Modular composition via factorization

Joris van der Hoeven, Grégoire Lecerf

Modular composition via factorization

JORIS VAN DER HOEVEN[a], GRÉGOIRE LECERF[b]

Laboratoire d'informatique de l'École polytechnique, CNRS UMR 7161
École polytechnique
91128 Palaiseau Cedex, France

a. *Email:* vdhoeven@lix.polytechnique.fr
b. *Email:* lecerf@lix.polytechnique.fr

Preprint version, February 6, 2017

Modular composition is the problem to compute the composition of two univariate polynomials modulo a third one. For polynomials with coefficients in a finite field, Kedlaya and Umans proved in 2008 that the theoretical bit complexity for performing this task could be made arbitrarily close to linear. Unfortunately, beyond its major theoretical impact, this result has not led to practically faster implementations yet. In this article, we explore particular cases of moduli over finite fields for which modular composition turns out to be cheaper than in the general case. In the most favourable cases, our algorithms achieve quasi-linear costs.

1. INTRODUCTION

Let $\mathbb{K}$ be an effective field, and let f, g, h be polynomials in $\mathbb{K}[x]$. The problem of *modular composition* is to compute $f \circ g$ modulo h. Modular composition is an important problem in complexity theory because of its applications to polynomial factorization [27, 28, 29]. It also occurs very naturally whenever one wishes to perform polynomial computations over $\mathbb{K}$ inside an algebraic extension of $\mathbb{K}$. Given two different representations $\mathbb{K}[x]/(h(x)) \cong \mathbb{K}[\tilde{x}]/(\tilde{h}(\tilde{x}))$ of an algebraic extension of $\mathbb{K}$, the implementation of an explicit isomorphism also boils down to modular composition.

In this paper, we study the problem of composition modulo a fixed polynomial h mostly in the case when $\mathbb{K} = \mathbb{F}_q$ is a finite field. We assume that h is separable; the case of moduli of the form $h = \hbar^t$ is studied in a separate paper [22]. Our results are based on the following simple observation: if a factorization $h = h_1 \cdots h_t$ is known, then composition modulo h reduces to t composition modulo the h_i with $i = 1, \ldots, t$. Curiously, this observation does not seem to be exploited in the standard literature on modular composition. In the case when h is irreducible over $\mathbb{K}$, but $n = \deg h$ admits a non trivial divisor m, then the second crucial observation is that h factors over $\mathbb{F}_{q^m}$. We may then apply the first observation in order to obtain an efficient algorithm for composition modulo h. Finding the factorizations of h over $\mathbb{F}_{q^m}$ can be quite expensive in general, but such computations can be regarded as pre-computations if the modulus h is fixed.

Besides modular composition, we also study the related problem of computing the *characteristic polynomial* χ of g modulo h. More precisely, we understand χ to be the characteristic polynomial of the multiplication endomorphism by g in $\mathbb{K}[x]/(h(x))$. In particular, we have $\chi \circ g = 0$ modulo h. Theoretically speaking, asymptotically fast algorithms for these tasks are due to Kedlaya and Umans [28, 29]. The advantage of our new algorithms lies in their practical efficiency.

1.1. Previous work

Denote by $\mathsf{M}_{\mathbb{K}}(n)$ the number of operations in $\mathbb{K}$ required to multiply two polynomials of degrees $<n$ in $\mathbb{K}[x]$. Let f, g and h be polynomials in $\mathbb{K}[x]$ of degrees $<n$, $<n$ and n. The naive

modular composition algorithm takes $O(n\,\mathsf{M}_{\mathbb{K}}(n))$ operations in $\mathbb{K}$. In 1978, Brent and Kung [5] gave an algorithm with cost $O\big(\sqrt{n}\,\mathsf{M}_{\mathbb{K}}(n)+n^2\big)$. It uses the *baby-step giant-step* technique due to Paterson and Stockmeyer [37], and even yields a sub-quadratic cost $O\big(n^{\varpi}+\sqrt{n}\,\mathsf{M}_{\mathbb{K}}(n)\big)$ when using fast linear algebra (see [26, p. 185]). The constant $\varpi>1.5$ is such that a $\sqrt{n}\times\sqrt{n}$ matrix over $\mathbb{K}$ may be multiplied with another $\sqrt{n}\times n$ rectangular matrix in time $O(n^{\varpi})$. The best current bound $\varpi<1.6667$ is due to Huang and Pan [24, Theorem 10.1].

