldots, c_n] \simeq (R/\mathfrak{p})[c_1, \ldots, c_n]$ is a domain. Therefore $R/\mathfrak{p} \otimes_k k(c_1, \ldots, c_n) \simeq (R \otimes_k k(c_1, \ldots, c_n))/(\mathfrak{p})$, which is a localization of the previous ring, is a domain too. Thus $(\mathfrak{p})$ is prime. Let us assume that $\mathfrak{p}$ is minimal, and let $\mathfrak{P} \subseteq (\mathfrak{p})$ be a prime

ideal of $R \otimes_k k(c_1, \ldots, c_m)$. Then $\mathfrak{P} \cap R = \mathfrak{p}$ by the minimality of $\mathfrak{p}$, thus $(\mathfrak{P} \cap R) = (\mathfrak{p})$. As $(\mathfrak{P} \cap R) \subseteq \mathfrak{P}$, we have $\mathfrak{P} = (\mathfrak{p})$. We conclude that $(\mathfrak{p})$ is a minimal prime ideal. Now let $\mathfrak{P}$ be any minimal prime ideal of $R \otimes_k k(c_1, \ldots, c_m)$. Then by what we just have shown $(\mathfrak{P} \cap R)$ is then a prime ideal of $R \otimes_k k(c_1, \ldots, c_m)$. Moreover, this ideal is obviously contained in $\mathfrak{P}$ and thus equal to $\mathfrak{P}$. Now $\mathfrak{P} \cap R$ is also minimal because otherwise $(\mathfrak{P} \cap P)$ was not minimal either.)

So let us fix a minimal prime ideal $\mathfrak{p}$ of R. By assumption the residue classes $[f_1]_\mathfrak{p}, \ldots, [f_m]_\mathfrak{p}$ generate $R/\mathfrak{p}$. This implies in particular that there exists some $i \in \{1, \ldots, m\}$ such that $[f_i]_\mathfrak{p} \neq 0$. Therefore $c_1[f_1]_\mathfrak{p} + \cdots + c_m[f_m]_\mathfrak{p} \neq 0 \in R/\mathfrak{p} \otimes_k k(c_1, \ldots, c_m) \simeq (R \otimes_k k(c_1, \ldots, c_m))/(\mathfrak{p})$. Thus $c_1 f_1 + \cdots + c_m f_m \notin (\mathfrak{p})$. □

Proposition 3.61 implies immediately:

Proposition 3.63 *Given $F_1, \ldots, F_m \in k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ as above, one can determine if $V(F_1, \ldots, F_m)$ is empty in a number of field and bit operations which is polynomially bounded in $m \cdot n! \cdot d^n$.*

We now prove:

Proposition 3.64 *Given multihomogeneous polynomials $F_1, \ldots, F_n \in k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ of multidegree $(d, d, \ldots, d)$ as well as a list of $(n+1) \cdot d^n$ distinct elements from k, one can determine in a number of field and bit operations which is polynomially bounded in $n! \cdot d^n$ if $V(F_1, \ldots, F_n)$ is zero-dimensional. If this is the case, one can compute in an expected number of field and bit operations which is polynomially bounded in $n! \cdot d^n$ all its k-rational points.*

Proof. Let for $i = 1, \ldots, n$ $p_i : (\mathbb{P}^1_k)^n \longrightarrow \mathbb{P}^1_k$ be the projection to i^{th} component. Then $V(F_1, \ldots, F_n)$ is not zero-dimensional if and only if there is some $i = 1, \ldots, n$ such that $p_i(V(F_1, \ldots, F_n))$ is equal to $\mathbb{P}^1_k$. (If $p_i(V(F_1, \ldots, F_n)) = \mathbb{P}^1_k$ for some i, then clearly $V(F_1, \ldots, F_n)$ is not zero-dimensional. Otherwise $V(F_1, \ldots, F_n)$ is contained in the finite set $\bigcap_{i=1}^n p_i^{-1}(p_i(V(F_1, \ldots, F_n)))$.)

For each $i = 1, \ldots, n$ we consider the multigraded resultant of $F_1, \ldots, F_n$ with respect to all coordinates except $\mathrm{X}_i, \mathrm{Y}_i$. Let us denote this resultant by $\mathrm{Res}_{\vee\atop(\mathrm{X}_i, \mathrm{Y}_i)}(F_1, \ldots, F_n)$. By definition $\mathrm{Res}_{\vee\atop(\mathrm{X}_i, \mathrm{Y}_i)}(F_1, \ldots, F_n)$ vanishes exactly on $p_i(V(F_1, \ldots, V_n))$, and one easily see with Proposition 3.57 c) that this is a homogeneous polynomial of degree $n! \cdot d^n$. We conjecture in fact that it defines the scheme-theoretic image of $V(F_1 \ldots, V_n)$ under p_i but we do not need this statement.

We thus see that $V(F_1, \ldots, F_n)$ is not zero-dimensional if and only if

at least one of the resultants $\operatorname{Res}_{\vee_{(\mathrm{X}_i,\mathrm{Y}_i)}}(F_1,\ldots,F_n)$ vanishes. We can thus decide if $V(F_1,\ldots,F_n)$ is zero-dimensional or not by checking if all these resultants are non-trivial.

Each of these resultants is a homogeneous polynomial of degree $n!\cdot d^n$, thus it vanishes if and only if it vanishes on $n!\cdot d^n+1$ distinct points in $\mathbb{P}^1(\overline{k})$.

By assumption we have a list of $n!\cdot d^n$ elements of k at our disposal. Including ∞ these give $n!\cdot d^n+1$ elements of $\mathbb{P}^1(k)$. We can therefore check if the resultant $\operatorname{Res}_{\vee_{(\mathrm{X}_i,\mathrm{Y}_i)}}(F_1,\ldots,F_n)$ vanishes by computing all the resultants obtained by substituting X_i and Y_i with the $n!\cdot d^n+1$ elements of $\mathbb{P}^1(k)$ and checking if the result is 0. By Proposition 3.60 each of these computations can be performed in a number of field operations which is polynomially bounded in $n!\cdot d^{n-1}$, and there are $n\cdot(n!\cdot d^n+1)$ resultants to be computed. The overall running time is thus polynomially bounded in $n!\cdot d^n$. We have shown the first statement of the proposition.

We now come to the computation of the k-rational solutions, provided that the system is indeed zero-dimensional.

We start off in the same way as above, and from the "evaluated resultants" we compute the resultants $\operatorname{Res}_{\vee_{(\mathrm{X}_i,\mathrm{Y}_i)}}(F_1,\ldots,F_n)$ by interpolation. For this we again compute the "evaluated resultants" as determinants as in Proposition 3.59. Here for each i all the $n!\cdot d^n+1$ matrices have to be computed with respect to the same ordering of monomials in order that the sign is consistent.

By assumption all these resultants are non-trivial. We factorize them and determine their roots in k; let L_i be a list of the roots of the i^{th} resultant, that is, of the k-rational points of $p_i(V(F_1,\ldots,F_n))$.

We now compute the solutions in an iterative manner, by successively imposing conditions of the coordinates. We start out with the k-rational points in $p_1(V(F_1,\ldots,F_n))$, that is, L_1. Suppose now that we know the k-rational points of $(p_1,\ldots,p_i)(V(F_1,\ldots,F_n))$, which we have stored in a list S_i. Then for each point of $P=(P_1,\ldots,P_i)$ in S_i and Q in L_{i+1}, we check if the system obtained by substituting P for $\mathrm{X}_1,\mathrm{Y}_1,\ldots,\mathrm{X}_i,\mathrm{Y}_i$ and Q for $(\mathrm{X}_{i+1},\mathrm{Y}_{i+1})$ is consistent, that is if it has a solution over $\overline{k}$. Then all tuples (P,Q) are inserted into a new list S_{i+1} for later inspection. Note here the important point that the list S_i has $\leq n!\cdot d^n$ elements.

Let us give the algorithm in a more formal way:

Algorithm for solving multihomogeneous zero-dimensional systems

Input: Multihomogeneous polynomials $F_1, \ldots, F_n \in k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ of multidegree $(d, d, \ldots, d)$ and $n! \cdot d^n$ distinct field elements $a_1, \ldots, a_{n! \cdot d^n}$ such that $V(F_1, \ldots, F_n)$ is zero-dimensional.

1. For each $i = 1, \ldots, n$, compute $\mathrm{Res}_{\vee_{(\mathrm{X}_i, \mathrm{Y}_i)}}(F_1, \ldots, F_n)$ by interpolation. (*Each of these resultants is non-trivial by assumption.*)
2. Factorize these resultants and compute their roots in $\mathbb{P}^1(k)$. Let L_i be a list of roots of the i^{th} resultant.
3. Let $S_1 \longleftarrow L_1$.
4. For $i = 1, \ldots, n-1$ do
   Determine a list S_{i+1} consisting of elements of $(\mathbb{P}^1(k))^{i+1}$ as follows:
   For each $P = (P_1, \ldots, P_i) \in L$ and $Q \in L_{i+1}$ check if the system obtained by substituting P for $\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_i, \mathrm{Y}_i$ and Q for $\mathrm{X}_{i+1}, \mathrm{Y}_{i+1}$ is consistent. If this is the case, insert (P, Q) into S_{i+1}.
5. Output S_n.

It is obvious that each of the lists S_i contains exactly the k-rational points of $(p_1, \ldots, p_i)V(F_1, \ldots, F_n)$. Thus the output of the algorithm consists of the k-rational points of $V(F_1, \ldots, F_n)$.

Let us analyze the complexity: Step 1 can clearly be performed in a number of field operations which is polynomially bounded in $n! \cdot d^n$ (cf. Proposition 3.60). Step 3 can be performed in an expected running time of $\mathcal{O}(n! \cdot d^n)$ bit and field operations. Each of the checks in Step 4 can be performed with a number of field and bit operations which is polynomially bounded in $n! \cdot d^n$ by Proposition 3.61. Now, as already remarked each list S_i contains at most $n! \cdot d^n$ elements. Therefore, there are at most $(n! \cdot d^n)^2$ tuples (P, Q) to be considered for each value of i. Thus Step 4 can also be performed with a number of bit and field operations which is polynomially bounded in $n! \cdot d^n$. □

Over finite fields one might have to pass to a field extension of degree $\leq \log_2(n! \cdot d^n)$ in order that enough field elements are available. By doing so, one obtains:

Proposition 3.65 *Given multihomogeneous polynomials $F_1, \ldots, F_n \in \mathbb{F}_q[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ of multidegree $(d, d, \ldots, d)$, one can determine if $V(F_1, \ldots, F_{n+1})$ is zero-dimensional and if this is the case compute all its $\mathbb{F}_q$-rational points in an expected time which is polynomially bounded in $n! \cdot d^n \cdot \log(q)$.*

3.5.4.4 Interpolation

Similarly to the algorithm above, the computation of the summation polynomials will be based on an interpolation. In contrast to the computation above, the result is however a multihomogeneous polynomial. Here we consider the corresponding interpolation problem.

Let us first consider the classical 1-dimensional interpolation problem in the context of homogeneous polynomials: Let $d \in \mathbb{N}$ and $(a_j, b_j) \in k^2 - \{0\}$ for $j = 1, \ldots, d+1$ such that the induced elements in $\mathbb{P}^1(k)$ are pairwise distinct. Moreover, let $c_1, \ldots, c_{d+1} \in k$. Then there is exactly one homogeneous polynomial $F(\mathrm{X}, \mathrm{Y}) \in k[\mathrm{X}, \mathrm{Y}]$ of degree d with $F(a_j, b_j) = c_j$ for all $j = 1, \ldots, d+1$. Moreover, with

$$L_j := \prod_{\ell \neq j} \frac{b_\ell \mathrm{X} - a_\ell \mathrm{Y}}{a_j b_\ell - a_\ell b_j} \tag{3.30}$$

we have

$$F = \sum_j c_j L_j \ . \tag{3.31}$$

Proposition 3.66 *Let $\underline{d} \in \mathbb{N}^n$, and let $S := \{1, \ldots, d_1 + 1\} \times \cdots \times \{1, \ldots, d_n + 1\}$. Let k be a field, let $(a_{i,j}, b_{,j}) \in k^2 - \{0\}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, d_i + 1$ such that for each i, the elements $(a_{i,1} : b_{i,1}), \ldots, (a_{i,d_i+1} : b_{i,d_i+1}) \in \mathbb{P}^1(k)$ are pairwise distinct, and let $c_{\underline{j}} \in k$ for $\underline{j} \in S$.*

Then there is exactly one multihomogeneous polynomial $F \in k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ of multidegree $\underline{d}$ with $F(a_{1,j_1}, b_{1,j_2}, \ldots, a_{n,j_n}, b_{n,j_n}) = b_{\underline{j}}$ for all $\underline{j} \in S$.

Proof. The case $n = 1$ is treated above. For the general case we proceed by induction on n.

Let us first prove the uniqueness. For this, let $\underline{d}$, S, k, and $(a_{i,j}, b_{,j}) \in k^2 - \{0\}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, d_i + 1$ be as in the proposition, and let $F \in k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ be of multidegree $\underline{d}$ with $F(a_{1,j_1}, b_{1,j_2}, \ldots, a_{n,j_n}, b_{n,j_n}) = 0$ for all $\underline{j} \in S$.

Then be induction hypothesis, for each $j = 1, \ldots, d_n + 1$, $F(\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_{n-1}, \mathrm{Y}_{n-1}, a_{n,j}, b_{n,j}) = 0 \in k[\mathrm{X}_1, Y_1, \ldots, \mathrm{X}_{n-1}, Y_{n-1}]$. We now regard $F(\mathrm{X}_1, Y_1, \ldots, \mathrm{X}_n, Y_n)$ as a bivariate homogeneous polynomial in the ring $k(\mathrm{X}_1, Y_1, \ldots, \mathrm{X}_{n-1}, Y_{n-1})[\mathrm{X}_n, \mathrm{Y}_n]$. Then by the uniqueness of the solution of the 1-dimensional interpolation problem, we conclude that $F = 0$.

We come to the existence. So let objects as in the proposition be given.

For each $j = 1, \ldots, d_n + 1$ there is by induction assumption exactly one multihomogeneous polynomial $C_j \in k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_{n-1}, \mathrm{Y}_{n-1}]$ of multidegree $(d_1, \ldots, d_{n-1})$ with $C_j(a_{1,j_1}, b_{1,j_2}, \ldots, a_{n-1,j_{n-1}}, b_{n-1,j_{n-1}}) = c_{\underline{j}}$ for all

$\underline{j} \in S$ with $j_n = j$. Let $L_j := \prod_{\ell \neq j} \frac{b_\ell X_n - a_\ell Y_n}{a_j b_\ell - a_\ell b_j}$ for $j = 1, \ldots, d_n + 1$. Then the polynomial $F := \sum_j C_j L_j$ fulfills the requirements. □

We call the computation problem to determine the polynomial F, given the data in the proposition the *multihomogeneous interpolation problem*. One can solve this problem with an obvious linear algebra approach. We therefore obtain:

Proposition 3.67 *The multihomogeneous interpolation problem can be solved in a number of field and bit operations which is polynomially bounded in* $(d_1+1)\cdots(d_n+1)$*, where as above* $\underline{d}$ *is the multidegree of the polynomial to be computed.*

3.5.5 The summation polynomials

In this subsection we prove Propositions 3.35 and 3.37 on the summation polynomials. Let E be an elliptic curve over a field k, let $m \in \mathbb{N}, m \geq 2$, and let $\varphi : E \longrightarrow \mathbb{P}^1_k$ be a covering of degree 2 which satisfies $\varphi \circ [-1] = \varphi$.

Now let N_m (or N) be the kernel of the addition map $E^m \longrightarrow E$, $(P_1, \ldots, P_m) \mapsto P_1 + \cdots + P_m$. (Here the P_i are Z-valued points for some k-scheme Z.) Note that N is isomorphic to E^{m-1} via the projection $(P_1, \ldots, P_m) \mapsto (P_1, \ldots, P_{m-1})$.

We now consider the projection $E^m \longrightarrow (\mathbb{P}^1_k)^m$ induced by φ. Note that $[-1]$ operates on N, and the map $N \hookrightarrow E^m \longrightarrow (\mathbb{P}^1_k)^m$ factors through the quotient $N/[-1]$.

Definition 3.68 Let $H_{\varphi,m}$ (or H_m or H) be the image of N in $(\mathbb{P}^1_k)^m$ (with the induced subscheme structure).

Lemma 3.69

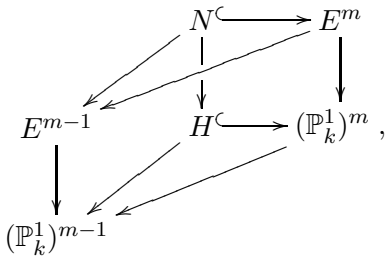
a) The induced map $N/[-1] \longrightarrow H$ *is finite and birational.*

b) H *is a hyperplane* $(\mathbb{P}^1_k)^m$ *of multidegree* $(2^{m-2}, \ldots, 2^{m-2})$.

c) The projections $H \longrightarrow \mathbb{P}^{m-1}_K$ *to any* $m-1$ *of the* m *components are flat coverings of degree* 2^{m-2}.

Proof. The maps $N \hookrightarrow E^m \longrightarrow (\mathbb{P}^1_k)^m$ and $H \longrightarrow (\mathbb{P}^1_k)^m$ are clearly finite. It follows immediately that the induced map $N \longrightarrow H$ is also finite. This in turn implies that the induced map $N/[-1] \longrightarrow H$ is finite too (by definition of the geometric quotient).

Let us now consider the commutative diagram



where the morphisms $E^m \longrightarrow E^{m-1}$ and $(\mathbb{P}^1_k)^m \longrightarrow (\mathbb{P}^1_k)^{m-1}$ are the projections to the first $m-1$ coordinates. Then the induced morphism $N \longrightarrow E^{m-1}$ is an isomorphism, and the morphism $E^{m-1} \longrightarrow (\mathbb{P}^1_k)^{m-1}$ is finite separable of degree 2^{m-1}.

Below we show that the map $N \longrightarrow H$ has degree 2, and the map $H \longrightarrow (\mathbb{P}^1_k)^{m-1}$ generically has degree 2^{m-2}. This statement implies statements a) and b) in the lemma. Indeed, first as $N \longrightarrow H$ has degree 2, the induced map $N/[-1] \longrightarrow H$ has degree 1, that is, it is birational. Second, the fact that the map $H \longrightarrow (\mathbb{P}^1_k)^{m-1}$ is quasi-finite and generically of degree 2^{m-2} implies that the last component of the multidegree of H is 2^{m-2}. By "by symmetry" (or by a repetition of the argument with projections to different components) then all components of the multidegree are 2^{m-2}.

Note first that we have already established that both maps are separable, and that the product of the two degrees is 2^{m-1}. Therefore, it suffices to show that the extension of function fields $k(N)|k(H)$ has separability degree 2.

We are going to apply the isomorphism $E^{m-1} \longrightarrow N$ which is the inverse of the projection $N \longrightarrow E^{m-1}$ and consider the extension $k(E^{m-1})|k(H)$.

Let $\Omega := \overline{k(E^{m-1})}$, let $p_i : E^{m-1} \longrightarrow E$ be the projection to the i^{th} coordinate, and let $P_i \in E(\Omega)$ be the induced points. (That is, P_i is the morphism $\mathrm{Spec}(\Omega) \longrightarrow \mathrm{Spec}(k(E^{m-1})) \longrightarrow E^{m-1} \xrightarrow{p_i} E$, where the first two morphisms are the canonical ones.) Let $p_m := -\sum_{i=1}^{m-1} p_i$ and $P_m := -\sum_{i=1}^{m-1} P_i$.

Then the inverse of the projection $N \longrightarrow E^{m-1}$ to the first $n-1$ coordinates is given by $(p_1, \ldots, p_m)$; the corresponding $k(E^{n-1})$-valued point of N is given by $(P_1, \ldots, P_m)$.