When linear algebra benefits from very fast implementations, its contribution is expected to be smaller than the other polynomial operations, on a certain bounded range for n. In fact, for fixed values of ϖ and $\mathsf{M}_{\mathbb{K}}$, the sizes of the "baby" and "giant" steps may be optimized in order to balance cost contributions of matrix and polynomial operations (this was studied for finite fields in an unpublished preprint of Shoup and Smolensky in 1992). Further improvements have been proposed in [25], based on the *Lagrange inversion formula* for the reversion of formal power series.

A major breakthrough has been achieved by Kedlaya and Umans [28, 29] in the case when $\mathbb{K}$ is the finite field $\mathbb{F}_q$. For any positive $\varepsilon>0$, they have shown that the composition $f\circ g$ modulo h can be computed using $O((n\log q)^{1+\varepsilon})$ bit operations. Unfortunately, it remains a major open problem to turn this theoretical bit complexity bound into practically useful implementations.

In the special case of power series composition (i.e. when $h=x^n$), the best known complexity bound is again due to Brent and Kung: in [5], they showed that this requires $O\big(\sqrt{n}\,\mathsf{M}_{\mathbb{K}}(n)\log^{1/2}n\big)$ operations in $\mathbb{K}$, under the condition that $g'(0)$ is invertible and that the characteristic is at least n/l, where $l=\big\lceil\sqrt{n/\log n}\big\rceil$. The variant proposed by van der Hoeven [18, section 3.4.3] removes the condition on $g'(0)$. For fields of small characteristic, Bernstein [1] proposed an algorithm that is softly linear in the precision n but linear in the characteristic. These algorithms are generalized to moduli h of the form $\hbar^m$ in [22]; we show there that the composition reduces to one power series composition at order n in $\mathbb{K}[z]/(\hbar(z))$, plus m compositions modulo $\hbar$, and one characteristic polynomial computation modulo $\hbar$. Let us finally mention that series with integer, rational or floating point coefficients can often be composed in quasi-linear time as well in suitable bit complexity models, as shown by Ritzmann [40]; see also [19].

The expression $f\circ g\operatorname{rem}h$ is linear in f. It is well known that the transposition of the application $f\mapsto f\circ g\operatorname{rem}h$ corresponds to the *power projection* task, which is an important ingredient for computing minimal and characteristic polynomials. In [43], Shoup studied the computation of minimal polynomials in algebraic extensions of the form $\mathbb{F}_q[\alpha]$ or $\mathbb{F}_q[\alpha][\beta]$, explicitly given by defining polynomials. He designed fast practical algorithms with low memory consumption, built from the smart combination of "baby-step giant-step" to transposed algorithms. However his method does not improve upon the one of Brent and Kung from the asymptotic complexity point of view.

The characteristic polynomial of g modulo h may be obtained from suitable power projections of g modulo h thanks to the well-known *Newton–Girard identities*, which involve solving a first order differential equation to precision n. This is rather elementary when the characteristic is zero or sufficiently large. Otherwise, p-adic arithmetic is needed. A complete solution is described in [15]. More generally, a framework for using p-adic arithmetic to solve ordinary differential equations in positive characteristic may be found in [32].

1.2. Contributions and outline of the article

The aim of this article is to achieve practical speed-ups for modular composition and the computation of characteristic polynomials. Most of the new results are derived from the following simple observation: if h splits into linear factors in $\mathbb{K}$, and if its roots are given, then modular composition basically reduces to evaluating g at the roots of h, evaluating f at these values of g, and interpolate $f\circ g$. It is well-known that each of these steps can be done in softly linear time using multiple

point evaluation and interpolation. More generally, whenever h can be factored into $h = h_1 \cdots h_t$, the computation of $f \circ g \operatorname{rem} h$ reduces to the computations of $f \circ g \operatorname{rem} h_i$ for $i = 1, \ldots, t$.

Of course, the existence of factorizations of h heavily depends on h itself and on fields over which we allow ourselves to factor h. For instance, if $\mathbb{K} = \mathbb{Q}$, then we might consider computing the roots of h in $\mathbb{C}$ using a sufficient precision, or factoring h over the p-adic numbers $\mathbb{Q}_p$ for some well chosen prime number p. If $\mathbb{K} = \mathbb{F}_q$ is a finite field, and h is an irreducible polynomial of composite degree $n = m_1 m_2$, then we may consider factorizations over the intermediate field $\mathbb{F}_{q^{m_1}}$. In a separate paper, we study the case when $\mathbb{K}$ is the field of computable complex numbers [23]. In this paper, we mainly focus on the finite field case.