The P_i are linearly independent, since the p_i are linearly independent, the map $\mathrm{Mor}_k(E^{m-1}, E) \longrightarrow E(k(E^{m-1}))$ is injective (in fact, it is an isomorphism), and the map $E(k(E^{m-1})) \longrightarrow \mathrm{Spec}(\Omega)$ is injective too.

Now let us consider the preimage of $x(P_1, \ldots, P_m) = (x \circ P_1, \ldots, x \circ P_m) \in H(\Omega)$ in $N(\Omega)$. This set consists of all tuples $(\epsilon_1 P_1, \ldots, \epsilon_m P_m) \in E^m(\Omega)$ with $\epsilon_i = \pm 1$ and $\sum_{i=1}^m \epsilon_i P_i = O$. Clearly, there are exactly two such tuples:

$\pm(P_1, \ldots, P_m)$.

We conclude: There are exactly two Ω-valued points of E^{m-1} which induce the Ω-valued point $(x \circ P_1, \ldots, x \circ P_m) \in H(\Omega)$ under the projection $N \longrightarrow H$. This means that there are exactly two extensions of the canonical inclusion $k(E^{m-1}) \longrightarrow \Omega$ to $k(N)$. Therefore, the separability degree of the extension $k(E^{m-1})|k(H)$ is 2.

We come to c). We still (wlog.) only consider the projection $p : H \longrightarrow (\mathbb{P}^1_k)^{m-1}$ to the first $m-1$ components. As the map is quasi-finite and as H has multidegree $(2^{m-2}, \ldots, 2^{m-2})$, each fiber has degree 2^{m-2}. With other words: The Hilbert polynomials of the fibers are equal to 2^{m-2}. With [Har77, Theorem 9.9] we conclude that p is flat.

Note that H is a projective over $(\mathbb{P}^1)^{m-1}$, thus in particular proper. Moreover, p is quasi-finite. These two properties together are equivalent to being finite by [Gro61, Proposition 4.4.2]. □

Now clearly, if S is any irreducible polynomial in $k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_m, \mathrm{Y}_m]$ which is multihomogeneous, then S satisfies the conditions of Proposition 3.35 if and only if $H = V(S)$. This establishes Proposition 3.35.

Thus the m^{th} summation polynomial (cf. Definition 3.36) with respect to φ is the (up to a multiplicative constant unique) polynomial S with $V(S) = H$.

Remark 3.70 Let $\alpha \in \mathrm{Aut}(\mathbb{P}^1_k)$. Then $H_{\alpha\circ\varphi,m} = \alpha(H_{\varphi,m})$, with other words: $H_{\alpha^{-1}\circ\varphi,m} = \alpha^{-1}(H_{\varphi,m})$. This implies that $S_{\alpha^{-1}\circ\varphi,m} = \alpha^*(S_{\varphi,m})$.

We now discuss how the summation polynomials for elliptic curves in Weierstraß form can be given in an explicit and constructive way, following [Sem98].

Lemma 3.71 *Let E be an elliptic curve in $\mathbb{P}^2_k$ in Weierstraß form:*

$$E = V(Y^2Z + a_1XYZ + a_3YZ^2 - (X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3))$$

with $a_1, a_2, a_3, a_4, a_6 \in k$ and $O = [0 : 1 : 0]$. Then the 3^{rd} summation polynomial of E with respect to $x_{|E}$ is given by

$$\begin{aligned}
&\big((\mathrm{x}_1^2\mathrm{x}_2^2 + \mathrm{x}_2^2\mathrm{x}_3^2 + \mathrm{x}_1^2\mathrm{x}_3^2) - 2(\mathrm{x}_1^2\mathrm{x}_2\mathrm{x}_3 + \mathrm{x}_1\mathrm{x}_2^2\mathrm{x}_3 + \mathrm{x}_1\mathrm{x}_2\mathrm{x}_3^2)\\
&-(a_1^2 + 4a_2)\mathrm{x}_1\mathrm{x}_2\mathrm{x}_3 - (a_1a_3 + 2a_4)\cdot(\mathrm{x}_1\mathrm{x}_2 + \mathrm{x}_2\mathrm{x}_3 + \mathrm{x}_1\mathrm{x}_3)\\
&-(a_3^2 + 4a_6)\cdot(\mathrm{x}_1 + \mathrm{x}_2 + \mathrm{x}_3)\\
&-a_1^2a_6 + a_1a_3a_4 - a_2a_3^2 - 4a_2a_6 + a_4^2)\cdot \mathrm{Y}_1^2\mathrm{Y}_2^2\mathrm{Y}_3^2\ .
\end{aligned}$$

Sketch of the proof. Let S be the polynomial in the lemma. Using the inversion and addition formulae for elliptic curves in Weierstraß form (cf. [Sil86]),

one can check (with a rather lengthy computation) that for all $P_1, P_2 \in E(\overline{k})$, $S(x(P_1), x(P_2), x(P_1 + x(P_3)) = 0$. This implies that S_3 divides S. As both polynomials have multidegree $(2, 2, 2)$, it follows that they are equal. Let us note here that one only has to check that $S(x(P_1), x(P_2), x(P_1 + P_3))$ for $P_1 \neq \pm P_2$ and $P_1, P_2 \neq O$ because then S vanishes on an open part of H_3 and thus also on all of H_3. □

Let us indicate how the polynomial S was *found*, following [Sem04]:

Let $P_1, P_2 \in E(\overline{k})$ with $P_1, P_2 \neq O$ and $P_1 \neq \pm P_2$. Then clearly both $x(P_1+P_2)$ and $x(P_1-P_2)$ satisfy the polynomial $(x-x(P_1+P_2))(x-x(P_1+P_2))$. So we computed this polynomial over the field $\mathbb{Q}(a_1, a_2, a_3, a_4, a_6)$ and for "generic" P_1, P_2, using the computer algebra system MAGMA. The polynomial S is then obtained by multiplication with the denominator and homogenization.

Lemma 3.72 *Let E still be an elliptic curve and $\varphi : E \longrightarrow \mathbb{P}^1_k$ a covering of degree 2 with $\varphi \circ [-1] = \varphi$. Let $s, t \in \mathbb{N}$ with $s, t \geq 2$. Then*

$$S_{\varphi,s+t}(\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_{s+t}, \mathrm{Y}_{s+t}) =$$

$$\mathrm{Res}_{(\mathrm{X},\mathrm{Y})}(S_{\varphi,s+1}(\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_s, \mathrm{Y}_s, \mathrm{X}, \mathrm{Y}),$$
$$S_{\varphi,t+1}(\mathrm{X}_{s+1}, \mathrm{Y}_{s+1}, \ldots, \mathrm{X}_{s+t}, \mathrm{Y}_{s+t}, \mathrm{X}, \mathrm{Y})) .$$

Here by $\mathrm{Res}_{(\mathrm{X},\mathrm{Y})}$ we mean the usual Sylvester resultant for homogeneous polynomials in X and Y of degrees 2^{s-1} and 2^{t-1}.

Proof. For $(P_1, \ldots, P_{s+t}) \in (E(\overline{k}))^{s+t}$ we have $P_1 + \cdots + P_{s+t} = O$ if and only if there exists some $P \in E(\overline{k})$ with $P_1 + \cdots + P_s + P = O$ and $P_{s+1} + \cdots + P_{s+t} + P = O$.

It follows that topologically the hyperplane H_{s+t} is the image of $V(S_{\varphi,s+1}(\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_s, \mathrm{Y}_s, \mathrm{X}, \mathrm{Y}), S_{\varphi,t+1}(\mathrm{X}_{s+1}, Y_{s+1}, \ldots, \mathrm{X}_{s+t-1}, \mathrm{Y}_{s+t}, \mathrm{X}, \mathrm{Y}))$ in $(\mathbb{P}^1_k)^n \times \mathrm{Proj}(k[\mathrm{X}, \mathrm{Y}])$ under the projection to $(\mathbb{P}^1_k)^n$. As H_{s+t} is irreducible it follows that the resultant in the lemma is (up to a multiplicative constant) a power of $S_{\varphi,s+t}$.

In order to prove that the resultant is (up to a constant) equal to $S_{\varphi,s+t}$, we consider their multidegrees.

The generic Sylvester resultant for polynomials of degrees a and b has degree b in the coefficients of the first polynomial and degree a in the coefficients of the second polynomial. We apply this with $a = 2^{s-1}$ and $b = 2^{t-1}$. In our case we obtain a polynomial of degree $2^{s-1} \cdot 2^{t-1} = 2^{s+t-2}$ in $(\mathrm{X}_i, \mathrm{Y}_i)$ for all $i = 1, \ldots, s + t$.

As $S_{\varphi,s+t}$ has multidegree $(2^{s+t-2}, \ldots, 2^{s+t-2})$, the result follows. □

The two preceeding lemmata give rise to algorithmic constructions of the summation polynomials.

First, given an elliptic curve in Weierstraß form and a covering of degree 2 $\varphi : E \longrightarrow \mathbb{P}^1_k$ with $\varphi \circ [-1] = \varphi$ (which means that the automorphism $\alpha \in \mathrm{Aut}(\mathbb{P}^1_k)$ with $\varphi = \alpha \circ x_{|E}$ is given), one can easily determine $S_{\varphi,3}$ via Lemma 3.71 and Remark 3.70.

Further, one can compute $S_{\varphi,m}$ for $m \geq 3$ from $S_{\varphi,m-1}$ and $S_{\varphi,3}$ by applying the above lemma with $s = m-2$ and $t = 2$. This computation can be performed via interpolation provided we have 2^{m-2} distinct field elements available (which give rise to $2^{m-2}+1$ distinct elements in $\mathbb{P}^1(k)$) (cf. Proposition 3.67). We therefore obtain:

Proposition 3.73 *Given an elliptic curve E over a field k in Weierstraß form and a covering $\varphi : E \longrightarrow \mathbb{P}^1_k$ of degree 2 with $\varphi \circ [-1] = \varphi$, a natural number $m \geq 3$ and a list of 2^{m-2} elements of k, one can compute the m^{th} summation polynomial of E with respect to φ in a number of field and bit operations of $\mathcal{P}oly(e^{m^2})$.*

Proposition 3.37 is a consequence of this proposition. (It might be necessary to compute in field extensions.)

3.5.6 Geometric background on the algorithm and analysis

The main purpose of this subsection is to prove Propositions 3.41 and 3.47. Additionally, we give some background information on the definition of the factor base from a geometric point of view.

3.5.6.1 Weil restrictions

We make use of *Weil restrictions* of schemes. Here we briefly recall the definition and some basic properties of Weil restrictions. For further information we refer to [BLR80, 7.6] and [Die01, Chapter 1]. Let us first fix the following terminology:

Terminology 3.74 Let X and Y be locally noetherian schemes. Then a finite and flat morphism $X \longrightarrow Y$ is also called a *flat covering*.

Note that a flat covering is locally free (see [Mat89, Theorem 7.10]).

Now let S' and S locally noetherian schemes, and let a flat covering $S' \longrightarrow S$ be fixed. Let X' be an S'-scheme such that the fibers of X' over S' are quasi-projective. Then one can show that the functor from S-schemes to sets $Z \mapsto \mathrm{Mor}_{S'}(Z_{S'}, X')$ is representable by an S-scheme; the (up to unique isomorphism unique) representing object is called the *Weil restriction* of X' with respect to $S' \longrightarrow S$. We denote it by $\mathrm{Res}^{S'}_S(X')$.[11]

[11] The similarity between the notations for Weil restrictions and resultants is accidental.

A reformulation of this definition is: The Weil restriction of X' with respect to $S' \longrightarrow S$ is a an S-scheme $\mathrm{Res}_S^{S'}(X')$ together with an S'-morphism $u : (\mathrm{Res}_S^{S'}(X'))_{S'} \longrightarrow X'$ such that the following holds: Whenever Z is an S-scheme, and $\alpha : Z \times_S S' = Z_{S'} \longrightarrow X'$ is an S'-morphism, there is a unique S-morphism $\beta : Z \longrightarrow \mathrm{Res}_S^{S'}(Z)$ with $\alpha = \beta_{S'} \circ u$, where $\beta_{S'} := \beta \times_S S' = \beta \times_S \mathrm{id}_{S'}$. We denote the morphism β by $\alpha_\circledcirc$.

The assignment $X \mapsto \mathrm{Res}_S^{S'}(X')$ gives rise to a functor (which we call *scalar-restriction functor*) from the category of S'-schemes with quasi-projective fibers to the category of S-schemes. Moreover, if X' is an affine S'-scheme (that is, the structural morphism is affine), then $\mathrm{Res}_S^{S'}(X')$ is an affine S-scheme.

We will use the following two lemmata.

Lemma 3.75 *Let $S' \longrightarrow S'$ be as above, and let X', Y', W' be S'-schemes with S'-morphisms $X' \longrightarrow W'$ and $Y' \longrightarrow W'$. Then we have a Cartesian diagram*

$$\begin{array}{ccc} \mathrm{Res}_S^{S'}(X' \times_{W'} Y') & \longrightarrow & \mathrm{Res}_S^{S'}(Y') \\ \downarrow & & \downarrow \\ \mathrm{Res}_S^{S'}(X') & \longrightarrow & \mathrm{Res}_S^{S'}(W') \end{array}$$

with the obvious canonical morphisms.

Proof. Let for this Z be an S-scheme, and let two S-morphisms $Z \longrightarrow \mathrm{Res}_S^{S'}(X')$ and $Z \longrightarrow \mathrm{Res}_S^{S'}(Y')$ be given which induce the same morphism $Z \longrightarrow \mathrm{Res}_S^{S'}(W')$. These morphisms induce S'-morphisms $Z_{S'} \longrightarrow X'$ and $Z_{S'} \longrightarrow Y'$, again inducing the same morphism to W'. Thus by the universal property of the product, we have an induced morphism $Z_{S'} \longrightarrow X'$. Again this morphism corresponds to a morphism $Z \longrightarrow \mathrm{Res}_S^{S'}(X')$; it is immediate that it has the correct property. □

Lemma 3.76 *Let $S' \longrightarrow S'$ as above, let T be an S-scheme, and let $T' := T \times_S S'$. Let X' be a T'-scheme with structural morphism $\alpha : X' \longrightarrow T'$.*

Let $v : (\mathrm{Res}_T^{T'}(X'))_{T'} \longrightarrow X'$ be the universal morphism; v is thus a T'-morphism. We have $(\mathrm{Res}_T^{T'}(X')) \times_T T' \simeq (\mathrm{Res}_T^{T'}(X')) \times_S S'$, and v is in particular an S'-morphism. Thus by the universal property of $\mathrm{Res}_S^{S'}(X')$ we have an induced S-morphism $v_\circledcirc : \mathrm{Res}_T^{T'}(X') \longrightarrow \mathrm{Res}_S^{S'}(X')$.

Now we have a Cartesian diagram

$$\begin{array}{ccc} \mathrm{Res}_T^{T'}(X') & \longrightarrow & \mathrm{Res}_S^{S'}(X') \\ \downarrow & & \downarrow \\ T & \longrightarrow & \mathrm{Res}_S^{S'}(T') \ , \end{array}$$

where the morphisms are defined as follows: The left morphism is the structural morphism, the right morphism is $\mathrm{Res}_S^{S'}(\alpha)$*, the upper morphism is* $v_\circledcirc$*, and the lower morphism is the morphism* $\mathrm{id}_\circledcirc : T \longrightarrow \mathrm{Res}_S^{S'}(T')$ *corresponding to the identity on* T' *under the defining functorial property of* $\mathrm{Res}_S^{S'}(T')$*.*

This lemma follows from [Die01, Chapter I, Lemma 1.2].

Let now $K|k$ be a finite field extension. Then if X' is a quasi-projective (resp. projective) scheme over K, $\mathrm{Res}_k^K(X')$ is a quasi-projective (resp. projective) scheme of dimension $[K : k] \cdot \dim(X')$ over k. Note that by the defining functorial property of the Weil restriction we have in particular a bijection

$$\begin{aligned} X'(K) = \mathrm{Mor}_K(\mathrm{Spec}(K), X') \longrightarrow \mathrm{Res}_k^K(X')(k) = \mathrm{Mor}_k(\mathrm{Spec}(k), \mathrm{Res}_k^K(X')), \\ \mapsto P_\circledcirc \ . \end{aligned}$$

If X' is a group scheme over K, $\mathrm{Res}_k^K(X')$ is in a natural way again a group scheme, and in particular if A' is an abelian variety over K, then $\mathrm{Res}_k^K(A)$ is in a natural way an abelian variety too.

Let $K|k$ now be an extension of finite fields of degree n, and let σ be the relative Frobenius automorphism of $K|k$. We denote the induced isomorphism $\mathrm{Spec}(k) \longrightarrow \mathrm{Spec}(k)$ again by σ. Let X' be a quasi-projective K-scheme. Then we have a canonical isomorphism

$$(\mathrm{Res}_k^K(X'))_K \simeq \prod_{i=0}^{n-1} \sigma^i(X')$$

of K-schemes under which the universal morphism $u : (\mathrm{Res}_k^K(X'))_K \longrightarrow X'$ corresponds to the projection

$$u : \prod_{i=0}^{n-1} \sigma^i(X') \longrightarrow X' \ .$$

Moreover, if Z is any k-scheme and $\alpha : Z_K \longrightarrow X'$ is a morphism, then $(\alpha_\circledcirc)_K$ corresponds to

$$(\alpha, \sigma(\alpha), \ldots, \sigma^{n-1}(\alpha)) : Z_K \longrightarrow \prod_{i=0}^{n-1} \sigma^i(X')$$

and if $\varphi : X' \longrightarrow Y'$ is a morphism of quasi-projective K-schemes, then $\mathrm{Res}_k^K(\varphi)$ corresponds to

$$\varphi \times \sigma(\varphi) \times \cdots \times \sigma^{n-1}(\varphi) : \prod_{i=0}^{n-1} \sigma^i(X') \longrightarrow \prod_{i=0}^{n-1} \sigma^i(Y') \ .$$

3.5.6.2 Background on the factor base

Let still $K|k$ be an extension of finite fields of degree n, and as above let σ the Frobenius automorphism relative to k. Let E be an elliptic curve over K, and let us fix a covering $\varphi : E \longrightarrow \mathbb{P}^1_K$ of degree 2 with $\varphi \circ [-1] = \varphi$.

Let $\iota = \mathrm{id}_{\circledcirc} : \mathbb{P}^1_k \longrightarrow \mathrm{Res}^K_k(\mathbb{P}^1_K)$ be the morphism corresponding to the identity on $\mathbb{P}^1_K$. One can easily see (for example via base change to K) that ι is a closed immersion.

Let V be the preimage of $\iota_{\circledcirc}(\mathbb{P}^1_k)$ under $\mathrm{Res}^K_k(\varphi) : \mathrm{Res}^K_k(E) \longrightarrow \mathrm{Res}^K_k(\mathbb{P}^1_k)$. This means by definition that we have a Cartesian diagram

$$\begin{array}{ccc} V & \hookrightarrow & \mathrm{Res}^K_k(E) \\ \downarrow & & \downarrow{\scriptstyle \mathrm{Res}^K_k(\varphi)} \\ \mathbb{P}^1_k & \xrightarrow{\iota_{\circledcirc}} & \mathrm{Res}^K_k(\mathbb{P}^1_k)\,. \end{array} \tag{3.32}$$

Note that $\mathrm{Res}^K_k(\varphi) : \mathrm{Res}^K_k(E) \longrightarrow \mathrm{Res}^K_k(\mathbb{P}^1_k)$ is a flat covering of degree 2^n (as one sees after base change to K), and therefore $V \longrightarrow \mathbb{P}^1_k$ is a flat covering of degree 2^n too.