Whether the approach leads to competitive algorithms for modular composition also depends on the question whether we require the factorization of h to be part of the complexity or not. Indeed, if we are doing a large polynomial computation over $\mathbb{K}$ in the algebraic extension $\mathbb{K}[x]/(h(x))$, then we typically need to perform many modular compositions $f_i \circ g_i \operatorname{rem} h$ for different f_i and g_i, but for a fixed modulus h. In such cases, it is reasonable to regard the factorization of h as a precomputation. Furthermore, if we want to perform computations in a large finite field extension $\mathbb{L} \supseteq \mathbb{K}$ and if we are free to select a suitable representation for elements of this finite field, then we may build a modulus h with $\mathbb{L} = \mathbb{K}[x]/(h(x))$ using dedicated algorithms; these algorithms are much faster than finding an irreducible modulus at random. In fact, testing the irreducibility of h in $\mathbb{F}_q[x]$ reduces to $O(\log^2 n)$ modular compositions in degree n over $\mathbb{F}_q$, plus $\tilde{O}(n \log^2 q)$ bit operations (see [29, section 8.2], based on Rabin's algorithm [38]).

In section 2, we begin with revisiting known techniques. We introduce cost functions for modular composition, power projection, and the computation of characteristic polynomials. In section 3, we examine the benefit for modular composition when a factorization of h in $\mathbb{K}[x]$ is given. More precisely, if the irreducible factorization $\hbar_1^{m_1} \cdots \hbar_t^{m_t}$ of the modulus is available, then our method reduces the composition modulo h to several compositions modulo $\hbar_1^{m_1}, \ldots, \hbar_t^{m_t}$ in softly linear time. A key ingredient, reused several times in the article, is the simultaneous computation of characteristic polynomials and modular compositions.

In section 4, we turn our attention to the specific situation of an irreducible modulus $h \in \mathbb{K}[x]$ of composite degree $n = m_1 m_2 = \deg h$ over a finite field $\mathbb{K} = \mathbb{F}_q$. We show how to exploit the existence of factorizations of h over the intermediate fields $\mathbb{F}_{q^{m_1}}$ into factors of degree m_2.

The natural generalization to degrees $n = m_1 \cdots m_t$ with $t \geqslant 3$ will be the subject of sections 5, 6 and 7. One important question is how to represent the elements of the intermediate fields $\mathbb{K}_i$ and it is convenient to introduce the special concept of an *effective algebraic tower* for this purpose. We also introduce the notion of a *composition tower* for h, which formalizes the requirement that we are given factorizations of h over each of the intermediate fields $\mathbb{K}_i$. In section 5, we generalize the algorithm from section 4 to arbitrary composition towers. In section 6 we also give a detailed complexity analysis in the case of triangular towers when each intermediate field $\mathbb{K}_i$ admits the form $\mathbb{K}_i = \mathbb{K}[\alpha_1, \ldots, \alpha_i]$ for suitable $\alpha_i \in \mathbb{K}_i$.

Section 7 is dedicated to primitive towers, in which case each $\mathbb{K}_i$ is generated by a single primitive element α_i over $\mathbb{K}$. If the m_i are pairwise coprime (see section 7.4), then the field $\mathbb{K}_i$ can be taken to be the composed product of $\mathbb{F}_{q^{m_1}}, \ldots, \mathbb{F}_{q^{m_i}}$, and computations in the tower become particularly efficient: in the case when n is "super-smooth" (in the sense that the largest prime-power divisor d of n satisfies $d = (\log n)^{O(1)}$), we will show that composition modulo h can be done in quasi-linear time (modulo precomputations). In this very particular situation, we notice that our method outperforms Kedlaya–Umans' algorithm. If $m_1 = \cdots = m_t = p$ and p is small, then a similar result holds when using so called Artin–Schreier towers (see section 7.5). For general smooth n, one may also consider nested towers (see section 7.3) for which the primitive elements α_i are compositions of polynomials of degrees m_1, …, m_i over $\mathbb{K}$. The existence of such towers for given $\mathbb{F}_q$ and n is an interesting open problem, with a generally positive answer in practice.

Our main complexity bounds for modular composition are summarized in Table 1.1. In this table, h is a fixed irreducible polynomial of degree $n = m_1 \cdots m_t$ and $\bar{m} = \max(m_1, \ldots, m_t)$. The entries correspond to the various types of towers that are considered in sections 6 and 7, while assuming that all necessary precomputations that depend on h have been done.

This leaves us with the issue of how to conduct the precomputations. This will be the subject of section 8, where we will analyze the cost of building composition towers of various types. We will see that the construction of composition towers for prescribed composite extension degrees n can usually be done fast. On the other hand, building composition towers for a prescribed modulus h is of the same order of difficulty as factoring h over the intermediate fields $\mathbb{K}_i$, or finding a root of h in $\mathbb{F}_{q^n}$ for a given representation of the elements of $\mathbb{F}_{q^n}$. Practical algorithms for this task are then well known but of a cost that is quadratic in n.

Tower type	Expected number of operations in $\mathbb{K} = \mathbb{F}_q = \mathbb{F}_{p^d}$	
Triangular	$O(7^t \mathsf{M}_{\mathbb{K}}(\bar{m}n)\log\bar{m})$	Proposition 6.4
Primitive	$O(\mathsf{M}_{\mathbb{K}}(\bar{m}n)(2^t + \log n))$	Corollary 7.5
Nested	$O((\mathsf{M}_{\mathbb{K}}(\bar{m}n) + t\,\mathsf{M}_{\mathbb{K}}(n))\log n)$	Corollary 7.7
Composed	$O(\mathsf{M}_{\mathbb{K}}(\bar{m}n)\log n)$	Corollary 7.16
Artin–Schreier	$O(p\,d\,n\log_p^2(dn)\log_p n + \mathsf{M}_{\mathbb{F}_p}(p\,d\,n)\log n)$	Corollary 7.19

Table 1.1. Complexity bound for modular composition for various types of towers.

2. PRELIMINARIES

2.1. Complexity models

Recall that an *effective ring* is a ring $\mathbb{A}$ with unity whose elements can be represented on a computer and such that we have algorithms for performing the ring operations. Effective fields $\mathbb{K}$ and effective algebras over an effective ring are defined similarly.

Given an effective ring $\mathbb{A}$, *algebraic complexity models* express running times in terms of the number of operations in $\mathbb{A}$. Unless otherwise stated, we will analyze the costs of the algorithms in this paper in this way. More precisely, our results both apply for the straight-line program and computation tree models [6, chapter 4].

For randomized algorithms over a finite effective ring $\mathbb{A}$, we assume a special instruction that uniformly generates random elements in $\mathbb{A}$. For simplicity we assume that this instruction has constant cost. The *expected cost* of a randomized algorithm and a given input is the average cost taken over all the possible executions.

When working over the finite field $\mathbb{K} = \mathbb{F}_q$ with q elements, we may also analyze the costs of algorithms in the *bit complexity model*, which relies on Turing machines with a sufficient number of tapes [36]. We will not explicitly consider this model in our paper, but most of our complexity bounds can easily be converted to this setting.

2.2. Polynomial multiplication

Let $\mathbb{A}$ be an effective ring with unity, let $n \in \mathbb{N}$, and denote

$$\mathbb{A}[x]_{<n} = \{f \in \mathbb{A}[x] : \deg f < n\}.$$

Given a polynomial or power series $f(x) = \sum_{i \geqslant 0} f_i x^i$ and $l \leqslant h$, it is convenient to write

$$u(x)_{l;h} = \sum_{0 \leqslant i < h-l} u_{i+l} x^i$$
$$u(x)_{;h} = \sum_{0 \leqslant i < h} u_i x^i.$$