Let us now explain the connection of these definitions to the definition of the factor base in the algorithm for Theorem 4: Let us consider a particular run of the algorithm. Then under the bijection $\mathbb{P}^1(K) \simeq \mathrm{Res}^K_k(\mathbb{P}^1_K)(k)$ the inclusion $\mathbb{P}^1(k) \subseteq \mathbb{P}^1(K)$ corresponds to $\iota_{\circledcirc}(\mathbb{P}^1_k(k)) \subseteq \mathrm{Res}^K_k(\mathbb{P}^1_K))(k)$. Therefore the factor base $\mathcal{F} = (\varphi^{-1}(\mathbb{A}^1_k))(k)$ corresponds to $V(k) - \{O\}$ under the bijection $E(K) \simeq \mathrm{Res}^K_k(E)(k)$. One can therefore say that the factor base is defined in a "geometric way" – something that immediately apparent from the definition of the factor base in the algorithm.

The addition on the Weil restriction induces a morphism $V^n \longrightarrow \mathrm{Res}^K_k(E)$, and – again under the bijection $E(K) \simeq \mathrm{Res}^K_k(E)(k)$ – for $P \in E(K)$ the tuples $(P_1, \ldots, P_n)$ with $\varphi(P_i) \in \mathbb{P}^1(k)$ and $\sum_i P_i = P$ correspond to the k-valued points of the fiber of $V^n \longrightarrow \mathrm{Res}^K_k(E)$ at $P_{\circledcirc}$, the k-rational point of $\mathrm{Res}^K_k(E)$ corresponding to P.

3.5.6.3 Study of the factor base

In this subsubsection we study the addition morphism $V^n \longrightarrow \mathrm{Res}^K_k(E)$.

Proposition 3.77 *Let Condition (3.45) be satisfied. Then*

a) V is geometrically reduced and geometrically irreducible (and thus birational to a curve),

b) the addition morphism $V^n \longrightarrow \mathrm{Res}^K_k(E)$ is surjective.

Proof. By (3.32) and Lemma 3.76 we have $V \simeq \mathrm{Res}^{\mathbb{P}^1_K}_{\mathbb{P}^1_k}(E)$, with respect to the covering $\varphi : E \longrightarrow \mathbb{P}^1_k$. This implies that

$$V_K \simeq E \times_{\mathbb{P}^1_K} \sigma(E) \times_{\mathbb{P}^1_K} \cdots \times_{\mathbb{P}^1_K} \sigma^{n-1}(E) \,, \tag{3.33}$$

where the morphisms are $\varphi : E \longrightarrow \mathbb{P}^1_K, \ldots, \sigma^{n-1}(\varphi) : \sigma^{n-1}(E) \longrightarrow \mathbb{P}^1_K$.

Let us now fix an algebraic closure $\overline{k(\mathrm{x})}$ of $k(\mathrm{x})$. Let us denote the Frobenius automorphism of $\overline{k}|k$ also by σ. Let us then prolong σ first to $\overline{k}(\mathrm{x})$ via $\sigma(\mathrm{x}) := \mathrm{x}$, and and let us fix any automorphism of $\overline{k(\mathrm{x})}|k(x)$ which restricts to σ; let us denote this automorphism again by σ. Moreover, let us fix an injection of $\overline{k}(E)$ into $\overline{k(\mathrm{x})}$ over $k(\mathrm{x})$.

We now consider the total quotient ring of the scheme $V_{\overline{k}}$, which is isomorphic to

$$\overline{k}(E) \otimes_{k(\mathrm{x})} \sigma(\overline{k}(E)) \otimes_{\overline{k}(\mathrm{x})} \cdots \otimes_{\overline{k}(\mathrm{x})} \sigma^{n-1}(\overline{k}(E)) \,.$$

By Condition (3.45) for $i = 1, \ldots, n-1$, the extension $\sigma^i(\overline{k}(E))|\overline{k}(\mathrm{x})$ is ramified at $\sigma^i(P)$ but for any $j = 0, \ldots, i-1$, the extension $\sigma^j(\overline{k}(E))|\overline{k}(\mathrm{x})$ is not ramified at $\sigma^i(P)$, thus the extension $\overline{k}(E)\sigma(\overline{k}(E)) \cdots \sigma^{i-1}(\overline{k}(E))|\overline{k}(\mathrm{x})$ in $\overline{k(\mathrm{x})}$ is also not ramified at $\sigma^i(P)$. Thus $\sigma^i(\overline{k}(E))$ is not contained in $\overline{k}(E)\sigma(\overline{k}(E)) \cdots \sigma^{i-1}(\overline{k}(E))$. It follows therefore by induction that the extension $\overline{k}(E)\sigma(\overline{k}(E) \cdots \sigma^{n-1}(\overline{k}(E))|\overline{k}(\mathrm{x})$ in $\overline{k(\mathrm{x})}$ has degree 2^n. Thus the total quotient ring of $V_{\overline{k}}$ is is isomorphic to the composite $\overline{k}(E)\sigma(\overline{k}(E)) \cdots \sigma^{n-1}(\overline{k}(E))$ in $\overline{k(\mathrm{x})}$ and therefore a field. We see that $V_{\overline{k}}$ is reduced and irreducible, thus V is geometrically reduced and geometrically irreducible.

We come to b). Let $\mathcal{C}$ be the curve which is birational to V, and let us fix a birational morphism $\pi : \mathcal{C} \longrightarrow V$. Let $\beta : \mathcal{C} \longrightarrow \mathrm{Res}^K_k(E)$ be the composition of π with the inclusion into $\mathrm{Res}^K_k(E)$. We claim that the induced homomorphism

$$\beta_* : J_{\mathcal{C}} \longrightarrow \mathrm{Res}^K_k(E)$$

is surjective.

We show this statement after base change to K. Let $v : V = \mathrm{Res}^{\mathbb{P}^1_K}_{\mathbb{P}^1_k}(E) \times_{\mathbb{P}^1_k} \mathbb{P}^1_K \simeq \mathrm{Res}^{\mathbb{P}^1_K}_{\mathbb{P}^1_k}(E) \times_k K \longrightarrow E$ be the universal morphism (this is the projection to the first factor in (3.33)), and as always let $u : (\mathrm{Res}^K_k(E))_K \longrightarrow E$ be the universal morphism. Let $\psi := v \circ \pi_K : \mathcal{C}_K \longrightarrow E$.

Note that the inclusion $V = \mathrm{Res}^{\mathbb{P}^1_K}_{\mathbb{P}^1_k}(E) \longrightarrow \mathrm{Res}^K_k(E)$ is given by $v_\circledcirc$ (cf. Lemma 3.76). We therefore have

$$\beta = v_\circledcirc \circ \pi = (v \circ \pi_K)_\circledcirc = \psi_\circledcirc \,.$$

Thus β_K is given by

$$(\psi, \sigma(\psi), \ldots, \sigma^{n-1}(\psi)) : \mathcal{C}_K \longrightarrow E \times \sigma(E) \times \cdots \times \sigma^{n-1}(E) \ ,$$

and we have

$$(\beta_*)_K = (\psi_*, \sigma(\psi)_*, \ldots, \sigma^{n-1}(\psi)_*) \ .$$

Let now $\gamma : E \times \sigma(E) \times \cdots \times \sigma^{n-1}(E) \longrightarrow J_{\mathcal{C}}$ be homomorphism which is induced by $(\sigma^i(\psi))^* : \sigma^i(E) \longrightarrow (J_{\mathcal{C}})_K$. (Under the identification of $J_{\mathcal{C}}$ with its dual, this is the dual homomorphism to $(\beta_*)_K$.)

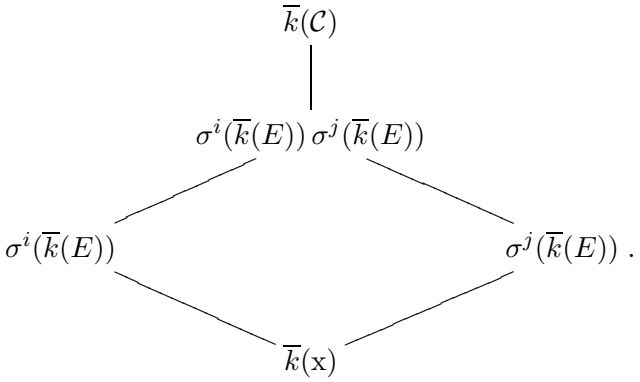
We claim that

$$(\beta_*)_K \circ \gamma = [2^{n-1}] \ .$$

For this we have to show:

- $i = 0, \ldots, n-1 : (\sigma^i(\psi))_* \circ (\sigma^i(\psi))^* = [2^{n-1}]$
- $i, j = 0, \ldots, n$ with $i \neq j : (\sigma^j(\psi))_* \circ (\sigma^i(\psi))^* = 0$

By the proof for item a) the extension $K(\mathcal{C})|K(\mathrm{x}) \simeq K(V)|K(\mathrm{x})$ is isomorphic to the extension $K(E) \cdots \sigma(K(E)) \cdots \sigma^{n-1}(K(E))|K(\mathrm{x})$ (where the composite is with respect to the fixed inclusions into $\overline{k(\mathrm{x})}$). The first claim is therefore obvious. For the second claim we consider the diagram of function fields



As $\sigma^i(\overline{k}(E))\,\sigma^j(\overline{k}(E))|\overline{k}(\mathrm{x})$ is a $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$-extension, the conorm-norm homomorphism from $\mathrm{Cl}^0(\sigma^i(\overline{k}(E)))$ to $\mathrm{Cl}^0(\sigma^j(\overline{k}(E)))$ via $\sigma^i(\overline{k}(E))\,\sigma^j(\overline{k}(E))$ is trivial. Thus the conorm-norm homomorphism via $\overline{k}(C)$ is trivial too. This gives the second claim.

We have therefore derived that $\beta_* : J_{\mathcal{C}} \longrightarrow \mathrm{Res}_k^K(E)$ is surjective.

Let for $i \in \mathbb{N}$ $a_i : V^i \longrightarrow \mathrm{Res}_k^K(E)$ be the map which is induced by the addition in $\mathrm{Res}_k^K(E)$. Assume now that the map $a_n : V^n \longrightarrow \mathrm{Res}_k^K(E)$ is not surjective. Then $a_n(V^n)$ a variety of dimension $< n$. Therefore $\dim(a_i(V^i)) = \dim(a_{i+1}(V^{i+1}))$ for some $i < n$. Moreover, $a_{i+1}(V^{i+1})$

clearly contains a translate of $a_i(V^i)$. Thus $a_{i+1}(V^{i+1})$ is equal to a translate of $a_i(V^i)$. By induction one easily sees that $a_m(V^m)$ is a translate of $a_i(V^i)$ for all $m \geq i$. But the image of β_* is $a_{g(\mathcal{C})}(V^{g(\mathcal{C})})$, a contradiction. □

□

Proposition 3.78 *Let us still assume that Condition 3.45) is satisfied, and let $\mathcal{C}$ be as above. Then*

a) The genus of $\mathcal{C}$ is $\leq (2n-1)\cdot(2^n-1)$.

b) $\mathcal{C}(\overline{k})$ contains at most $n \cdot 2^{n+2}$ points which map to singular points under the birational morphism $\pi : \mathcal{C} \longrightarrow V$.

Proof. By a general result on elementary abelian extensions (see e.g. [KR89]) we have

$$g(\mathcal{C}) = \sum_L g(L) \ ,$$

where L runs over all subextensions of $\overline{k}(\mathcal{C})|\overline{k}(\mathrm{x})$ of degree 2. We show below that the genus of a function field L as in the sum is always $\leq 2n-1$. This implies that $g(\mathcal{C}) \leq (2n-1)\cdot(2^n-1)$.

To show the claim on the subfields L we proceed with a case distinction.

Let q be even. By Artin-Schreier theory every subfield L of $\overline{k(\mathrm{x})}|k(\mathrm{x})$ of degree 2 corresponds to the a 1-dimensional subspace of the $\mathbb{F}_2$-vector space $\overline{k}(\mathrm{x})/\mathcal{P}(\overline{k}(\mathrm{x}))$, where $\mathcal{P}$ is the Artin-Schreier operator.

If now $\overline{k}(E)$ corresponds to $\langle \overline{f} \rangle$, where $\overline{f}$ is the residue class of some $f \in \overline{k}(\mathrm{x})$, then each field L as in the sum corresponds to $\langle a_0\overline{f} + a_1\overline{\sigma(f)} + \cdots + a_{n-1}\overline{\sigma^{n-1}(f)} \rangle$ for a uniquely defined tuple $(a_0, \ldots, a_{n-1}) \in \mathbb{F}_2^n - \{0\}$.

Let first $j(E) = 0$. In this case the extension $\overline{k}(E)|\overline{k}(\mathrm{x})$ is ramified at one place, and $\overline{k}(E)$ corresponds to some space $\langle \overline{f} \rangle$, where f is either a polynomial of degree 3 or of the form $\frac{g}{(x-\lambda)^3}$ for $\lambda \in \overline{k}$ and $\deg(g) = 3$.

Using [Sti93, Proposition III.7.8] one sees: If L is any field as in the sum, then $L|\overline{k}(\mathrm{x})$ is ramified at at most n places (this is also immediately obvious), and the corresponding discriminant exponents are all 4. This implies that the genus of L is $\leq 2n-1$.

Let now $j(E) \neq 0$. In this case $\overline{k}(E)|\overline{k}(\mathrm{x})$ is ramified at 2 places, and $\overline{k}(E)$ corresponds to $\langle \overline{f} \rangle$, where f is the sum of two distinct polynomials f_1, f_2 such that each of these polynomials is either x or $\frac{1}{\mathrm{x}-a}$ for some $a \in \overline{k}$. Now each subfield L as in the sum is ramified over at most $2n$ places and the different exponents are all 2. Again the genus of L is $\leq 2n-1$.

Let q be odd. In this case $\overline{k}(E)|\overline{k}(\mathrm{x})$ is (tamely) ramified at 4 places. If thus L is as in the sum, $L|\overline{k}(\mathrm{x})$ is ramified at at most $4n$ places. Thus the genus of L is $\leq 2n-1$.

We come to b). Let S be the set of points of $\mathbb{P}^1(\overline{k})$ over which one of the coverings $\sigma^i(E) \longrightarrow \mathbb{P}^1_k$ is ramified. Using the fact that a morphism obtained from an étale morphism via base change is étale we obtain: The canonical morphism $V \longrightarrow \mathbb{P}^1_k$ is étale outside S. This implies that V is smooth outside the preimage of S, and the birational morphism $\pi : \mathcal{C} \longrightarrow V$ is an isomorphism outside the preimages of S. With other words: All points in $\mathcal{C}(\overline{k})$ which map to singular points of V are contained in the preimage of S.

As the covering $\mathcal{C} \longrightarrow \mathbb{P}^1_k$ has degree 2^n, the preimage of the set S has at most $\#S \cdot 2^n \leq 4n \cdot 2^n$ elements. □

Remark 3.79 Let $k = \mathbb{F}_q$. Then under Condition 3.45 by the above propositions, and the Hasse-Weil bound we have

$$\begin{aligned} &\#\{P \in E(K) \mid \varphi(P) \in \mathbb{P}^1(k)\} = \#V(k) \\ &\geq q + 1 - 2 \cdot (2n-1) \cdot (2^n - 1) \cdot q^{\frac{1}{2}} - n \cdot 2^{n+2} + 1 \, . \end{aligned}$$

For fixed n this is clearly $\sim q$. Moreover, for $\log_2(q) \geq 3n$, that is, $2^{\frac{3}{2} \cdot n} \leq q^{\frac{1}{2}}$, and n large enough we have

$$V(k) \geq \frac{1}{2} \cdot (q+1) \, .$$

(The bound $q \geq 3 \log_2(n)$ is a bit arbitrary but it serves its purposes, and in order to complete the analysis for the algorithm for Theorem 4, we anyway have to impose a more restricted bound.)

This result shows that if φ satisfies Condition 3.45, the set $\{P \in E(K) \mid \varphi(P) \in \mathbb{P}^1(k)\}$ is "reasonably large". Note that this applies then of course in particular to the factor base constructed in the algorithm for Theorem 4. We remark however that the main purpose of showing that V is birational to a curve is not to prove that a suitably large factor base can be efficiently constructed – this goal can also easily be reached by choosing the automorphism α used to define φ in a randomized fashion. Rather the key statement is that V^n contains an irreducibility component which maps surjectively to $\mathrm{Res}^K_k(E)$ under the addition morphism and which contains "enough" elements.

Remark 3.80 There is a connection between the considerations here and the GHS-attack ([GHS02]) for elliptic curves: Let the notations be as above but let us drop Condition 3.45. Then $V_{\overline{k}}$ can be reducible. Let us assume that V contains an irreducibility component which is geometrically irreducible; let $\mathcal{D}$ be a curve which is birational to such a component, and let us fix a map from $\mathcal{D}$ to V which is birational to its image. The map from $\mathcal{D}$ to $\mathrm{Res}^K_k(E)$ corresponds to a covering $c : \mathcal{D}_K \longrightarrow E$.

Now the GHS-attack applied to E consists of finding such a $\mathcal{D}$ and the covering explicitly and transferring the discrete logarithm problem in $E(K)$ to a discrete logarithm problem in $\mathrm{Cl}^0(\mathcal{D})$ via the homomorphism

$$E(K) \xrightarrow{c^*} \mathrm{Cl}^0(\mathcal{D}_K) \longrightarrow \mathrm{Cl}^0(\mathcal{D}) \ ,$$

where the second homomorphism is the norm. This homomorphism is also related to the considerations above: One can show that under the isomorphism $E(K) \simeq \mathrm{Res}_k^K(E)$ this homomorphism corresponds to the pull-back homomorphism from $\beta^* : \mathrm{Res}_k^K(E) \longrightarrow J_{\mathcal{D}} \simeq \mathrm{Cl}^0(\mathcal{D})$.

The idea of the GHS attack is that it might be faster to solve the discrete logarithm problem in $\mathrm{Cl}^0(\mathcal{D})$ than in $E(K)$. Of course, for this one requires that the kernel of the homomorphism is "small" and the genus of $\mathcal{D}$ is as small as possible under this condition.

For the analysis of the present algorithm it is important that $V_{\overline{k}}$ is irreducible. In the GHS attack one has in a certain sense an opposite requirement on the covering $\varphi : E \longrightarrow \mathbb{P}^1_K$ used to define V than we have here: one wants that V contains a component which is birational a curve $\mathcal{C}$ of "small genus", however still such that the induced addition morphism $\mathcal{C}^n \longrightarrow \mathrm{Res}_k^K(E)$ is surjective.

We would however like to stress that the Weil restriction $\mathrm{Res}_k^K(E)$ is not needed for an explicit construction of such a curve $\mathcal{D}$. Indeed, the function field of $K(\mathcal{D})$ is a composite of $K(E), K(\sigma(E)), \ldots, K(\sigma^{n-1}(E))$, and one can construct such a composite and then the function field of a suitable curve $\mathcal{D}$ over k (provided that such a curve exists) with a purely field-theoretic approach. For further information on the function field theoretic approach to the GHS attack we refer to [Die03] and [Heß03].

3.5.6.4 The role of the summation polynomials

Let the hyperplane $H = H_{n+1}$ of $(\mathbb{P}^1_k)^{n+1}$ be defined as in subsection 3.5.5.

By applying the scalar-restriction functor, we obtain:

$$\mathrm{Res}_k^K(H) \longrightarrow \mathrm{Res}_k^K((\mathbb{P}^1_K)^{n+1}) \simeq (\mathrm{Res}_k^K(\mathbb{P}^1_K))^{n+1} \ .$$

Via base change to K one sees immediately that we have a closed immersion.