We write $\mathsf{M}_{\mathbb{A}}: \mathbb{N} \to \mathbb{R}^{\geq}$ for a cost function such that two polynomials in $\mathbb{A}[x]_{<n}$ can be multiplied using $\mathsf{M}_{\mathbb{A}}(n)$ operations in $\mathbb{A}$. The schoolbook algorithm allows us to take $\mathsf{M}_{\mathbb{A}}(n) = O(n^2)$. The fastest currently known algorithm [7] yields $\mathsf{M}_{\mathbb{A}}(n) = O(n \log n \log\log n) = \tilde{O}(n)$. Here, the *soft-Oh* notation $f(n) \in \tilde{O}(g(n))$ means that $f(n) = g(n) \log^{O(1)} g(n)$ (we refer the reader to [12, chapter 25, section 7] for technical details). If $\mathbb{A}$ is a field of finite characteristic, then it has been shown [17] that $\mathsf{M}_{\mathbb{A}}(n) = O(n \log n\, 8^{\log^* n})$, where $\log^*$ denotes the *iterated logarithm* function. In what follows, we will always assume that $\mathsf{M}_{\mathbb{A}}(n)/n$ is an increasing function in n. This customary assumption implies the *super-additivity* of $\mathsf{M}_{\mathbb{A}}$, namely $\mathsf{M}_{\mathbb{A}}(n_1) + \mathsf{M}_{\mathbb{A}}(n_2) \leqslant \mathsf{M}_{\mathbb{A}}(n_1 + n_2)$ for all $n_1 \geqslant 0$ and $n_2 \geqslant 0$.

More generally, if $\mathbb{B}$ is an effective $\mathbb{A}$-algebra, then it is sometimes convient to denote by $\mathsf{M}_{\mathbb{B}/\mathbb{A}}: \mathbb{N} \to \mathbb{R}^{\geq}$ a cost function such that two polynomials in $\mathbb{B}[x]_{<n}$ can be multiplied using $\mathsf{M}_{\mathbb{B}/\mathbb{A}}(n)$ operations in $\mathbb{A}$.

2.3. Univariate arithmetic

Let $\mathbb{K}$ be an effective field. The *remainder* (resp. *quotient*) of the euclidean division of g by h in $\mathbb{K}[x]$ is denoted by $g \operatorname{rem} h$ (resp. by $g \operatorname{quo} h$). For a fixed modulus of degree n, euclidean divisions by h are usually performed by computing a *pre-inverse* φ of h. More precisely, φ is the inverse of $x^{-n} h$ in $\mathbb{K}[[x^{-1}]]$, computed at precision $O(x^{-n})$. Given $g \in \mathbb{K}[x]_{<2n}$, one obtains the quotient $f \operatorname{quo} h$ by multiplying $f_{n;2n}$ with φ and the remainder as $f \operatorname{rem} h = f - (f \operatorname{quo} h)\, h$. Given $f, g \in \mathbb{K}[x]_{<n}$ we may thus compute the modular product $fg \operatorname{rem} h$ using $3\,\mathsf{M}_{\mathbb{K}}(n) + O(n)$ operations in $\mathbb{K}$.

We recall that the greatest common divisor of two polynomials of degrees at most n over $\mathbb{K}$ can be computed using $O(\mathsf{M}_{\mathbb{K}}(n) \log n)$ operations in $\mathbb{K}$ [12, Algorithm 11.4].

Let $f \in \mathbb{K}[x]_{<n}$ and consider n points $\sigma_1, \ldots, \sigma_n \in \mathbb{K}$. Then the evaluations $f(\sigma_1), \ldots, f(\sigma_n)$ can be computed using $O(\mathsf{M}_{\mathbb{K}}(n) \log n)$ operations in $\mathbb{K}$ [12, chapter 10]. This operation is also called *multipoint evaluation*. The inverse operation is the *interpolation*, which consists of recovering f from $f(\sigma_1), \ldots, f(\sigma_n)$; it can be performed with a similar cost. If the σ_i are fixed, then it is often possible to gain an additional factor of $\log\log n$ using FFT trading [20].

More generally, if $g_1, \ldots, g_l \in \mathbb{K}[x]$ are polynomials with $\deg g_1 + \cdots + \deg g_l = O(n)$, then all the remainders $f \operatorname{rem} g_i$ can be computed using $O(\mathsf{M}_{\mathbb{K}}(n) \log l)$ operations in $\mathbb{K}$. The inverse problem, called *Chinese remaindering*, again admits the same complexity $O(\mathsf{M}_{\mathbb{K}}(n) \log l)$.

2.4. Bivariate arithmetic

Let $\mathbb{A}$ be an effective ring. Given a bivariate polynomial $f \in \mathbb{A}[z, x]$, we define its *bi-degree* to be the pair (m, n) with $m = \deg_z f$ and $n = \deg_x f$. Using Kronecker substitution [12, chapter 8, section 4], two polynomials of bi-degree (m, n) may be multiplied using $O(\mathsf{M}_{\mathbb{A}}(n\,m))$ ring operations in $\mathbb{A}$.