Let X be the scheme-theoretic preimage of $\mathrm{Res}_k^K(H)$ in $(\mathbb{P}^1_k)^n \times \mathrm{Res}_k^K(\mathbb{P}^1_K)$ under the closed immersion $\iota_\circledcirc \times \iota_\circledcirc \times \cdots \times \iota_\circledcirc \times \mathrm{id} : (\mathbb{P}^1_k)^n \times \mathrm{Res}_k^K(\mathbb{P}^1_K) \longrightarrow \mathrm{Res}_k^K((\mathbb{P}^1))^{n+1}$. This means by definition that we have a Cartesian diagram

$$\begin{array}{ccc} X & \hookrightarrow & \mathrm{Res}_k^K(H) \\ \downarrow & & \downarrow \\ (\mathbb{P}^1_k)^n \times \mathrm{Res}_k^K(\mathbb{P}^1_K) & \hookrightarrow & (\mathrm{Res}_k^K(\mathbb{P}^1_K))^{n+1} \ . \end{array} \tag{3.34}$$

Note that – again under the obvious bijections – the elements of $X(k)$ correspond to the tuples $(Q_1, \ldots, Q_n, Q)$ with $Q_i \in \mathbb{P}^1(k)$ and $Q \in \mathbb{P}^1(K)$ with $(Q_1, \ldots, Q_n, Q) \in H(K)$. The latter condition means of course that there are $P_1, \ldots, P_n, P \in E(\overline{K})$ with $\varphi(P_i) = Q_i$ and $\sum_i P_i = P$.

Notation 3.81 Let $p_1 : (\mathbb{P}^1_k)^n \times \mathrm{Res}^K_k(\mathbb{P}^1_K) \longrightarrow (\mathbb{P}^1_k)^n$ and $p_2 : (\mathbb{P}^1_k)^n \times \mathrm{Res}^K_k(\mathbb{P}^1_K) \longrightarrow \mathrm{Res}^K_k(\mathbb{P}^1_K)$ be the two projections.

Lemma 3.82 *$(p_1)_{|X} : X \longrightarrow (\mathbb{P}^1_k)^n$ is a flat covering of degree $2^{(n-1)\cdot n}$.*

Proof. By Lemma 3.69 c) the projection to the first n components $H \longrightarrow (\mathbb{P}^1_K)^n$ is a flat covering of degree 2^{n-1}. Therefore the induced map $\mathrm{Res}^K_k(H) \longrightarrow \mathrm{Res}^K_k((\mathbb{P}^1_K)^n) \simeq (\mathrm{Res}^K_k(\mathbb{P}^1))^n$ is a flat covering of degree $2^{(n-1)\cdot n}$. The map $(p_1)_{|X} : X \longrightarrow (\mathbb{P}^1_k)^n$ is obtained from this map via base change with $\iota \times \cdots \times \iota : (\mathbb{P}^1_k)^n \longrightarrow (\mathrm{Res}^K_k(\mathbb{P}^1_K))^n$. □

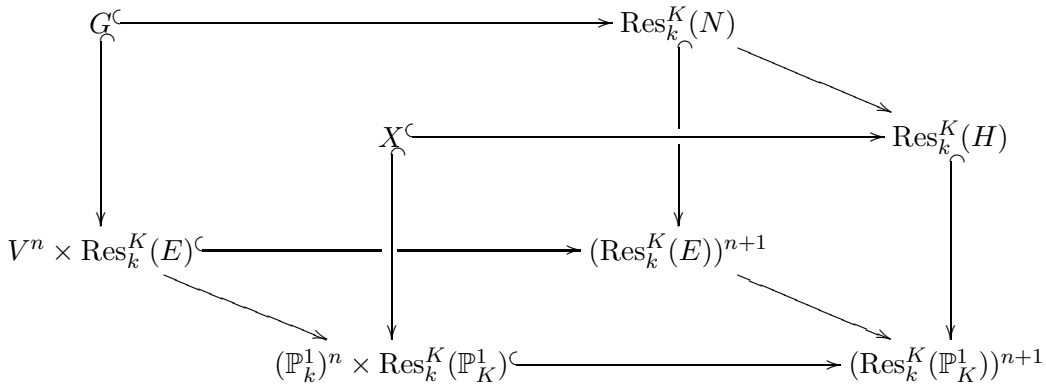
Notation 3.83 Let G be the graph of $-a_n : V^n \longrightarrow \mathrm{Res}^K_k(E)$, where as in the proof of Proposition 3.77 a_n is the restriction of the addition morphism to V^n. (Note the minus sign!)

As in subsection 3.5.5 let for $m \in \mathbb{N}$ N_m be the kernel of the addition morphism $E^m \longrightarrow E$. One easily sees that $\mathrm{Res}^K_k(N_m)$ is (as a subscheme of $\mathrm{Res}^K_k(E^m)$) the kernel of the addition homomorphism on $\mathrm{Res}^K_k(E^m)$. Let now $N := N_{n+1}$. By considering Z-valued points for any k-scheme Z, one obtains immediately:

Lemma 3.84 *G is the scheme-theoretic intersection of $V^n \times \mathrm{Res}^K_k(E)$ and $\mathrm{Res}^K_k(N)$ in $\mathrm{Res}^K_k(E^{n+1}) \simeq (\mathrm{Res}^K_k(E))^{n+1}$.*

Proposition 3.85 *There is a canonical surjective morphism $G \longrightarrow X$. Moreover, if Condition 3.45 is satisfied, then X is geometrically irreducible and $(p_2)_{|X} : X \longrightarrow \mathrm{Res}^K_k(\mathbb{P}^1_K)$ is surjective.*

Proof. Let us consider the commutative diagram



with the obvious canonical morphisms. As by definition of X the right-lower subdiagram (i.e. diagram (3.34)) is Cartesian, we have an induced morphism $G \longrightarrow X$.

It suffices to prove the surjectivity on $\overline{k}$-valued points. So let $Q \in X(\overline{k})$. As the map $N \longrightarrow H$ is surjective, so is $\mathrm{Res}_k^K(N) \longrightarrow \mathrm{Res}_k^K(H)$. Let us consider Q as a point in $\mathrm{Res}_k^K(H)$, and let us fix a preimage $P \in \mathrm{Res}_k^K(N)(\overline{k})$.

We claim that P lies in $G(\overline{k})$, or with other words that the image of P in $(\mathrm{Res}_k^K(E))^{n+1}(\overline{k})$ lies in $(V^n \times \mathrm{Res}_k^K(E))(\overline{k})$. For this we have to check that the image of P in $\mathrm{Res}_k^K(\mathbb{P}^1_K))(\overline{k})$ lies in $((\mathbb{P}^1)^n \times \mathrm{Res}_k^K(\mathbb{P}^1_K))(\overline{k})$. But this is obvious as the image is nothing but the point Q we started with.

Let now Condition 3.45 be satisfied. By Proposition 3.77 a) V is then geometrically reduced and geometrically irreducible, thus so is V^n, which is isomorphic to the graph G. As the map $G \longrightarrow X$ is surjective, X is then also geometrically irreducible.

For the second claim we consider the commutative diagram

$$\begin{array}{ccc} G & \longrightarrow & X \\ \downarrow & & \downarrow \\ V^n \times \mathrm{Res}_k^K(E) & \longrightarrow & (\mathbb{P}^1_k)^n \times \mathrm{Res}_k^K(\mathbb{P}^1_K) \\ \downarrow & & \downarrow{\scriptstyle p_2} \\ \mathrm{Res}_k^K(E) & \longrightarrow & \mathrm{Res}_k^K(\mathbb{P}^1_K)\ . \end{array}$$

The image of G in $\mathrm{Res}_k^K(E)$ is the image of $-a_n : V^n \longrightarrow \mathrm{Res}_k^K(E)$ (which is equal to the image of a_n), and by Proposition 3.77 b) this is $\mathrm{Res}_k^K(E)$. Moreover, the morphism $\mathrm{Res}_k^K(E) \longrightarrow \mathrm{Res}_k^K(\mathbb{P}^1_K)$ (being a flat covering) is also surjective. Thus the morphism $(p_2)_{|X} : X \longrightarrow \mathrm{Res}_k^K(\mathbb{P}^1_K)$ is surjective too. □

Let us now fix some $Q \in \mathbb{P}^1(K)$. Following our notation, let $Q_\circledcirc$ be the corresponding k-rational point of $\mathrm{Res}_k^K(\mathbb{P}^1_K)$. Let $X_{Q_\circledcirc}$ be the fiber of X at $Q_\circledcirc$, that is, we have the Cartesian diagram

$$\begin{array}{ccc} X_{Q_\circledcirc} & \hookrightarrow & X \\ \downarrow & & \downarrow \\ (\mathbb{P}^1_k)^n & \hookrightarrow & (\mathbb{P}^1_k)^n \times \mathrm{Res}_k^K(\mathbb{P}^1_K) \\ \downarrow & & \downarrow \\ \mathrm{Spec}(k) & \xrightarrow{Q_\circledcirc} & \mathrm{Res}_k^K(\mathbb{P}^1_K)\ . \end{array}$$

Then we have the following connection with the decomposition algorithm:

Proposition 3.86 *As a subscheme of* $(\mathbb{P}^1_k)^n \times \mathrm{Res}^K_k(\mathbb{P}^1_K)$, $X_{Q_\circledcirc}$ *is* $V(S^{(1)}, \ldots, S^{(n)})$, *where the polynomials* $S^{(j)} \in k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$ *are defined as in Equation (3.28).*

We show first:

Lemma 3.87 *Let* $H_Q \subset (\mathbb{P}^1_K)^n$ *be the restriction of* H *to* $(\mathbb{P}^1_K)^n$ *via the closed immersion* $\mathrm{id} \times \cdots \times \mathrm{id} \times Q : (\mathbb{P}^1_K)^n \simeq (\mathbb{P}^1_K)^n \times_K \mathrm{Spec}(K) \longrightarrow (\mathbb{P}^1_K)^{n+1}$. *Then we have a Cartesian diagram*

$$\begin{array}{ccc} X_{Q_\circledcirc} & \hookrightarrow & \mathrm{Res}^K_k(H_Q) \\ \downarrow & & \downarrow \\ (\mathbb{P}^1_k)^n & \hookrightarrow & (\mathrm{Res}^K_k(\mathbb{P}^1_K))^n \end{array},$$

where the lower arrow is given by $\iota \times \cdots \times \iota$.

Proof. We have $\mathrm{Res}^K_k(\mathrm{Spec}(K)) = \mathrm{Spec}(k)$ and $\mathrm{Res}^K_k(Q) = Q_\circledcirc$. By Lemma 3.75 the defining Cartesian diagram

$$\begin{array}{ccc} H_Q & \hookrightarrow & H \\ \downarrow & & \downarrow \\ (\mathbb{P}^1_K)^n & \hookrightarrow & (\mathbb{P}^1)^{n+1} \end{array}$$

gives rise to the Cartesian diagram

$$\begin{array}{ccc} \mathrm{Res}^K_k(H_Q) & \hookrightarrow & \mathrm{Res}^K_k(H) \\ \downarrow & & \downarrow \\ (\mathrm{Res}^K_k(\mathbb{P}^1_K))^n & \hookrightarrow & (\mathrm{Res}^K_k(\mathbb{P}^1_K))^{n+1} \end{array},$$

where the lower arrow is given by $\mathrm{id} \times \cdots \times \mathrm{id} \times Q_\circledcirc : (\mathrm{Res}^K_k(\mathbb{P}^1_K))^n \simeq (\mathrm{Res}^K_k(\mathbb{P}^1_K))^n \times_k \mathrm{Spec}(k) \longrightarrow (\mathrm{Res}^K_k(\mathbb{P}^1_K))^{n+1}$.

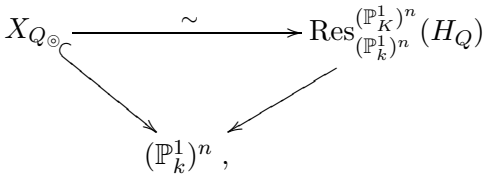
Now $X_{Q_\circledcirc}$ is the pull-back of $\mathrm{Res}^K_k(H)$ to $(\mathbb{P}^1_k)^n$ under the map $\iota \times \cdots \times \iota \times Q_\circledcirc : (\mathbb{P}^1_k)^n \simeq (\mathbb{P}^1_k)^n \times_k \mathrm{Spec}(k) \longrightarrow (\mathrm{Res}^K_k(\mathbb{P}^1_K))^{n+1}$. This implies that we have a Cartesian diagram

$$\begin{array}{ccccc} X_{Q_\circledcirc} & \hookrightarrow & \mathrm{Res}^K_k(H_Q) & \hookrightarrow & \mathrm{Res}^K_k(H) \\ \downarrow & & \downarrow & & \downarrow \\ (\mathbb{P}^1)^n_k & \hookrightarrow & (\mathrm{Res}^K_k(\mathbb{P}^1_K))^n & \hookrightarrow & (\mathrm{Res}^K_k(\mathbb{P}^1_K))^{n+1} \end{array}.$$

□

We come to the *proof* of Proposition 3.86.

By Lemma 3.87 and Lemma 3.76 we have a commutative diagram



where the arrow to the left is the structural morphism, which of course is then also a closed immersion.

To establish the result we thus have to show that as a closed subscheme of $(\mathbb{P}^1_k)^n$, $\mathrm{Res}^{(\mathbb{P}^1_K)^n}_{(\mathbb{P}^1_k)^n}(H_Q)$ is equal to $V(S^{(1)}, \ldots, S^{(n)})$.

Let now $S_{\varphi,n+1}$ be the same summation polynomial as in subsubsection 3.5.2.1 (recall that the $(n+1)^{\text{th}}$ summation polynomial with respect to φ is only unique up to multiplication by a non-trivial constant). Also, let $b_1, \ldots, b_n$ be the fixed k-basis of K from subsubsection 3.5.2.1. Note that $b_1, \ldots, b_n$ is then also a basis of the free $k[\mathrm{x}_1, \ldots, \mathrm{x}_n]$-module $K[\mathrm{x}_1, \ldots, \mathrm{x}_n]$. Moreover, let $S' := S_{\varphi,n+1}(\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n, Q)$ be the polynomial obtained by inserting the same coordinates of $Q = \varphi(P)$ into the summation polynomial as in 3.5.2.1 (again these are only unique up to multiplication by a non-trivial constant).

We now prove the result by restriction to affine parts of $(\mathbb{P}^1_k)^n$.

Let for the moment $\mathrm{X}_{i,1} := \mathrm{X}_i$ and $X_{i,2} := \mathrm{Y}_i$. Moreover, let for some multihomogeneous polynomial $F \in k[X_1, Y_1, \ldots, X_n, Y_n]$ $U_F := (\mathbb{P}^1_k)^n - V(F)$ be the corresponding open subscheme.

One can now show that for any $\underline{a} \in \{1,2\}^n$, the restrictions of both schemes to $U_{\mathrm{X}_{1,a_1}} \cap U_{\mathrm{X}_{2,a_2}} \cap \cdots \cap U_{\mathrm{X}_{n,a_n}}$ are equal; and this implies that the schemes are equal. For notational convenience we consider in the following the case of $\underline{a} = (2, \ldots, 2)$ ("dehomogenization with respect to $Y_1, \ldots, Y_n$"); the other cases can be established in exactly the same way.

Let $s(\mathrm{x}_1, \ldots, \mathrm{x}_n) := S'(\mathrm{x}_1, 1, \mathrm{x}_2, 1, \ldots, \mathrm{x}_n, 1) \in K[\mathrm{x}_1, \ldots, \mathrm{x}_n]$. Then $H_Q \cap \mathbb{A}^n_k \subseteq \mathbb{A}^n_k = \mathrm{Spec}(k[\mathrm{x}_1, \ldots, \mathrm{x}_n])$ corresponds to the quotient ring $k[\mathrm{x}_1, \ldots, \mathrm{x}_n]/(s)$ of $k[\mathrm{x}_1, \ldots, \mathrm{x}_n]$.

As the formation of the Weil restriction commutes with base-change on the base, we have $(\mathrm{Res}^{(\mathbb{P}^1_K)^n}_{(\mathbb{P}^1_k)^n}(H_Q)) \cap \mathbb{A}^n_k = \mathrm{Res}^{\mathbb{A}^n_K}_{\mathbb{A}^n_k}(H_Q \cap \mathbb{A}^n_K)$ as closed subschemes of $\mathbb{A}^n_k$. A defining system of polynomials for $\mathrm{Res}^{\mathbb{A}^n_K}_{\mathbb{A}^n_k}(H_Q \cap \mathbb{A}^n_K)$ can be derived via the well-known method to obtain defining equations for Weil restrictions of affine schemes over rings (see example [Die01, Chapter 1] or the proof of [BLR80, §7.6., Theorem 4]):

Let $s^{(1)}, \ldots, s^{(n)} \in k[\mathrm{x}_1, \ldots, \mathrm{x}_n]$ be defined by the equation

$$\sum_j b_j \, s^{(j)} = s \; .$$

Then $\mathrm{Res}^{\mathbb{A}^n_K}_{\mathbb{A}^n_k}(H_Q \cap \mathbb{A}^n_K) = \mathrm{Spec}(k[\mathrm{x}_1, \ldots, \mathrm{x}_n]/(s^{(1)}, \ldots, s^{(n)})) = V(s^{(1)}, \ldots, s^{(n)}) \subset \mathbb{A}^n_k$. But the $s^{(j)}$ are exactly the dehomogenizations of the polynomials $S^{(j)}$, and thus $(X_{Q_\circledcirc}) \cap \mathbb{A}^n_k = (\mathrm{Res}^{(\mathbb{P}^1_K)^n}_{(\mathbb{P}^1_k)^n}(H_Q)) \cap \mathbb{A}^n_k = \mathrm{Res}^{\mathbb{A}^n_K}_{\mathbb{A}^n_k}(H_Q \cap \mathbb{A}^n_K) = V(s^{(1)}, \ldots, s^{(n)}) = V(S^{(1)}, \ldots, S^{(n)}) \cap \mathbb{A}^n_k$ as subschemes of $\mathbb{A}^n_k$. □

3.5.6.5 Determination of non-zero-dimensional fibers

We are interested in the number of points $Q \in \mathbb{P}^1(K)$ for which the fiber $X_{Q_\circledcirc} = p_2^{-1}(Q_\circledcirc)$ is not zero-dimensional. For this we first consider a base change to K, such that X_K is a closed subscheme of $(\mathbb{P}^1_K)^n \times (\mathbb{P}^1_K)^n$, and we perform explicit computations in the Chow ring of $(\mathbb{P}^1_K)^n \times (\mathbb{P}^1_K)^n$. We identify for notational reasons $(\mathbb{P}^1)^n \times (\mathbb{P}^1)^n$ componentwise with $\prod_{i=1}^n \mathrm{Proj}(\mathbb{Z}[\mathrm{X}_{1,i}, \mathrm{Y}_{1,i}]) \times \prod_{i=1}^n \mathrm{Proj}(\mathbb{Z}[\mathrm{X}_{2,i}, \mathrm{Y}_{2,i}])$, and let $h_{\ell,i}$ be the class of $\mathrm{X}_{\ell,i}$ in the Chow ring of $(\mathbb{P}^1_K)^n \times (\mathbb{P}^1_K)^n$.

Lemma 3.88 *The class of X_K in $\mathrm{CH}((\mathbb{P}^1_K)^n \times (\mathbb{P}^1_K)^n)$ is $2^{(n-1)\cdot n} \prod_{i=1}^n (h_{1,1} + \cdots + h_{1,n} + h_{2,i})$.*

Proof. X_K is defined inside $(\mathbb{P}^1_K)^n \times (\mathbb{P}^1_K)^n$ by the polynomials

$$F_j := S_{\varphi,n+1}(\mathrm{X}_{1,1}, \mathrm{Y}_{1,1}, \ldots, \mathrm{X}_{1,n}, \mathrm{Y}_{1,n}, \mathrm{X}_{2,j}, \mathrm{Y}_{2,j})$$

for $j = 1, \ldots, n$. One can easily see with this explicit description that for all $\ell = 2, \ldots, n$ no irreducibility component of $V(F_1, \ldots, F_{\ell-1})$ is contained in $V(F_\ell)$.