If h is a monic polynomial of degree m in $\mathbb{A}[z]$, and if f and g are two polynomials in $(\mathbb{A}[z]/h(z))[x]_{<n}$ then their preimages may be multiplied in $\mathbb{A}[z, x]$ with $O(\mathsf{M}_{\mathbb{A}}(m\,n))$ operations in $\mathbb{A}$ before being projected into $(\mathbb{A}[z]/h(z))[x]$ with $O(n\,\mathsf{M}_{\mathbb{A}}(m))$ additional operations. Consequently each ring operation in $(\mathbb{A}[z]/h(z))[x]$ in degree $\leqslant n$ reduces to $O(\mathsf{M}_{\mathbb{A}}(m\,n))$ operations in $\mathbb{A}$. If g is monic in x then $f \operatorname{rem} g$ also takes $O(\mathsf{M}_{\mathbb{A}}(m\,n))$ operations in $\mathbb{A}$.

The computation of bivariate subresultants usually relies on fast evaluation/interpolation, as in the following well known proposition.

PROPOSITION 2.1. [12, Corollary 11.18] *Let $\mathbb{K}$ be an effective field with $>2\,m\,n$ elements. Any polynomial subresultant in x of two polynomials A and B in $\mathbb{K}[z, x]$ of bidegrees (m, n) can be computed using $O(n\,\mathsf{M}_{\mathbb{K}}(m\,n) \log(m\,n))$ operations in $\mathbb{K}$.*

Proof. The subresultant polynomial R_i of degree i of A and B in $\mathbb{K}[z, x]$ has degree $\leqslant 2\,m\,(n - i)$ in z. It can be computed by evaluating A and B at $O(m\,n)$ values for z in $\mathbb{K}$, computing $O(m\,n)$ subresultants in $\mathbb{K}[x]$ of degree $\leqslant n$, and interpolating the coefficients of R_i. In total this costs $O(n^2\,\mathsf{M}_{\mathbb{K}}(m) \log m + m\,n\,\mathsf{M}_{\mathbb{K}}(n) \log n + i\,\mathsf{M}_{\mathbb{K}}(m\,n) \log(m\,n))$. □

However for the sake of generality we will rely on the following result.

PROPOSITION 2.2. *Let $\mathbb{K}$ be an effective field. Any polynomial subresultant in x of two polynomials A and B in $\mathbb{K}[z,x]$ of bidegrees (m,n), with the corresponding Bézout relation, can be computed using $O(\mathsf{M}_{\mathbb{K}}(mn^2)\log n)$ operations in $\mathbb{K}$ that comprise at most $\min(\deg_x A, \deg_{xB})+1$ inversions in $\mathbb{K}$.*

Proof. This result corresponds to [34, Corollary 26]. The number of inversions comes from the fact that the underlying algorithm only needs to perform exact divisions by subresultant coefficients in $\mathbb{K}[z]$. Each division requires to invert the leading coefficient of the divisor. There exist at most $\min(\deg_x A, \deg_x B)+1$ such leading coefficients. □

2.5. Finite field arithmetic

Let $\mathbb{F}_q$ be the finite field with q elements. One way to represent elements of a finite field extension $\mathbb{F}_{q^m}$ is as remainder classes of polynomials in $\mathbb{F}_q[z]_{<m}$ modulo a monic reducible polynomial $\mu \in \mathbb{F}_q[z]$ of degree n. We write $\mathbb{F}_{q^n} = \mathbb{F}_q[z]/(\mu(z))$ in order to emphasize that we use this representation. Multiplication in $\mathbb{F}_{q^m}$ can be done using $3\,\mathsf{M}_{\mathbb{F}_q}(n)+O(n)$ operations in $\mathbb{F}_q$. Given an element $\alpha \in \mathbb{F}_{q^m}$ of degree $d \mid m$ over $\mathbb{F}_q$, we write $\mathbb{F}_q[\alpha]$ for the subfield of $\mathbb{F}_{q^m}$ generated by α over $\mathbb{F}_q$, where we understand that elements in $\mathbb{F}_q[\alpha]$ are represented as evaluations of polynomials in $\mathbb{K}[z]_{<d}$ at $z=\alpha$.

For the bulk of the algorithms in this paper, we will work over the finite field $\mathbb{K}=\mathbb{F}_q$. In that case, it can be shown that two polynomials in $\mathbb{F}_q[x]_{<n}$ can be multiplied in time $O(n \log q \log(n \log q)\, 8^{\log^*(n\log q)})$ on a Turing machine with a sufficient number of tapes [17]. The algebraic complexity bounds in this paper are easy to adapt to this model: it mainly suffices to replace $\mathsf{M}_{\mathbb{F}_q}(n)$ by $O(n \log q \log(n \log q)\, 8^{\log^*(n\log q)})$ in all bounds.

2.6. Matrix multiplication

The constant $\omega > 2$ represents a *feasible exponent* for the multiplication cost of matrices: two square matrices of size $n \times n$ can be multiplied using $O(n^\omega)$ operations in their coefficient ring. The constant $\varpi > 1.5$ is defined similarly but for multiplying a $\sqrt{n} \times \sqrt{n}$ matrix by a $\sqrt{n} \times n$ rectangular one. At present time the best known bound $\omega < 2.3729$ is due to Le Gall [33]. This naturally yields $\varpi \leqslant (\omega+1)/2 < 1.6845$. However the latter bound does not improve upon the earlier bound $\varpi < 1.6667$ due to Huang and Pan [24, Theorem 10.1].

2.7. The cost of modular composition and related operations

Let $\mathbb{A}$ be an effective ring. Let $\mathbb{B}$ an effective $\mathbb{A}$-algebra of dimension d whose elements are represented by vectors of size d in a given basis. We introduce the following cost functions:

- $\mathsf{C}_{\mathbb{A}}(n)$: the cost of computing the *modular composition* $f \circ g \operatorname{rem} h$, where $h \in \mathbb{A}[x]$ is a monic polynomial of degree n, and $f, g \in \mathbb{A}[x]_{<n}$.
- $\mathsf{C}_{\mathbb{B}/\mathbb{A}}(n)$: the cost of computing the *modular composition* $f \circ g \operatorname{rem} h$ in terms of operations in $\mathbb{A}$, where $h \in \mathbb{B}[x]$ is a monic polynomial of degree n, and $f, g \in \mathbb{A}[x]_{<n}$.
- $\mathsf{Q}_{\mathbb{A}}(n)$: the cost of computing the *characteristic polynomial* χ of $g \in \mathbb{A}[x]_{<n}$ modulo a monic polynomial $h \in \mathbb{A}[x]$ of degree n. This is the characteristic polynomial $\chi \in \mathbb{A}[x]$ of the multiplication endomorphism by $g \bmod h$ in $\mathbb{A}[x]/(h(x))$.
- $\mathsf{P}_{\mathbb{A}}(n)$: the cost of *modular power projections*, i.e. the cost to compute $\varphi(1)$, $\varphi(g)$, …, $\varphi(g^{n-1} \operatorname{rem} h)$, where $h \in \mathbb{A}[x]$ is monic of degree n and φ is a linear form on $\mathbb{A}[x]_{<n}$.

Let us recall a few known results about these functions.

THEOREM 2.3. *Let h be a monic polynomial of degree n over a ring $\mathbb{A}$, and let $f, g \in \mathbb{A}[x]_{<n}$. The composed polynomial $f \circ g \operatorname{rem} h$ may be computed with*

1. *$O(n\,\mathsf{M}_{\mathbb{A}}(n))$ operations in $\mathbb{A}$, or*

2. $O(n^{\varpi}+n^{1/2}\mathsf{M}_{\mathbb{A}}(n))$ *or* $O(n^{\varpi})$ *operations in* $\mathbb{A}$.

Proof. The first bound is immediate. The proof of the second bound is detailed in [12, section 12.2]. □

For a fixed monic polynomial h in $\mathbb{A}[x]$ of degree n, the modular composition $f \circ g \operatorname{rem} h$ is a linear operation in f. For f and g of degrees $<n$, the corresponding transposed application is precisely the operation of modular power projections. If a modular composition algorithm with cost $\mathsf{C}_{\mathbb{A}}(n)$ can be transposed in the sense of [3], then this leads to a power projection algorithm with cost $\mathsf{P}_{\mathbb{A}}(n)=\mathsf{C}_{\mathbb{A}}(n)+O(n)$.