Indeed, let C be an irreducibility component of $V(F_1, \ldots, F_{\ell-1})$. Then $C = C' \times (\mathbb{P}^1_K)^{n-\ell+1}$ for some $C' \subseteq (\mathbb{P}^1_K)^n \times (\mathbb{P}^1_K)^{\ell-1}$. Let $(Q_1, Q_2) \in C'(\overline{K})$, where $Q_1 \in (\mathbb{P}^1)^n(\overline{K})$ and $Q_2 \in (\mathbb{P}^1)^{\ell-1}(\overline{K})$. Now there are at most 2^{n-1} points in $Q_3 \in \mathbb{P}^1(\overline{K})$ with $F_\ell(Q_1, Q_3) = 0$. Choose some $Q_3 \in \mathbb{P}^1(\overline{K})$ which is distinct from these points, and choose $Q_4 \in (\mathbb{P}^1)^{n-\ell}(\overline{K})$ arbitrarily. Then (Q_1, Q_2, Q_3, Q_4) is a $\overline{K}$-valued point of C which does not lie in $V(F_\ell)$.

We therefore have $[X_K] = [V(F_1)] \cdots [V(F_n)]$ in the Chow ring of $(\mathbb{P}^1_K)^n \times (\mathbb{P}^1_K)^n$ (cf. Remark 3.53). Moreover, $[V(F_i)] = 2^{n-1}(h_{1,1} + \cdots + h_{1,n} + h_{2,i})$. This gives the statement. □

Remark 3.89 By the lemma we have in particular $((p_2)_K)_\circledcirc([X_K]) = n! \cdot 2^{(n-1)\cdot n}$, thus $(p_2)_K(X_K)$ is equal to the ambient space

$\prod_{i=1}^{n} \mathrm{Proj}(K[\mathrm{X}_{2,i}, \mathrm{Y}_{2,i}])$, or with other words: $(p_2)_{|X}$ is surjective. Another way to see this is: Let $Q = (Q_1, \ldots, Q_n) \in \prod_{i=1}^{n} \mathrm{Proj}(K[\mathrm{X}_{2,i}, \mathrm{Y}_{2,i}])(\overline{K})$. Then the geometric fiber X_Q is the subscheme of $\prod_{i=1}^{n} \mathrm{Proj}(\overline{K}[\mathrm{X}_{1,i}, \mathrm{Y}_{1,i}])$ defined by $F_i(\mathrm{X}_{1,1}, \mathrm{Y}_{1,n}, \ldots, \mathrm{X}_{1,n}, \mathrm{Y}_{1,n}, Q_i)$ for $i = 1, \ldots, n$. We see in particular that the fiber is never empty. More precisely, if it is zero-dimensional then its degree is $n! \cdot 2^{(n-1)\cdot n}$. Under Condition 3.45 we have already proven in Proposition 3.85 that $(p_2)_{|X}$ is surjective.

Let now $q_i : \prod_{i=1}^{n}(\mathrm{Proj}(K[\mathrm{X}_{1,i}, \mathrm{Y}_{1,i}])) \longrightarrow \mathrm{Proj}(K[\mathrm{X}_{1,i}, \mathrm{Y}_{1,i}])$ be the projection to the i^{th} component.

For some $Q \in \prod_{i=1}^{n}(\mathrm{Proj}(K[\mathrm{X}_{2,i}, \mathrm{Y}_{2,i}]))(\overline{K})$ the geometric fiber X_Q (which is contained in $\prod_{i=1}^{n} \mathrm{Proj}(\overline{K}[\mathrm{X}_{1,i}, \mathrm{Y}_{1,i}])$) is zero-dimensional if and only if for no $i = 1, \ldots, n$ the image of X_Q under q_i is equal to $\mathrm{Proj}(\overline{K}[\mathrm{X}_{1,i}, \mathrm{Y}_{1,i}])$.

Let $R_i \in K[\mathrm{X}_{1,i}, \mathrm{Y}_{1,i}, \mathrm{X}_{2,1}, \mathrm{Y}_{2,1}, \ldots, \mathrm{X}_{2,n}, \mathrm{Y}_{2,n}]$ be the multigraded resultant of $F_1, \ldots, F_n$ with respect to the variables $\mathrm{X}_{1,1}, \mathrm{Y}_{1,1}, \ldots, \mathrm{X}_{1,i-1}$, $\mathrm{Y}_{1,i-1}, \mathrm{X}_{1,i+1}, \mathrm{Y}_{1,i+1}, \ldots, \mathrm{X}_{1,n}, \mathrm{Y}_{1,n}$. Then for $Q = (Q_1, \ldots, Q_n) \in \prod_{i=1}^{n} \mathrm{Proj}(K[\mathrm{X}_{2,i}, \mathrm{Y}_{2,i}])(\overline{K})$ the geometric fiber X_Q is zero-dimensional if and only if for all $i = 1, \ldots, n$ $R_i(\mathrm{X}_i, \mathrm{Y}_i, Q_1, \ldots, Q_n)$ is non-trivial (cf. also the proof of Proposition 3.64).

Note now that not all fibers are non-zero-dimensional because X has dimension n (see Lemma 3.82) and $(\mathbb{P}^1_K)^n$ has dimension n too. Thus the polynomials $R_1, \ldots, R_n$ are all non-trivial.

Lemma 3.90 *Each polynomial R_i has multidegree $(n! \cdot 2^{(n-1)\cdot n}, (n-1)! \cdot 2^{(n-1)\cdot n}, \ldots, (n-1)! \cdot 2^{(n-1)\cdot n})$.*

Proof. The polynomials $F_1, \ldots, F_n$ have multidegree $(2^{n-1}, \ldots, 2^{n-1}) \in \mathbb{N}^{n-1}$ with respect to the variables under consideration. Therefore the corresponding generic resultant is homogeneous in the coefficients of each of the polynomials of degree $(n-1)! \cdot 2^{(n-1)^2}$. This implies that the degree with respect to $\mathrm{X}_{2,i}, \mathrm{Y}_{2,i}$ for some i is $(n-1)! \cdot 2^{(n-1)^2} \cdot 2^{n-1} = (n-1)! \cdot 2^{(n-1)\cdot n}$. Moreover, the degree with respect to $\mathrm{X}_{1,i}, \mathrm{Y}_{1,i}$ is $(n-1)! \cdot 2^{(n-1)^2} \cdot n \cdot 2^{n-1} = n! \cdot 2^{(n-1)\cdot n}$. □

Let us now for every $i = 1, \ldots, n$ fix some non-trivial coefficient C_i of R_i regarded as a polynomial in $K[\mathrm{X}_{2,n}, \mathrm{Y}_{2,n}, \ldots, \mathrm{X}_{2,n}, \mathrm{Y}_{2,n}][\mathrm{X}_{1,i}, \mathrm{Y}_{1,i}]$. Then clearly the points $Q \in \prod_{i=1}^{n} \prod \mathrm{Proj}(K[\mathrm{X}_{2,i}, \mathrm{Y}_{2,i}])$ for which the fiber X_Q is not zero-dimensional are contained in

$$\bigcup_{i=1}^{n} V(C_i) \subseteq (\mathbb{P}^1_K)^n \ .$$

Let us fix some $i = 1, \ldots, n$. Then $V(C_i)$ is an effective Cartier divisor of multidegree $((n-1)!\cdot 2^{(n-1)\cdot n}, \ldots, (n-1)!\cdot 2^{(n-1)\cdot n})$ in $\prod_{i=1}^{n} \mathrm{Proj}(K[\mathrm{X}_{2,i}, \mathrm{Y}_{2,i}])$, and $(p_2)_K^{-1}(V(C_i))$ is an effective Cartier divisor of multidegree $(0, \ldots, 0, (n-1)! \cdot 2^{(n-1)\cdot n}, \ldots, (n-1)! \cdot 2^{(n-1)\cdot n})$ in $(\mathbb{P}^1_K)^n \times (\mathbb{P}^1_K)^n$.

It follows that

$$[X_K] \cdot [(p_2)_K^{-1}(V(C_i))] = (n-1)! \cdot 2^{2(n-1)\cdot n} \cdot (\prod_{i=1}^{n}(h_{1,1} + \cdots + h_{1,n} + h_{2,i})) \cdot (h_{2,1} + \cdots + h_{2,n})$$

in $\mathrm{CH}((\mathbb{P}^1_K)^n \times (\mathbb{P}^1_K)^n)$. With Lemma 3.56 this implies that

$$\begin{aligned} & ((p_1)_K)_{\circledcirc}([X_K] \cdot [(p_2)_K^{-1}(V(C_i))]) \\ = \; & (n-1)! \cdot 2^{2(n-1)\cdot n} \cdot n \cdot (h_{1,1} + \cdots + h_{1,n}) \qquad (3.35) \\ = \; & n! \cdot 2^{2(n-1)\cdot n} \cdot (h_{1,1} + \cdots + h_{1,n}) \; . \end{aligned}$$

Assumption 3.91 Let us from now on assume that Condition 3.45 is satisfied.

Notation 3.92 Let $k = \mathbb{F}_q$ (such that $K = \mathbb{F}_{q^n}$).

Recall that X is now geometrically irreducible (Proposition 3.85). Clearly X_K is not contained in $V((p_2)_K^{-1}(C_i))$ (because otherwise $(p_2)_K(X_K)$ would be contained in $V(C_i)$, contradicting the surjectivity of p_2). Thus we have $[X_K] \cdot [V((p_2)_K^{-1}(C_i))] = [X_K \cap V((p_2)_K^{-1}(C_i))]$ (cf. Remark 3.53). As the map $(p_1)_K : X_K \longrightarrow \prod_{i=1}^{n} \mathrm{Proj}([\mathrm{X}_{1,i}, \mathrm{Y}_{1,i}])$ is finite and flat (cf. Lemma 3.82), the dimension of $(p_1)_K(X_K \cap C_i)$ is equal to the dimension of $X_K \cap C_i$. With (3.35) we conclude:

Lemma 3.93 *$(p_1)_K(X_K \cap C_i)$ (with the induced reduced scheme structure) is a reduced effective Cartier divisor of $\prod_{i=1}^{n} \mathrm{Proj}([\mathrm{X}_{1,i}, \mathrm{Y}_{1,i}])$ whose multidegree is componentwise $\leq (n! \cdot 2^{2(n-1)\cdot n}, \ldots, n! \cdot 2^{2(n-1)\cdot n})$.*

The subscheme

$$\bigcup_{i=1}^{n} \bigcup_{j=0}^{n-1} \sigma^j((p_1)_K(X_K \cap C_i))$$

of $\prod_{i=1}^{n} \mathrm{Proj}([\mathrm{X}_{1,i}, \mathrm{Y}_{1,i}])$ is $\mathrm{Gal}(K|k)$-invariant. It thus descends to a subscheme of $(\mathbb{P}^1_k)^n$; let B be this scheme.

Lemma 3.94

a) B is a reduced effective Cartier divisor whose multidegree is componentwise $\leq (n^2 \cdot n! \cdot 2^{2(n-1)\cdot n}, \ldots, n^2 \cdot n! \cdot 2^{2(n-1)\cdot n})$.

b) Let $Q \in (\mathbb{P}^1(\overline{k}))^n - B(k)$, and let Q' be any preimage of Q under p_1. Then the fiber $X_{p_2(Q')}$ is zero-dimensional.

c) There are at most $n^3 \cdot n! \cdot 2^{2(n-1)\cdot n} \cdot (q+1)^{n-1}$ points in $B(k)$.

Proof. Let A_i be a multihomogeneous polynomial defining $(p_1)_K(X_K \cap C_i)$. Then B is $V(\prod_{j=0}^{n-1} A_1 \cdots A_n)^{\mathrm{red}}$. The polynomial in question has a multidegree which is componentwise $\leq (n^2 \cdot n! \cdot 2^{2(n-1)\cdot n}, \ldots, n^2 \cdot n! \cdot 2^{2(n-1)\cdot n})$.

Statement b) follows immediately from the definition of B.

Statement c) follows from a) and the following lemma. □

Lemma 3.95 *Let H be an effective Cartier divisor of multidegree $\underline{d}$ in $(\mathbb{P}^1_k)^n$. Then*

$$\#H(k) \leq (\sum_{i=1}^{n} d_i) \cdot (q+1)^{n-1} .$$

Proof. It clearly suffices to show the result under the condition that all indices of the multidegree are positive.

We proceed with induction by n. For $n = 1$ the claim is that $\#H(k) \leq d_1$, and this is surely correct.

Now let H be defined by the polynomial $F(\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n) \in k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_n, \mathrm{Y}_n]$. Let us consider the projection to the first $n-1$ components: $(\mathbb{P}^1_k)^n \longrightarrow (\mathbb{P}^1_k)^{n-1}$ and the induced morphism $H \longrightarrow (\mathbb{P}^1_k)^{n-1}$. Now for every point $P = (P_1, \ldots, P_{n-1}) \in (\mathbb{P}^1_k)^{n-1}(k)$ for which $F(P_1, \ldots, P_{n-1}, \mathrm{X}_n, \mathrm{Y}_n)$ does not vanish, the fiber has degree d_n, thus in particular it contains at most d_n k-rational points. Let now C be a non-trivial coefficient of F regarded as a polynomial in $k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_{n-1}, \mathrm{Y}_{n-1}][\mathrm{X}_n, \mathrm{Y}_n]$. Then all points $P \in (\mathbb{P}^1_k)^{n-1}(k)$ for which $F(P_1, \ldots, P_{n-1}, \mathrm{X}_n, \mathrm{Y}_n)$ vanishes are contained in $V(C)$. Now C has multidegree $(d_1, \ldots, d_{n-1})$, and thus $\#V(C)(k) \leq (\sum_{i=1}^{n-1} d_i) \cdot (q+1)^{n-2}$ by induction. We conclude:

$$\begin{aligned} \#H(k) &\leq d_n \cdot (q+1)^{n-1} + \#V(C)(k) \cdot (q+1) \\ &\leq d_n \cdot (q+1)^{n-1} + (\textstyle\sum_{i=1}^{n-1} d_i) \cdot (q+1)^{n-1} \\ &= (\textstyle\sum_{i=1}^{n} d_i) \cdot (q+1)^{n-1} \end{aligned}$$

□

Given an element $P \in E(K)$, the decomposition algorithm succeeds when applied to P if and only if the fiber $X_{\varphi(P)_\otimes}$ is zero-dimensional and contains a k-rational point $(Q_1, \ldots, Q_n)$ such that there exist $P_1, \ldots, P_n \in E(K)$ with $\varphi(P_i) = Q_i$ and $\sum_i P_i = P$.

We want to derive a lower bound on the number of such elements $P \in E(K)$. More generally, given any subset M of $\{(P_1, \ldots, P_n) \in E(K)^n \mid$

$\varphi(P_i) \in \mathbb{P}^1(k)$ for all $i = 1, \ldots, n\}$, we want to derive a lower bound on the number of elements $P \in E(K)$ such that the decomposition algorithm succeeds and there exist $P_1, \ldots, P_n \in M$ with $\sum_i P_i = \pm P$.

Let us for this consider the commutative diagram of sets of k-valued points

$$\begin{array}{ccc} G(k) & \xrightarrow{\quad\rho\quad} & X(k) \\ {\scriptstyle\gamma}\uparrow & & \downarrow{\scriptstyle(p_1)_{|X}} \\ V^n(k) & \xrightarrow{\quad\tau\quad} & \prod_{i=1}^n \mathrm{Proj}(k[\mathrm{X}_{1,i}, \mathrm{Y}_{1,i}])(k) \end{array} ,$$

where the map $\gamma : V(k) \longrightarrow G(k)$ is induced by the graph morphism, that is, it is explicitly given by $(P_1, \ldots, P_n) \mapsto (P_1, \ldots, P_n, -\sum_i P_i)$, the map $\rho : G(k) \longrightarrow X(k)$ is induced by the morphism $G \longrightarrow X$ defined in Proposition 3.85, and the map $\tau : V^n(k) \longrightarrow \prod_{i=1}^n \mathrm{Proj}(k[\mathrm{X}_{1,i}, \mathrm{Y}_{1,i}])(k)$ is induced componentwise by the canonical morphism in diagram (3.32).

Note that under the scalar restriction functor and in the context of the algorithms $V(k)$ corresponds to $\{P \in E(K) \mid \varphi(P) \in \mathbb{P}^1(k)\}$, $G(k)$ corresponds to the set of tuples $(P_1, \ldots, P_n, P)$ with $\varphi(P_i) \in \mathbb{P}^1(k)$ and $P = -\sum_i P_i$, and $X(k)$ corresponds to the set of tuples $(Q_1, \ldots, Q_n, Q)$ with $Q_i \in \mathbb{P}^1(k)$ and $Q \in \mathbb{P}^1(K)$ and $S_{n+1}(Q_1, \ldots, Q_n, Q) = 0$. The map γ corresponds then to the map which is again given by $(P_1, \ldots, P_n) \mapsto (P_1, \ldots, P_n, -\sum_i P_i)$, and the maps ρ and τ correspond to the componentwise application of φ.

Let now $M \subseteq \{(P_1, \ldots, P_n) \in E(K)^n \mid \varphi(P_i) \in \mathbb{P}^1(k) \text{ for all } i = 1, \ldots, n\}$, and let $M_\circledcirc$ be the corresponding subset of $V(k)$. Then every element $P \in E(K)$ such that $\varphi(P)_\circledcirc \in \mathrm{Res}_k^K(\mathbb{P}^1_K)(k)$ is the image under p_2 of an element in $(\rho \circ \gamma)(M_\circledcirc) - p_1^{-1}(B(k))$ is an element as desired. (Indeed, if P is such an element, first the fiber $X_{\varphi(P_\circledcirc)}$ is zero-dimensional because $\varphi(P_\circledcirc) \in p_2(p_1^{-1}(B(k)))$ (cf. Lemma 3.94 b)), and second there exist $P_1, \ldots, P_n \in M$ with $\varphi(P_1 + \cdots + P_n) = \varphi(P)$, thus $P_1 + \cdots + P_n = \pm P$.)

We are thus interested in the cardinality of the set

$$p_2\big((\rho \circ \gamma)(M_\circledcirc) - p_1^{-1}(B(k))\big) .$$

For this we first derive a lower bound on

$$(\rho \circ \gamma)(M_\circledcirc) - p_1^{-1}(B(k)) .$$

The image of this set in $\prod_{i=1}^n \mathrm{Proj}(k[\mathrm{X}_{1,i}, \mathrm{Y}_{1,i}])(k)$ is contained in

$$\tau(M_\circledcirc) - B(k) .$$

As τ corresponds to the componentwise application of φ, we have $\#\tau(M_\circledcirc) \geq \frac{1}{2^n}\#M_\circledcirc = \frac{1}{2^n}\#M$.

With Lemma 3.94 c) we obtain:

$$\begin{aligned} & \#((\rho \circ \gamma)(V^n(k)) - p_1^{-1}(B(k))) \\ \geq\ & \#(\tau(M_{\circledcirc}) - B(k)) \qquad (3.36) \\ \geq\ & \tfrac{\#M}{2^n} - n^3 \cdot n! \cdot 2^{2(n-1)\cdot n} \cdot (q+1)^{n-1} \ . \end{aligned}$$

Now if an element Q in the set $p_2((\rho \circ \gamma)(V^n(k)) - p_1^{-1}(B(k)))$ is given, the fiber of $p_2(Q)$ under p_2 is zero-dimensional, and thus its degree is $n! \cdot 2^{(n-1)\cdot n}$ (see Remark 3.89 or consider the structure of the system in the decomposition algorithm in subsubsection 3.5.2.1). We therefore have the following proposition.

Proposition 3.96 *Let*

$$M \subseteq \{(P_1, \ldots, P_n) \in E(K)^n \mid \varphi(P_i) \in \mathbb{P}^1(k) \text{ for all } i = 1, \ldots, n\} \ .$$

Then the number of elements $P \in E(K)$ such that the decomposition algorithm succeeds and there exist $P_1, \ldots, P_n \in M$ with $P_1 + \cdots P_n = \pm P$ is

$$\geq \frac{\#M - n^3 \cdot 2^{2n^2-n} \cdot (q+1)^{n-1}}{n! \cdot 2^{n^2}} \ .$$

This proposition is crucial for the analysis of the algorithm for Theorem 5. For Theorem 4 we set the factor base equal to the full set $\{P \in E(K) \mid \varphi(P) \in \mathbb{P}^1(k)\}$ and we vary n and q.

As mentioned in Remark 3.79 for $\log_2(q) \geq 3n$ and n large enough we have $\#V(k) \geq \frac{q+1}{2}$, thus $\#V^n(k) \geq \frac{(q+1)^n}{2^n}$. With Proposition 3.96 we obtain that the number of elements $P \in E(K)$ such that there exist $P_1, \ldots, P_n \in E(K)$ with $\varphi(P_i) \in \mathbb{P}^1(k)$ and $\sum P_i = P$ is

$$\geq \frac{(q+1)^{n-1}}{n! \cdot 2^{n\cdot(n+1)}} \cdot (q + 1 - n^3 \cdot 2^{2n^2}) \ .$$

Let now $\epsilon > 0$. Then for n large enough this is

$$\geq \frac{q^{n-1}}{n! \cdot 2^{n\cdot(n+1)}} \cdot (q - \frac{1}{2} \cdot 2^{(2+\epsilon)\cdot n^2}) \ .$$

Then for $\log_2(q) \geq (2+\epsilon) \cdot n^2$ this is

$$\geq \frac{q^n}{n! \cdot 2^{n\cdot(n+1)+1}} \ .$$

Again for n large enough and $\log_2(q) \geq (2+\epsilon) \cdot n^2$ this is

$$\geq 2 \cdot q^{n-\frac{1}{2}} \ .$$

We therefore have:

Proposition 3.97 *Let $\epsilon > 0$. Then for n large enough and $(2+\epsilon) \cdot n^2 \leq \log_2(q)$ there are at least $2 \cdot q^{n-\frac{1}{2}}$ elements in $E(K)$ for which the decomposition algorithm succeeds.*

And this implies the main result for the analysis of the algorithm for Theorem 4:

Proposition 3.98 *Let $\epsilon > 0$. Then for n large enough and $(2+\epsilon) \cdot n^2 \leq \log_2(q)$ the following holds: Let $E/\mathbb{F}_{q^n}$ be an elliptic curve, and let φ be chosen such that Condition 3.45 holds. Then the probability that the decomposition algorithm succeeds if applied to a uniformly randomly distributed element in $E(\mathbb{F}_{q^n})$ is $\geq q^{-\frac{1}{2}}$.*

Bibliography

[Adl79] L. Adleman. A subexponential algorithm for the discrete logarithm problem with applications to cryptography. In *20th IEEE Found. Comp. Sci. Symp.*, pages 55–60, 1979.

[AL86] L. Adleman and H.W. Lenstra, Jr. Finding irreducible polynomials over finite fields. In *Proc. 18th ACM Symp. on Computing (STOC)*, pages 350–353, 1986.

[AT06] R. Avanzi and N. Thériault. Index Calculus for Hyperelliptic Curves. In H. Cohen and G. Frey, editors, *Handbook of Elliptic and Hyperelliptic Curve Cryptogrpahy*, chapter 21. Chapman & Hall/CRC, 2006.

[BCS91] P Bürgisser, M. Clausen, and M.A. Shokrollahi. *Algebraic Complexity Theory*. Springer-Verlag, 1991.

[BCSS98] L. Blum, F. Chucker, M. Shub, and S. Smale. *Complexity and real computation*. Springer-Verlag, 1998.

[BG04] J. Brawley and S. Gau. On the density of primitive elements for field extensions. preprint, 2004.

[BL94] J. Buchmann and H.W. Lenstra, Jr. Approximating rings of integers in number fields. *J. Théo. Nombres Bordx.*, 6:221–260, 1994.

[BLR80] S. Bosch, W. Lütkebohmert, and W. Raynaud. *Néron Models*. Springer-Verlag, 1980.

[BN74] A. Brill and M. Noether. Über die algebraischen Functionen und ihre Anwendung in der Geometrie. *Math. Ann.*, pages 269–310, 1874.

[BSS89] L. Blum, M. Shub, and S. Smale. On a theory of computation over the real numbers: NP completeness, recursive functions and unversal machines. *Bull. Amer. Math. Soc.*, 21(1), 1989.

[CF05] H. Cohen and G. Frey, editors. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. Chapman & Hall/CRC, 2005.

[Chi87] A. Chistov. Efficent factorization of polynomials over local fields (Russian). *Dokl. Akad. Nauk SSSR*, 293:1073–1077, 1987. English translation: *Soviet Math. Dokl.* 35 (1987), 434-438.

[Chi89] A. Chistov. The complexity of constructing the ring of integers of a global field (Russian). *Dokl. Akad. Nauk SSSR*, 306:1063 – 1067, 1989. English translation in *Soviet Math. Dokl.* 39 (1989), 597 - 600.

[Coh96] H. Cohen. *A Course in Computational Algebraic Number Theory*. Springer-Verlag, 1996.

[Cou01] J.-M. Couveignes. Algebraic groups and discrete logarithms. In *Public key cryptography and computational number theory*. de Gruyter, 2001.

[Die01] C. Diem. *A study on theoretical and practical aspects of Weil-restrictions of varieties*. PhD thesis, University of Essen, 2001.

[Die03] C. Diem. The GHS Attack in odd Characteristic. *J. Ramanujan Math. Soc.*, 18:1–32, 2003.

[Die06] C. Diem. An Index Calculus Algorithm for Plane Curves of Small Degree. In F. Hess, S. Pauli, and M. Pohst, editors, *Algorithmic Number Theory — ANTS VII*, LNCS 4076, pages 543 – 557, Berlin, 2006. Springer.

[dJ98] T. de Jong. An algorithm for computing the integral closure. *J. Symb. Comput.*, 23:273–277, 1998.

[DKT87] P. Domich, R. Kannan, and L. Trotter, Jr. Hermite normal form computations using modulo determinant arithmetic. *Mathematics of Operations Research*, 12:50–59, 1987.

[EG02] A. Enge and P. Gaudry. A general framework for subexponential discrete logarithm algorithms. *Acta. Arith.*, 102:83–103, 2002.

[Eis95] D. Eisenbud. *Commutative Algebra with a View Toward Algebraic Geometry*. Springer-Verlag, 1995.

[EK97] W. Eberly and E. Kaltofen. On randomized Lanczos algorithms. In W. Küchlin, editor, *Proceedings ISSAC 1997*, pages 176–183. ACM Press, 1997.

[Eng02] A. Enge. Computing discrete logarithms in high-genus hyperelliptic jacobians in provably subexponential time. *Math. Comp*, 71:729–742, 2002.

[Evd89] S. Evdokimov. Factorization of a solvable polynomial over finite fields and the generalized Riemann hypothesis (Russian). *Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov. (LOMI)*, 176, 1889. English translation: *J. Soviet Math.* 59 (1992), 842 - 849.

[Ful93] W. Fulton. *Introduction to Toric Varieties*. Princeton University Press, 1993.

[Gau00] P. Gaudry. An algorithm for solving the discrete log problem on hyperelliptic curves. In *Advances in Cryptology — EUROCRYPT 2000*, LNCS 1807, pages 19–34. Springer-Verlag, 2000.

[Gau04a] P. Gaudry. Index calculus for abelian varieties and the elliptic curve discrete logarithm problem. Available under http://eprint.iacr.org/2004/073, Mar. 4 2004.

[Gau04b] P. Gaudry. Index calculus for abelian varieties and the elliptic curve discrete logarithm problem. Available under http://www.loria.fr/∼gaudry/papers.html (as of May 4 2008), accepted for publication at J. Symbolic Comput., Oct. 26 2004.

[GHS02] P. Gaudry, F. Heß, and N. Smart. Constructive and destructive facets of Weil descent on elliptic curves. *J. Cryptology*, 15(19-46), 2002.

[GKZ94] I. Gelfand, M. Kapranov, and A. Zelevinsky. *Discriminants, Resultants, and Multidimensional Determinants*. Birkäuser, 1994.

[GLS01] G.-M. Greuel, C. Lossen, and M. Schulze. Three algorithms in Algebraic Geometry, Coding Theory, and Singularity Theory. In C. Ciliberto et al, editor, *Application of Algebraic Geometry to Coding Theory, Physics and Computation, Proceedings*, pages 161–194. Kluwer, 2001.

[Gop82] V. Goppa. Algebraic-geometric codes. *Izv. Akad. Nauk SSSR Ser. Mat*, 46(4):762–781, 896, 1982. English translation: *Math. USSR-Izv.* 21 (1983), 75-91.

[GR71] H. Grauert and R. Remmert. *Analytische Stellenalgebren*. Springer-Verlag, 1971.

[Gro61] A. Grothendieck. Eléments de Géométrie Algébrique III, Première Partie. *Publication Mathématiques*, 11, 1961.

[Gro67] A. Grothendieck. Eléments de Géométrie Algébrique IV, Quatrième Partie. *Publication Mathématiques*, 32, 1967.

[GTTD07] P. Gaudry, E. Thomé, N. Thériault, and C. Diem. A double large prime variation for small genus hyperelliptic index calculus. *Math. Comp.*, 76:475–492, 2007.

[Har77] R. Hartshorne. *Algebraic Geometry*. Springer-Verlag, 1977.

[Heß01] F. Heß. Computing Riemann-Roch spaces in algebraic function fields and related topics. *J. Symb. Comput.*, 11, 2001.

[Heß03] F. Heß. The GHS Attack Revisited. In E. Biham, editor, *Advances in Cryptology — EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 374–387. Springer-Verlag, 2003.

[Heß05] F. Heß. Computing relations in divisor class groups of algebraic curves over finite fields. preprint, ca. 2005.

[HI94] M.-D. Huang and D. Ierardi. Efficient Algorithms for the Riemann-Roch Problem and for Addition in the Jacobian of a Curve. *J. Symb. Comput.*, 18:519–539, 1994.

[HL02] K. Hensel and G. Landsberg. *Theorie der algebraischen Funktionen einer Variablen*. Teubner, 1902.

[HR83] M. Hellman and J. Reyneri. Fast computation of discrete logarithms in GF(q). In D. Chaum, R. Rivest, and A. Sherman, editors, *Advances in Cryptology — CRYPTO 1982*, pages 3–13. Plenum Press, 1983.

[Jac39] C.G. Jacobi. *Canon arithmeticus sive tabulae quibus exhibentur pro singulis numeris primis vel primorum potestatibus infra 1000 numeri ad datos indices et indices ad datos numeros pertinentes*. Berlin, 1839.

[KM04a] K. Khuri-Makdisi. Asymptotically fast group operations on Jacobians of general curves. available on the arXiv under math.NT/0409209 (version 2), 2004.

[KM04b] K. Khuri-Makdisi. Linear algebra algorithms for divisors on an algebraic curve. *Math. Comp.*, 73:333–357, 2004.

[KR89] E. Kani and M. Rosen. Idempotent relations and factors of Jacobians. *Math. Ann.*, 284:307–327, 1989.

[Kra22] M. Kraitchik. *Théorie des nombres (Tome I)*. Gauthier-Villars, 1922.

[KS91] E. Kaltofen and B. Saunders. On wiedemann's method for solving sparse linear systems. In H. Mattson, T. Mora, and T. Rao, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, volume 539 of *LNCS*, pages 29–38. Springer-Verlag, 1991.

[KS98] E. Kaltofen and V. Shoup. Subquadratic-time factoring of polynomials over finite fields. *Math. Comp.*, 67(223):1179–1197, 1998.

[Lan93] S. Lang. *Algebra (Third Edition)*. Addison-Wesley Publishing Company, 1993.

[LBR88] D. Le Brigand and J. Risler. Algorithme de Brill-Noether et codes de Goppa. *Bull. Soc. Math. France*, 116:231–253, 1988.

[LM91] A.K. Lenstra and M.S. Manasse. Factoring with two large primes (extended abstract). In *Advances in cryptology — EUROCRYPT 1990*, volume 473 of *LNCS*, pages 72–82. Springer-Verlag, 1991.

[LM94] A.K. Lenstra and M.S. Manasse. Factoring with two large primes. *Math. Comp*, 63:785–798, 1994.

[LP92] H.W. Lenstra and C. Pomerance. A rigorous time bound for factoring integers. *J. Amer. Math. Soc.*, 5, 1992.

[LW] A. Lauder and D. Wan. Counting points on varieties over finite fields of small characteristic. In J. Buhler and P. Stevenhagen, editors, *Algorithmic Number Theory: Lattices, Number Fields, Curves and Cryptography*. Cambridge University Press. forthcoming, available on the arXiv under math.NT/0612147.

[Mat89] H. Matsumura. *Commutative Ring Theory*. Cambridge University Press, 1989.

[Mil75] J. Miller. On Factorization, with a Suggested New Approach. *Math. Comp.*, 29:155–172, 1975.

[MS03] T. Mulders and A. Storjohann. On lattice reduction for polynomial matrices. *J. Symb. Comput.*, 35:377–401, 2003.

[Mum65] D. Mumford. *Geometric Invariant Theory*. Springer-Verlag, 1965.

[Nag04] K. Nagao. Improvement of Thériault Algorithm of Index Calculus of Jacobian of Hyperelliptic Curves of Small Genus. Cryptology ePrint Archive, Report 2004/161, http://eprint.iacr.org/2004/161, 2004.

[Neu91] J. Neukirch. *Algebraische Zahlentheorie*. Springer-Verlag, 1991.

[Noe84] M. Noether. Rationale Ausführung der Operationen in der Theorie der algebriaschen Functionen. *Math. Ann.*, 23:311–358, 1884.

[Odl84] A. Odlyzko. Discrete logarithms in finite fields and their cryptographic significance. In T. Beth, N. Cot, and I. Ingemarsson, editors, *Advances in Cryptology — EUROCRYPT 1984*, volume 209 of *LNCS*, pages 224–314, 1984.

[Pap94] C. Papadimitriou. *Computational Complexity*. Addison Wesley, 1994.

[Pil90] J. Pila. Frobenius maps of abelian varieties and fining roots of unity in finite fields. *Math. Comp.*, 55:745–763, 1990.

[Pil91] J. Pila. Counting points on curves over families in polynomial time. available on the arXiv under math.NT/0504570, 1991.

[Pol78] J. Pollard. Monte Carlo methods for index computations (mod p). *Math. Comp.*, pages 918–924, 1978.

[Pom87] C. Pomerance. Fast, rigorous factorization and discrete logarithm algorithms. In D. Johnson, T. Nishizeki, A. Nozaki, and H. Wolf, editors, *Discrete Algorithms and Complexity, Proceedings of the Japan US Joint Seminar, June 4-6, 1986, Kyoto, Japan*, pages 119–143, 1987.

[PS85] F. Preparata and M. Shamos. *Computational Geometry*. Springer-Verlag, 1985.

[RS62] J. Rosser and L. Schoenfeld. Approximate formulas for some functions of prime numbers. *Illinois J. Math.*, 6(64-94), 1962.

[Sch85] R. Schoof. Elliptic curves over finite fields and the compuation of square roots mod p. *Math. Comp.*, 44:483–494, 1985.

[Sch91] W. Schmidt. Construction and estimation of bases in function fields. *J. Number Th.*, 39:181–224, 1991.

[Sem98] I. Semaev. Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p. *Math. Comp.*, 67:353–356, 1998.

[Sem04] I. Semaev. Summation polynomials and the discrete logarithm problem on elliptic curves. available under http://eprint.iacr.org/2004/031, Feb. 2004.

[Ser79] J.-P. Serre. *Local Fields*. Springer, 1979.

[Sil86] J. Silverman. *The Arithmetic of Elliptic Curves*. Springer-Verlag, 1986.

[Sti93] H. Stichtenoth. *Algebraic Function Fields and Codes*. Springer-Verlag, 1993.

[SZ94] B. Sturmfels and A. Zelevinsky. Multigraded Resultants of Sylvester Type. *J. Algebra*, 163:115–127, 1994.

[Thé03] N. Thériault. Index calculus attack for hyperelliptic curves of small genus. In *Advances in Cryptology — ASIACRYPT 2003*, volume 2894 of *LNCS*, pages 75–92. Springer-Verlag, 2003.

[Vol94] E. Volcheck. Computing in the Jacobian of a Plane Algebraic Curve. In L. Adleman and M.-D. Huang, editors, *Algebraic Number Theory – ANTS I*, volume 877 of *LNCS*, pages 221–233. Springer-Verlag, 1994.

[Vol95] E. Volcheck. Addition in the Jacobian of a Curve over a Finite Field. Manuscript for a presentation given at the Oberwolfach institute, available under http://www.emilvolcheck.com, 1995.

[vzGG03] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge Unversity Press, 2003.

[Wie86] D. Wiedemann. Solving Sparse Linear Equations Over Finite Fields. *IEEE Trans. Inform. Theory*, 32(1):54–62, 1986.

[WM68] A.E. Western and J.C.P. Miller. *Tables of Indices and Primitive Roots*, volume 9. Royal Society Mathematical Tables, 1968. Published for the Royal Society at the Cambridge University Press.

[Zas67] H. Zassenhaus. Ein Algorithmus zur Berechnung einer Minimalbasis über gegebener Ordnung. In L Collarz, G. Meinardus, and H. Ungers, editors, *Funktionalanalysis, Approximationstheorie, Numerische Mathematik (Oberwolfach 1965)*, pages 90 – 103. Birkhäuser, 1967.

Zusammenfassung

In dieser Arbeit werden Berechnungsprobleme, die einen Bezug zu Divisoren und Divisorklassen auf Kurven haben, von einem komplexitätstheoretischen Standpunkt aus untersucht. Besondere Betonung liegt hierbei auf der Komplexität des diskreten Logarithmusproblems in Grad 0 Klassengruppen von Kurven über endlichen Körpern. Hierbei studieren wir die folgende allgemeine Fragestellung: Welche interessanten Resultate über erwartete Laufzeiten kann man erhalten, wenn man Bedingungen an das *Geschlecht* und den *Grundkörper* aber keine weiteren Bedingungen an die Kurven stellt und die erwartete Laufzeit entweder mittels q^g, wobei q die Körpergröße und g das Geschlecht ist, oder mittels der Kardinalität der Grad 0 Klassengruppe der Kurve ausdrückt?

Man beachte hier, dass, wenn E eine elliptische Kurve über einen Körper K ist, die Gruppe der rationalen Punkte $E(K)$ von E kanonisch isomorph zu $\mathrm{Cl}^0(E)$, der Grad 0 Klassengruppe, ist. Wir betrachten also insbesondere das diskrete Logarithmusproblem in in Gruppen rationaler Punkte elliptischer Kurven.

Außerdem wird in dieser Arbeit ein Beitrag zu den Grundlagen der algorithmischen Mathematik geleistet.

Es folgt eine Zusammenfassung der wesentliche Aspekte der einzelnen Kapitel der Arbeit.

Kapitel 1
Berechnungsprobleme, Berechnungsmodelle und Komplexität

Ausgangspunkt der Fragestellungen in Kapitel 1 ist die Beobachtung, dass man die folgenden drei Fragen beantworten muss, um ein Berechnungsproblem wohldefiniert zu machen:

- Auf welchem Berechnungsmodell ist das Problem basiert?
- Wie lautet das Komplexitätsmaß / die Kostenfunktion?

- Wie werden die betrachteten mathematischen Objekte dargestellt?

Das Kapitel beginnt mit einigen Definitionen, die es ermöglichen, über Berechnungsprobleme vom "mathematischen Inhalt" her zu sprechen, d.h. ohne eine zu große Betonung darauf zu legen, wie die betrachteten Objekte für Berechnungen dargestellt werden, und dabei gleichzeitig nicht zu unpräzise zu sein. Die grundlegende Idee der Definitionen ist, dass sich Darstellungen von Objekten wohlverhalten sollen bezüglich Isomorphismen aber nicht notwendigerweise bezüglich anderen Morphismen.

Die wesentliche Definition lautet dann wie folgt:

Seien $\mathcal{C}$ und $\mathcal{X}$ zwei große Gruppoide (d.h. Kategorien, in denen jeder Morphismus ein Isomorphismus ist). Dann ist eine Darstellung *von $\mathcal{C}$ durch $\mathcal{X}$ ein essenziell surjektiver partieller Funktor von $\mathcal{X}$ nach $\mathcal{C}$, dessen Definitionsbereich eine volle Unterkategorie von $\mathcal{X}$ ist.*

Es ist naheliegend, nach einen Berechnungsmodell zu fragen, dass die intuitive Idee von *bit-Komplexität* erfasst und dabei in effizienter Weise indirekte Adressierung erlaubt. Sicher bieten RAM-Modelle eine angemessene Antwort, wenn indirekte Adressierung gewünscht wird. Ein Problem ist jedoch, dass RAM-Modelle immer auf einer bestimmten Menge von Kommandos beruhen, und die Auswahl der Kommandos geht mit einer gewissen Willkürlichkeit einher. Um diese Willkürlichkeit zu überwinden, schlagen wir eine gewisse bit-orientierte Registermaschine, die wir *bit-orientierte Random Access Machine (bit-RAM)* nennen.

Eine intuitive Beschreibung dieser Maschine ist schnell zu geben: wie üblich besteht die Maschine aus Registern und einem Programm. Der Befehlssatz besteht aus den üblichen Befehlen LOAD, STORE, GOTO, IFGOTO, END, und zusätzlich gibt es für jede Turing Maschine T (in einem bestimmten Modell) einen Befehl c_T.

Außerdem definieren wir *randomisierte bit-RAMs*.

Der bit-orientierte Zugang ist jedoch nicht immer der angemessenste, um Berechnungsprobleme zu modellieren. Zum Beispiel könnte man, gegeben ein Körper k, nach der Komplexität der Multiplikation zweier Matrizen mit Einträgen in k fragen, gemessen in Körperoperationen. In diesem Fall scheint es nicht angemessen, zu verlangen, dass die Körperelemente durch bit-Strings dargestellt werden, was ohnehin nur möglich ist, wenn der Körper abzählbar ist. Stattdessen wünscht man, was in [BCS91] ein "makroskopischer Standpunkt, ausgehend von einen idealisierten Computer" genannt wird, einzunehmen.

In der Tat gibt es einige Modelle in der Literatur, die es erlauben, einen solchen "makroskopischen Standpunkt" einzunehmen. Hier sei zunächst das Modell der Rechenbäume erwähnt. Dieses Modell ist jedoch nicht uniform über verschiedene Eingabelängen, während wir ein uniformes Modell anstre-

ben, um die Idee eines Algorithmus zu modellieren. Ein anderes bekanntes Modell ist das Modell von Blum, Shub und Smale ([BSS89]), welches als ein algebraisches Turing-Maschinenmodell betrachtet werden kann. Wir sind jedoch an einem Registermodell interessiert. Ausgehend von dem bit-RAM Modell definieren wir so ein so ein Berechnungsmodell, das über einem festen Ring R arbeitet. Wir nennen die resultierenden Maschinen *R-RAMs* (wobei sich R auf den Ring bezieht).

Sei nun R ein kommutativer Ring. Wenn S eine R-Algebra ist, definiert jede R-RAM in natürlicher Weise eine S-RAM. Wenn wir betonen wollen, dass eine R-RAM Eingaben aus allen R-Algebren entgegennimmt, sprechen wir von einer *R-Algebra RAM*. Eine $\mathbb{Z}$-Algebra RAM nennen wir auch eine *(generische) Ring RAM*; wenn wir uns auf Eingaben über Körpern statt Ringen beschränken, sprechen wir auch von einer *generischen Körper RAM*. Für Kapitel 2 ist die folgende Variante der generischen Körper RAM besonders relevant: Wir reichern das generische Körper RAM Modell mit einem Kommando an, um p-te Einheitswurzeln in Charakteristik $p > 0$ zu berechnen.

Außerdem definieren wir noch *randomisierte generische Körper RAMs*. Hierfür reichern wir das generische Körper RAM Modell mit einem Kommando zur Berechnung von Zufallszahlen und einem Kommando zum Faktorisieren an. Das Kommando zum Faktorisieren ahmt dabei das Verhalten der Bekannten (Las Vegas) Algorithmen zum Faktorisieren von Polynomen über endlichen Körpern nach.

Zuletzt beschäftigen wir uns in diesem Kapitel mit algebraischer Komplexität und endlichen Algebren und als Anwendung hiervon mit grundlegenden Fragen, die mit Berechnungsproblemen und Körpererweiterungen zu tun haben.

Kapitel 2
Darstellungen und grundlegende Berechnungen

In diesem Kapitel diskutieren wir verschiedene Methoden, um die den Berechnungen zugrundeliegenden Objekte darzustellen: Kurven sowie Punkte, Divisoren und Divisorklassen auf Kurven. Weiterhin geben wir einige Resultate zu grundlegenden Berechnungen mit Divisoren, insbesondere zu Berechnungen in der Divisorgruppe und die Berechnung der sogenannten "Riemann-Roch-Räume" $L(D)$, sowie Anwendungen auf die Arithmetik in der Divisorklassengruppe, an. Wir beschränken uns hierbei auf Kurven über vollkommenen Körpern.

Nach einer Darstellung der in dieser Arbeit verwendeten Notationen behandeln wir zunächst die Darstellung endlicher separabler Körpererweite-

rungen für rechnerische Zwecke. Die wichtigste Aussage ist hier die folgende:

Gegeben eine endliche Körpererweiterung $\lambda|k$, dargestellt durch eine Multiplikationstabelle, kann man mittels eines randomisierten Algorithmus ein primitives Element von $\lambda|k$ (und sein Minimalpolynom) in einer erwarteten Anzahl von Körperoperationen berechnen, welche polynomiell beschränkt in $[\lambda : k]$ ist.

Danach behandeln wir die Darstellung von Kurven, wobei im dieser Arbeit eine Kurve – wenn keine andere Information gegeben ist – immer geometrisch irreduzibel, eigentlich und glatt ist. Wir stellen Kurven immer durch *ebene Modelle* dar; ein ebenes Modell einer Kurve ist eine möglicherweise singuläre Kurve, die birational zur ursprünglichen Kurve ist. Die ebenen Modelle stellen wir durch eine definierende homogene Gleichung dar.

Die folgenden beiden Sätze legen nahe, dass diese Form der Darstellung vernünftig ist. (Der erste Satz ist [Heß05, Theorem 56].)

Jede Kurve über einem endlichen Körper hat ein ebenes Modell von Grad $\mathcal{O}(g)$ (unabhängig von den endlichen Körpern).

Jede Kurve von Geschlecht ≥ 1 mit einem Divisor von Grad 1 über einem undendlichen Körper hat ein ebenes Modell von Grad höchstens $4g$.

Im größten Abschnitt des Kapitels behandeln wir die Darstellung von Punkten und Divisoren und hiermit zusammenhängende Berechnungsprobleme. Wir geben hierbei verschiedene Möglichkeiten an, abgeschlossene Punkte und Divisoren auf Kurven über vollkommenen Körpern darzustellen, ausgehend von einem ebenen Modell. Dabei gehen wir auch auf hiermit zusammenhängende algorithmische Aspekte wie z.B. die Arithmetik in der Divisorgruppe, die Berechnung von Hauptdivisoren und die Berechnung von Riemann-Roch-Räumen, ein.

Es gibt drei klassische Zugänge, um abgeschlossene Punkte und Divisoren auf Kurven darzustellen:

Der erste (und wohl auch intuitivste Zugang), um abgeschlossene Punkte darzustellen, beruht auf Koordinaten (in endlichen Erweiterungskörpern des Grundkörpers) von Punkten in einem ebenen Modell. Divisoren können dann dargestellt werden, indem man den Träger und die Koeffizienten angibt. Um jedoch für eine feste Kurve und ein festes singuläres ebenes Modell alle abgeschlossenen Punkte beschreiben zu können, benötigt man oft noch zusätzliche Information. Wenn nämlich das ebene Modell Singularitäten mit mehreren Zweigen hat, benötigt man weitere "lokale Information", um die Punkte über den Singularitäten zu beschreiben. Es ist einfach, solche lokale Information zu geben, wenn das ebene Modell nur gewöhnliche Singularitäten hat (d.h. wenn die Tangenten der lokalen Zweige alle verschieden sind). Weiterhin kann man in Charakteristik 0 oder in "großer" positiver Charakteristik abgeschnittene Newton-Puiseux Entwicklungen verwenden.

In kleiner positiver Charakteristik stößt man allerdings auf größere technische Schwierigkeiten.

Die zweite Methode basiert auf der *Idealtheorie* in Funktionenkörpern. Hier stellt man eine Kurve als eine Überlagerung der projektiven Geraden dar und definiert zwei Ordnungen in ihrem Funktionenkörper, eine "endliche" und eine "unendliche" Ordnung. Nun gibt es eine Bijektion zwischen der Divisorklassengruppe der Kurve und dem Produkt der Idealklassen der beiden Ordnungen. Man kann jeden Divisor durch das korrespondierende Paar von Idealen darstellen. Die Ideale kann man durch Modulbasen dargestellen (zum Beispiel durch Hermite Normalform-Basen). Wir nennen diese Darstellung *zusammengesetzte Idealdarstellung*. Alternativ kann man auch jeden Divisor als formale Summe von Primdivisoren darstellen, wobei jeder Primdivisor (= abgeschlossener Punkt) als Primideal einer der beiden Ordnungen dargestellt wird. Wir nennen diese Darstellung *freie Idealdarstellung*.

Man beachte, dass die so genannte "Mumford Darstellung" von so genannten semi-reduzierten Divisoren auf hyperelliptischen Kurven in imaginär quadratischer Darstellung ein Spezialfall der zusammengesetzten Idealdarstellung ist.

Drittens kann man effektive Divisoren durch lineare Unterräume von Riemann-Roch-Räumen von Divisoren genügend hohen Grades oder allgemeiner von Räumen globaler Schnitte invertierbarer Garben genügend hohen Grades darstellen. Wir nennen die entsprechende Darstellung *zusammengesetzte globale Darstellung*, und wiederum haben wir eine verwandte freie Darstellung.

Aufgrund der schon erwähnten technischen Schwierigkeiten gehen wir auf die Darstellung mittels Entwicklungen an singulären Punkten nicht ein und richten unser Hauptaugenmerk auf die idealtheoretische Darstellung. Wir geben eine ausführlichen Überblick über die Darstellung und hiermit zusammenhängende Berechnungsprobleme. Insbesondere gehen wir hierbei – in einem Anhang zu diesem Kapitel – auf die Berechnung von Hauptordnungen in Funktionenkörpern ein. Wir zeigen, dass diese Berechnung mit einem deterministischen Algorithmus (ohne Polynomfaktorisation) in einer Anzahl von Körper- und Bitoperationen durchgeführt werden kann, welche polynomiell im Grad des ebenen Modells ist.

Die idealtheoretische Darstellung erlaubt auch auf einfache Weise die Berechnung des Riemann-Roch Raums $L(D)$ zu einem Divisor D. Wir geben den entsprechenden Algorithmus an, der von F. Heß in seiner Doktorarbeit entwickelt wurde ([Heß01]). Es gibt einen Zusammenhang zwischen Heß' Algorithmus und "nicht-archimedischen Gittern". Wir diskutieren diesen Zusammenhang und gehen dabei auch auf eine Verbindung zur expliziten Bestimmung der Riemann-Roch-Räume ein, die von K. Hensel und G. Landsberg in [HL02] angegeben wurde.

Außerdem gehen wir auf die globale Darstellung und den rechnerischen Transfer zwischen den verschiedenen Darstellungen (inklusive der Koordinatendarstellung von Punkten, die über nicht-singulären Punkten im ebenen Modell liegen) ein.

Schließlich kommen wir zur Darstellung der Divisorklassen und Berechnungen in der Klassengruppe. Hierbei sind die folgende Definitionen grundlegend:

Sei $\mathcal{C}$ eine Kurve und D_0 ein fester Divisor auf $\mathcal{C}$ von Grad ≥ 1. Sei $\tilde{D}$ ein effektiver Divisor auf $\mathcal{C}$. Dann ist $\tilde{D}$ *(maximal) reduziert entlang* D_0, wenn das lineare System $|\tilde{D} - D_0|$ leer ist.

Sei nun D ein Divisor auf $\mathcal{C}$ und $\tilde{D}$ ein entlang D_0 reduzierter effektiver Divisor mit $D \sim \tilde{D} + rD_0$ für ein $r \in \mathbb{Z}$. Dann heißt $\tilde{D}$ eine *Reduktion* von D entlang D_0.

Die Reduktionen von D entlang D_0 bilden ein nicht-leeres vollständiges Linearsystem. Ferner ist die Reduktion eindeutig, wenn D_0 den Grad 1 hat.

Um die Elemente der Divisorklassengruppe darzustellen, fixieren wir so einen Divisor D_0 und stellen dann eine beliebige Divisorklasse a mittels eines reduzierten Divisors $\tilde{D}$ und einer ganzen Zahl r mit $[\tilde{D}] - r[D_0] = a$ dar.

Der kanonische Divisor hat Grad $2g - 2$. Wenn man nun diesen Divisor (oder einen Divisor kleineren Grades) für die Darstellung verwendet und Divisoren in zusammengesetzter Idealdarstellung darstellt, erhält man die folgende Aussage:

Die Arithmetik in der Divisorklassengruppe kann in einer Anzahl von Körper- und Bitoperation durchgeführt werden, die polynomiell beschränkt im Grad des ebenen Modells ist.

Kurven über endlichen Körpern haben immer einen Divisor von Grad 1. Wir zeigen, wie man einen solchen Divisor effizient konstruieren kann. Für das 3. Kapitel ist nun die folgende Aussage grundlegend:

Wir betrachten Kurven über endlichen Körpern. Wie immer stellen wir Kurven durch ebene Modelle dar, und definieren d als den Grad des ebenen Modells. Wir stellen Divisorklassen durch entlang einem Divisor D_0 von Grad 1 und mit einer Höhe, welche polynomiell beschränkt in d ist, reduzierte Divisoren dar. Außerdem stellen wir Divisoren in freier Idealdarstellung *dar. Dann kann die Arithmetik in den Grad 0 Klassengruppen mittels eines randomisieren Algorithmus in einer erwarteten Zeit durchgeführt werden, welche polynomiell beschränkt in d und* $\log(q)$ *ist, wobei q die Größe des Grundkörpers ist. Wenn die Kurven durch ebene Modelle von Grad $\mathcal{O}(g)$ dargestellt werden, kann man mittels eines randomisierten Algorithmus die Arithmetik in den Grad 0 Klassengruppen in einer Zeit durchführen, die polynomiell beschränkt in g und* $\log(q)$ *ist; man kann dann mit einem randomisieren Algorithmus die Arithmetik in einer erwarteten Zeit durchführen,*

die polynomiell beschränkt in $\log(\#\,\mathrm{Cl}^0(\mathcal{C}))$ *ist, wobei* $\mathcal{C}$ *die Kurve ist.*

Kapitel 3
Berechnen diskreter Logarithmen

Kapitel 3 ist das Herz dieser Arbeit und beinhaltet die Berechnung von diskreten Logarithmen in Grad 0 Klassengruppen von Kurven, inklusive der Gruppen rationaler Punkte elliptischer Kurven, über endlichen Körpern.

Alle Resultate über das diskrete Logarithmusproblem in dieser Arbeit basieren auf der *Index Calculus Methode*. Grob gesagt besteht die Index Calculus Methode aus dem Folgenden: Man fixiert eine Menge von Primdivisoren $\mathcal{F}$, genannt *Faktorbasis*, und sucht nach *Relationen* zwischen den Faktorbasiselementen und den Eingabeelementen. Wenn man genug Relationen hat, versucht man, den gesuchten diskreten Logarithmus mittels linearer Algebra zu finden.

Wir geben hier einen groben Überblick über die Variante von Index Calculus, auf der dieses Kapitel beruht:

Seien eine Kurve $\mathcal{C}$ über einem endlichen Körper $\mathbb{F}_q$ sowie $a, b \in \mathrm{Cl}^0(\mathcal{C})$ mit $b \in \langle a \rangle$ gegeben, wobei die Kurve und die Klassen a, b dargestellt werden, wie in Kapitel 2 beschrieben. Wir setzen der Einfachheit halber voraus, dass die Grad 0 Klassengruppe prime Gruppenordnung hat und von a erzeugt wird. Ferner setzen wir voraus, dass die Gruppenordnung $N := \#\,\mathrm{Cl}^0(\mathcal{C})$ bekannt ist.

Sei $x \in \{0, \ldots, N-1\}$ mit $x \cdot a = b$ der unbekannte diskrete Logarithmus, den wir berechnen wollen.

Man wählt zuerst die Faktorbasis $\mathcal{F}$, sagen wir $\mathcal{F} = \{F_1, \ldots, F_k\}$ mit paarweise verschiedenen Primdivisoren F_j. Außerdem wählt man einen Divisor D_1 von Grad 1.

Dann erzeugt man *Relationen* zwischen den Eingabeelementen a, b, den Faktorbasiselementen und D_1:

$$\sum_j r_{i,j}[F_j] - (\sum_j r_{i,j} \deg(F_j)) \cdot [D_1] = \alpha_i a + \beta_i b$$

mit $r_{i,j}, \alpha_i, \beta_i \in \mathbb{Z}/N\mathbb{Z}$.

Es seien nun $k+1$ Relationen erzeugt; sei $R = ((r_{i,j}))_{i,j} \in \mathbb{Z}^{(k+1)\times k}$ die *Relationenmatrix*, d.h. die Matrix, deren Zeilen den Koeffizienten entsprechen, die in den Relationen vorkommen.

Nun ist der Links-Kern der Matrix R nicht-trivial. Man berechnet einen nicht-trivialen Zeilenvektor $\underline{\gamma} \in (\mathbb{Z}/N\mathbb{Z})^{1\times(k+1)}$ mit $\underline{\gamma}R = 0$. Man sieht nun leicht, dass

$$\sum_i \gamma_i \alpha_i a + \sum_i \gamma_i \beta_i b = 0\ .$$

Wenn nun $\sum_i \gamma_i\beta_i \neq 0$, dann haben wir mit $\xi := (\sum_i \gamma_i\alpha_i)(\sum_i \gamma_i\beta_i)^{-1} \in \mathbb{Z}/N\mathbb{Z}$

$$\xi \cdot a = b \in \mathrm{Cl}^0(\mathcal{C}) \ .$$

Das bedeutet, dass der eindeutige Repräsentant $x \in \{0, \ldots, N-1\}$ von ξ der gesuchte diskrete Logarithmus von b bezüglich a ist.

Nach einer Einleitung geben wir im 2. Abschnitt des Kapitels einen "allgemeinen Index Calculus Algorithmus" für das diskrete Logarithmusproblem in Grad 0 Klassengruppen von Kurven. Dieser Algorithmus ist eine Variante eines Algorithmus von A. Enge und P. Gaudry ([EG02]). Eine Besonderheit dieses Algorithmus ist, dass man den diskreten Logarithmus mit großer Wahrscheinlichkeit berechnen kann, wenn die Relationensuche beendet ist. Der Algorithmus basiert auf fünf Unterroutinen für

a) Berechnung der Ordnung der Grad 0 Klassengruppe

b) Faktorisierung ganzer Zahlen

c) Konstruktion einer Faktorbasis und Vorberechnung

d) Relationenerzeugung

e) dünne lineare Algebra.

Im weiteren Verlauf des Kapitels geben wir dann spezielle Unterroutinen für a), c) und d) und für spezielle Klassen von Kurven an. Für Unterroutinen b) und e) verwenden wir (respektive) den Algorithmus von Lenstra und Pommerance ([LP92]) sowie einen Algorithmus, der auf Wiedemanns Algorithmus ([Wie86]) beruht.

Doppelt große Primvariation ist eine Möglichkeit, die Relationensuche zu beschleunigen. Hier geht man wie folgt vor: Zusätzlich zur Faktorbasis $\mathcal{F}$ definiert man eine weitere Menge von Primdivisoren $\mathcal{L}$, der Menge der so genannten *großen Primdivisoren*, die disjunkt zu $\mathcal{F}$ ist. Man betrachtet dann Relationen mit höchstens zwei großen Primdivisoren. Die grundlegende Idee ist wie folgt: Solche Relationen werden wie folgt in einem Graphen auf der Menge $\mathcal{L} \dot\cup \{*\}$ abgespeichert: Relationen mit zwei großen Primdivisoren P, Q werden mittels einer Kante zwischen P und Q abgespeichert, und Relationen mit einem großen Primdivisor P werden durch eine Kante zwischen $*$ und P abgespeichert. Zykel in dem Graph führen dann zu "kombinierten Relationen" zwischen den Eingabeelementen und den Faktorbasiselementen.

Wir gehen in unseren Algorithmen allerdings etwas anders vor: Wir konstruieren nicht einen vollen Graphen sondern nur einen Baum, wobei wir zusätzlich noch diesen Baum in "Abschnitten" konstruieren. Bei Konstruktion eines Abschnitts betrachten wir nur solche Relationen, die eine Kante ergeben, die mit dem vorherigen Abschnitt verbunden ist.

Die gesamte Konstruktion des Baumes findet in der Unterroutine c) statt. In d) werden dann neue Relationen erzeugt, die zunächst auch Elemente aus $\mathcal{L}$ enthalten dürfen. Mittels des Baums werden dann diese "großen Primdivisoren" eliminiert.

Mittels dieser Methode zeigen wir in Abschnitt 3 das folgende Theorem:

Theorem 1 *Es sei eine natürliche Zahl $g \geq 2$ fixiert. Dann kann das diskrete Logarithmusproblem in den Grad 0 Klassengruppen von Kurven von Geschlecht g über endlichen Körpern mittels eines randomisierten Algorithmus in einer erwarteten Zeit von*

$$\tilde{\mathcal{O}}(q^{2-\frac{2}{g}})$$

gelöst werden, wobei $\mathbb{F}_q$ der Grundkörper der Kurve ist.

Für dieses Resultat wählen wir den Divisor D_1 als einen $\mathbb{F}_q$-rationalen Punkt P_0, die Faktorbasis als eine Teilmenge von $\mathcal{C}(\mathbb{F}_q) - \{P_0\}$ mit $\lceil q^{1-\frac{1}{g}} \rceil$ Elementen und $\mathcal{L} := \mathcal{C}(\mathbb{F}_q) - (\mathcal{F} \cup \{P_0\})$.

Neben dem Index Calculus Algorithmus beruht dieses Resultat auch auf einem effizienten Algorithmus, der mit hoher Wahrscheinlichkeit ein kleines Erzeugendensystem berechnet. Allerdings können wir nicht überprüfen, dass es sich tatsächlich um ein Erzeugendensystem handelt. Deshalb gehen wir wie folgt vor: Wir wenden diesen Algorithmus an, um ein "potenzielles Erzeugendensystem" zu berechnen, und dann wenden wir den Index Calculus Algorithmus an. Wenn der Index Calculus Algorithmus nicht in einer vorgegebenen Zeit terminiert, starten wir den ganzen Prozess von vorne.

Im nächsten Abschnitt zeigen wir dann ausgehend von dem obigen Theorem die folgenden beiden Theoreme.

Theorem 2 *Es sei eine natürliche Zahl $g_0 \geq 2$ fixiert. Dann kann das diskrete Logarithmusproblem in den Grad 0 Klassengruppen von Kurven von Geschlecht $\geq g_0$ über endlichen Körpern mittels eines randomisierten Algorithmus in einer erwarteten Zeit von*

$$\tilde{\mathcal{O}}((q^g)^{\frac{2}{g_0}(1-\frac{1}{g_0})})$$

gelöst werden, wobei $\mathbb{F}_q$ der Grundkörper der Kurve ist.

Theorem 3 *Es sei eine natürliche Zahl $g_0 \geq 2$ fixiert. Dann kann das diskrete Logarithmusproblem in den Grad 0 Klassengruppen von Kurven $\mathcal{C}/\mathbb{F}_q$ von Geschlecht $\geq g_0$ mittels eines randomisierten Algorithmus in einer erwarteten Zeit von*

$$\tilde{\mathcal{O}}((\#\,\mathrm{Cl}^0(\mathcal{C}))^{\frac{2}{g_0}(1-\frac{1}{g_0})})$$

gelöst werden.

Wenn man Theorem 3 mit $g_0 := 3$ anwendet erhält man:

Man kann das diskrete Logarithmusproblem in den Grad 0 Klassengruppen von Kurven $\mathcal{C}/\mathbb{F}_q$ von Geschlecht mindestens 3 in einer erwarteten Zeit von

$$\tilde{\mathcal{O}}((\#\operatorname{Cl}^0(\mathcal{C}))^{\frac{4}{9}})$$

lösen.

Im Gegensatz hierzu haben "generische Methoden" für jede Folge von Kurven, deren Gruppenordnung durch eine Primzahl der Größe $\Theta(\#\operatorname{Cl}^0(\mathcal{C}))$ teilbar ist, eine erwartete Laufzeit von $\Omega(\#\operatorname{Cl}^0(\mathcal{C})^{\frac{1}{2}})$.

Um diese beiden Theoreme zu erhalten, gehen wir wie folgt vor: Für kleines Geschlecht wenden wir einen Algorithmus für Theorem 1 an. Für großes Geschlecht ist der Algorithmus viel einfacher: Es ist ein "einfacher" Index Calculus Algorithmus ohne große Primvariation. Die Analyse für großes Geschlecht beruht auf einer Aussage aus [Heß05].

Im letzten Abschnitt zeigen wir die folgenden beiden Theoreme:

Theorem 4 *Es sei $\epsilon > 0$. Dann kann das diskrete Logarithmusproblem in den Gruppen rationaler Punkte von elliptischen Kurven über endlichen Körpern $\mathbb{F}_{q^n}$ mit $(2+\epsilon) \leq \log_2(q)$ mittels eines randomisierten Algorithmus in einer erwarteten Zeit gelöst werden, welche polynomiell beschränkt in q ist.*

Theorem 5 *Es sei eine natürliche Zahl $n \geq 2$ fixiert. Dann kann das diskrete Logarithmusproblem in den Gruppen rationaler Punkte von elliptischen Kurven über endlichen Körpern $\mathbb{F}_{q^n}$ mittels eines randomisierten Algorithmus in einer erwarteten Zeit von*

$$\tilde{\mathcal{O}}(q^{2-\frac{2}{n}})$$

gelöst werden.

Theorem 4 hat das folgende offensichtliche Korollar:

Sei $\epsilon > 0$, und sei $a > 2+\epsilon$. Dann kann das diskrete Logarithmusproblem in den Gruppen rationaler Punkte von elliptischen Kurven über endlichen Körpern $\mathbb{F}_{q^n}$ mit $(2+\epsilon) \cdot n^2 \leq \log_2(q) \leq a \cdot n^2$ in einer erwarteten Zeit von

$$e^{\mathcal{O}(1)\cdot(\log_2(q^n))^{2/3}}$$

gelöst werden.

In der Tat erhält man eine erwartete Laufzeit, welche polynomiell beschränkt in

$$q = 2^{\log_2(q)} = 2^{(\log_2(q))^{(1+1/2)\cdot 2/3}} \leq 2^{(\sqrt{a}\cdot n \log_2(q))^{2/3}}$$

ist.

Mit diesem Korollar wird zum ersten Mal gezeigt, dass es Folgen endlicher Körper wachsender Größe gibt, über denen das elliptische Kurven diskrete Logarithmusproblem in einer erwarteten Zeit gelöst werden kann, welche subexponententiell in der Eingabelänge (oder im Logarithmus der Körpergröße bzw. der Gruppenordnung) ist.

Im Algorithmus zu Theorem 4 ist die Faktorbasis wie folgt gegeben:

Es wird eine Überlagerung $\varphi : E \longrightarrow \mathbb{P}^1_{\mathbb{F}_{q^n}}$ von Grad 2 mit $\varphi \circ [-1] = \varphi$ fixiert. Dann ist die Faktorbasis

$$\mathcal{F} := \{P \in E(\mathbb{F}_{q^n}) \mid \varphi(P) \in \mathbb{F}_q\} .$$

Die Relationensuche beruht auf einem "Zerlegungsalgorithmus", der auf dem Lösen polynomieller Gleichungssysteme beruht. Diese Gleichungssysteme beruhen auf einer homogenisierten Variante der von I. Semaev eingeführten *Summationspolynome*. Diese Polynome sind durch die folgende Aussage definiert:

Es sei E eine elliptische Kurve über einem Körper k und $\varphi : E \longrightarrow \mathbb{P}^1_k$ eine Überlagerung von Grad 2 mit $\varphi \circ [-1] = \varphi$. Sei $m \in \mathbb{N}$ mit $m \geq 2$. Dann gibt es ein bis auf Multiplikation mit einer nicht-trivialen Konstanten eindeutig bestimmtes multihomogenes irreduzibles Polynom $S_{\varphi,m} \in k[\mathrm{X}_1, \mathrm{Y}_1, \ldots, \mathrm{X}_m, \mathrm{Y}_m]$ so dass für alle $P_1, \ldots, P_m \in E(\overline{k})$ gilt: $S_{\varphi,m}(\varphi(P_1), \ldots, \varphi(P_m)) = 0 \longleftrightarrow \exists \epsilon_1, \ldots, \epsilon_m \in \{1, -1\} : \epsilon_1 P_1 + \cdots + \epsilon_m P_m = O$. Das Polynom $S_{\varphi,m}$ hat Multigrad $(2^{m-2}, \ldots, 2^{m-2})$.

Diese Polynome werden wie folgt eingesetzt:

Zu einer festen Überlagerung φ wie oben, einem Punkt $P \in E(\mathbb{F}_{q^n})$ und einer $\mathbb{F}_q$-Basis von $\mathbb{F}_{q^n}$ definieren wir wie folgt Polynome $S^{(1)}, \ldots, S^{(n)} \in \mathbb{F}_q[X_1, Y_1, \ldots, X_n, Y_n]$:

$$\sum_{j=1}^{n} b_j S^{(j)} = S_{\varphi,n+1}(X_1, Y_1, \ldots, X_m, Y_m, \varphi(P)) .$$

Für $Q_1, \ldots, Q_n \in \mathbb{P}^1(\mathbb{F}_q)$ sind nun die folgenden beiden Aussagen äquivalent:

- Es gibt $P_1, \ldots, P_n \in E(\overline{\mathbb{F}}_q)$ so dass $P_1 + \cdots + P_n = P$ und $x(P_i) = Q_i$ für $i = 1, \ldots, n$.
- Für alle $j = 1, \ldots, n$ ist $S^{(j)}(Q_1, \ldots, Q_n) = 0$, d.h. $(Q_1, \ldots, Q_n)$ ist ein $\mathbb{F}_q$-rationaler Punkt von $V(S^{(1)}, \ldots, S^{(n)}) \subseteq (\mathbb{P}^1_{\mathbb{F}_q})^n$.

Unter einem *Zerlegungsalgorithmus* verstehen wir nun einen Algorithmus für das folgende Berechnungsproblem: Gegeben q, n, eine $\mathbb{F}_q$-Basis $b_1, \ldots, b_n$, $E/\mathbb{F}_{q^n}$ und eine Überlagerung $\varphi : E \longrightarrow \mathbb{P}^1_{\mathbb{F}_{q^n}}$ wie oben, bestimme man, ob

das Unterschema $V(S^{(1)}, \ldots, S^{(n)}) \subseteq (\mathbb{P}^1_{\mathbb{F}_q})^n$ 0-dimensional ist, und wenn dies der Fall ist, bestimme man alle seine $\mathbb{F}_q$-rationalen Punkte!

Der schwierigste Teil der Analyse des Algorithmus ist nun, für geeignetes φ eine untere Schranke für die Anzahl der Punkte $P \in E(\mathbb{F}_q)$ anzugeben, für die $V(S^{(1)}, \ldots, S^{(n)})$ 0-dimensional ist und es $P_1, \ldots, P_n$ wie soeben beschrieben gibt.

Hierfür verwenden wir insbesondere Schnitttheorie in Produkten projektiver Geraden. Es folgt ein kurzer Einblick in den Beweis.

Sei $\mathrm{Res}^{\mathbb{F}_{q^n}}_{\mathbb{F}_q}(\mathbb{P}^1_{\mathbb{F}_{q^n}})$ die Weil Restriktion von $\mathbb{P}^1_{\mathbb{F}_{q^n}}$ bezüglich der Erweiterung $\mathbb{F}_{q^n}|\mathbb{F}_q$; dies ist eine n-dimensionale Varietät über $\mathbb{F}_q$ so dass es insbesondere eine kanonische Bijektion $\mathbb{P}^1(\mathbb{F}_{q^n}) \simeq \mathrm{Res}^{\mathbb{F}_{q^n}}_{\mathbb{F}_q}(\mathbb{P}^1_{\mathbb{F}_{q^n}})(\mathbb{F}_q)$ gibt.

Für $Q \in \mathbb{P}^1(\mathbb{F}_{q^n})$ sei $Q_\circledcirc$ der entsprechende $\mathbb{F}_q$-rationale Punkt in $\mathrm{Res}^{\mathbb{F}_{q^n}}_{\mathbb{F}_q}(\mathbb{P}^1_{\mathbb{F}_{q^n}})$. Außerdem seien $p_1 : (\mathbb{P}^1_{\mathbb{F}_q})^n \times \mathrm{Res}^{\mathbb{F}_{q^n}}_{\mathbb{F}_q}(\mathbb{P}^1_{\mathbb{F}_{q^n}}) \longrightarrow (\mathbb{P}^1_{\mathbb{F}_q})^n$ und $p_2 : (\mathbb{P}^1_{\mathbb{F}_q})^n \times \mathrm{Res}^{\mathbb{F}_{q^n}}_{\mathbb{F}_q}(\mathbb{P}^1_{\mathbb{F}_{q^n}}) \longrightarrow \mathrm{Res}^{\mathbb{F}_{q^n}}_{\mathbb{F}_q}(\mathbb{P}^1_{\mathbb{F}_{q^n}})$ die beiden Projektionen.

Wir definieren ein gewisses Unterschema X von $(\mathbb{P}^1_{\mathbb{F}_q})^n \times \mathrm{Res}^{\mathbb{F}_{q^n}}_{\mathbb{F}_q}(\mathbb{P}^1_{\mathbb{F}_{q^n}})$ mit der folgenden Eigenschaft: Für $P \in E(\mathbb{F}_{q^n})$ ist die Faser $X_{Q_\circledcirc}$ kanonisch isomorph zu $V(S^{(1)}, \ldots, S^{(n)})$.

Es wesentlicher Aspekt des Beweises ist es, die Anzahl der $\mathbb{F}_q$-rationalen Punkte in den *nicht-nulldimensionalen Fasern* unter p_2 nach oben abzuschätzen.

Hierfür bestimmen wir einen effektiven Cartierdivisor B in $(\mathbb{P}^1_{\mathbb{F}_q})^n$ so dass alle nicht-nulldimensionalen Fasern bezüglich p_2 in $p_1^{-1}(B)$ enthalten sind. Mittels Schnitttheorie können wir den Multigrad von B nach oben abschätzen. Das gewünschte Ergebnis erhalten wir dann, weil p_1 eine flache Überlagerung eines bekannten Grades ist.

Auch der Algorithmus für Theorem 5 beruht auf dem Zerlegungsalgorithmus, und zusätzlich verwenden wir eine doppelt große Primvariation. Bis auf die Verwendung des Zerlegungsalgorithmus ist dieser Algorithmus sehr ähnlich zu dem für Theorem 1.

Bezug zu anderen Arbeiten

Wir erwähnen noch kurz, wie sich diese Arbeit von verwandten Arbeiten abgrenzt.

Die wesentlichen neuen Resultate sind die Theoreme 1 – 5 oben, und alle diese Theoreme sind neu.

Für hyperelliptische Kurven in imaginär quadratischer Darstellung mit zyklischer Grad 0 Klassengruppe wurde Theorem 1 in [GTTD07] bewiesen.

In einer Arbeit von P. Gaudry ([Gau04]) befindet sich auch ein “Theo-

rem", in der die Aussage von Theorem 5 behauptet wird. Jedoch wird kein Beweis der Aussage sondern lediglich eine knappe heuristische Argumentation gegeben.

Wie betrachten die Aussagen von Kapitel 2 als "mehr oder weniger bekannt" unter Experten. Grundlegende Ideen befinden sich in der Arbeit [Heß01] von F. Heß sowie im Buch [Coh96] über Algorithmen für Zahlkörper von H. Cohen. Wir möchten jedoch betonen, dass, obwohl die grundlegenden Techniken dieses Kapitels schon vorher bekannt waren, einige wesentliche Aussagen noch nicht in der Literatur erschienen sind. Wir erwähnen hier beispielhaft die effiziente Transformation zwischen den verschiedenen Darstellungen von Divisoren, die wir diskutieren, sowie die effiziente faktorisierungsfreie Berechnung der Maximalordnung.

Kapitel 1 besteht hauptsächlich aus Definitionen, wobei einige vollkommen neu und andere Varianten von Definitionen in der Literatur sind.

Wir kommen abschließend auf die zu Beginn gestellte allgemeine Frage zum diskreten Logarithmusproblem in Grad 0 Klassengruppen von Kurven über endlichen Körpern zurück. Neben den soeben beschriebenen neuen Resultaten sind uns die folgenden Resultate zu der Frage bekannt:

Mit dem baby-step-giant-step Algorithmus und Resultaten über Arithmetik in Klassengruppen in Kapitel 2 dieser Arbeit kann das diskrete Logarithmusproblem für Kurven $\mathcal{C}/\mathbb{F}_q$ in einer Zeit von $\tilde{O}(\#\operatorname{Cl}^0(\mathcal{C})^{\frac{1}{2}})$ gelöst werden. Mit der Schranke $\#\operatorname{Cl}^0(\mathcal{C}) \leq (\sqrt{q}+1)^{2g}$ erhält man auch obere Schranken mittels q^g anstatt mittels $\#\operatorname{Cl}^0(\mathcal{C})$.

Neben diesen Resultaten ist uns nur ein weiteres (bewiesenes) Resultat bekannt. Es ist das Resultat über "Kurven großen Geschlechts" von F. Heß in ([Heß05]). Um dieses Resultat zu formulieren, definieren wir die übliche Komplexitätsfunktion

$$L_N[\alpha, c] := e^{c \cdot \log(N)^{\alpha} \cdot (\log\log(N))^{1-\alpha}}$$

für Parameter $\alpha \in (0,1)$ und $c > 0$. Dann ist das Resultat wie folgt:

Wir betrachten eine Klasse von Kurven über endlichen Körpern so dass $\log(q) \in o(g\log(q))$*, wobei wie üblich* q *die Körpergröße und* g *das Geschlecht ist. Dann kann für jedes* $\epsilon > 0$ *das diskrete Logarithmusproblem in den Grad 0 Klassengruppen solcher Kurven in einer erwarteten Zeit von*

$$L_{q^g}[\frac{1}{2}, 32^{\frac{1}{2}} + \epsilon]$$

gelöst werden.

Ähnlich zum vorherigen Resultat kann man mit der Ungleichung $(\sqrt{q}-1)^{2g} \leq \#\operatorname{Cl}^0(\mathcal{C})$ auch obere Schranken mittels $\#\operatorname{Cl}^0(\mathcal{C})$ erhalten.