THEOREM 2.4. *Let h be a monic polynomial of degree n over a ring $\mathbb{A}$, and let $g\in\mathbb{A}[x]_{<n}$. The characteristic polynomial χ of g modulo h can be computed using*

1. $O(\mathsf{M}_{\mathbb{A}}(n^2)\log n+n\,\mathsf{M}_{\mathbb{A}}(n)\log^2 n)$ *operations in $\mathbb{A}$, including divisions in $\mathbb{A}$ (the partial division in $\mathbb{A}$ is supposed to be implemented), or*
2. $O(\mathsf{M}_{\mathbb{A}}(n^2)\log n)$ *operations in $\mathbb{A}$, if $\mathbb{A}$ is a field, or*
3. $O(n\,\mathsf{M}_{\mathbb{A}}(n)\log n)$ *operations in $\mathbb{A}$, if $\mathbb{A}$ is a field with $>n$ elements, or*
4. $\mathsf{P}_{\mathbb{A}}(n)+\mathsf{M}_{\mathbb{A}}(n)$ *operations in $\mathbb{A}$, if there exist given inverses of $2,3,\ldots,n$ in $\mathbb{A}$.*

Proof. See [22, section 2.2]. □

3. MODULAR COMPOSITION VIA FACTORIZATION

3.1. Separable moduli over algebraically closed fields

Let $\mathbb{K}$ be an effective algebraically closed field. A monic polynomial $h=x^n+h_{n-1}x^{n-1}+\cdots+h_0\in\mathbb{K}[x]$ is said to be *separable* if $\gcd(h,h')=1$. Since $\mathbb{K}$ is algebraically closed, this implies that h admits n pairwise distinct roots σ_1, …, σ_n in $\mathbb{K}$, and we may use the following algorithm for composition modulo h:

Algorithm 3.1

Input. Polynomials $f,g\in\mathbb{K}[x]_{<n}$ and pairwise distinct $\sigma_1,\ldots,\sigma_n\in\mathbb{K}$.

Output. $f\circ g\operatorname{rem} h$, where $h=(x-\sigma_1)\cdots(x-\sigma_n)$.

1. Compute $v_1=g(\sigma_1),\ldots,v_n=g(\sigma_n)$ using fast multi-point evaluation.
2. Compute $w_1=f(v_1),\ldots,w_n=f(v_n)$ using fast multi-point evaluation.
3. Retrieve $\varrho\in\mathbb{K}[x]_{<n}$ with $\varrho(\sigma_1)=v_1,\ldots,\varrho(\sigma_n)=v_n$ using fast interpolation.
4. Return ϱ.

THEOREM 3.1. *Algorithm 3.1 is correct and requires $O(\mathsf{M}(n)\log n)$ operations in $\mathbb{K}$.*

Proof. By construction, $\varrho(\sigma_i)=(f\circ g)(\sigma_i)=(f\circ g\operatorname{rem}h)(\sigma_i)$ for $i=1,\ldots,n$. Since $\deg\varrho<n$ and the σ_i are pairwise distinct, it follows that $\varrho=f\circ g\operatorname{rem}h$. This proves the correctness of the algorithm. The complexity bound follows from the fact that each of the steps 1, 2 and 3 can be performed in time $O(\mathsf{M}_{\mathbb{K}}(n)\log n)$. □

3.2. Pairwise coprime moduli

Let $\mathbb{K}$ again be a general effective field. The algorithm from the previous section may be generalized to composition modulo a polynomial h that can be factored partially as $h=h_1\cdots h_t$ in $\mathbb{K}[x]$, where the polynomials h_i are pairwise coprime (although not necessarily irreducible).

Algorithm 3.2

Input. Pairwise coprime polynomials $h_1, \ldots, h_t$ in $\mathbb{K}[x]$ such that $h = h_1 \cdots h_t$ has degree n; Polynomials f, g in $\mathbb{K}[x]_{<n}$.

Output. $f \circ g \operatorname{rem} h$, and the characteristic polynomial of g modulo h.

1. Use a multi-remainder algorithm to compute $g_i = g \operatorname{rem} h_i$, for all $1 \leqslant i \leqslant t$.
2. For all $1 \leqslant i \leqslant t$, compute the characteristic polynomial χ_i of g_i modulo h_i.
3. Use a multi-remainder algorithm to compute $f_i = f \operatorname{rem} \chi_i$, for all $1 \leqslant i \leqslant t$.
4. For all $1 \leqslant i \leqslant t$, perform the modular composition $\varrho_i = f_i \circ g_i \operatorname{rem} h_i$.
5. Use Chinese remaindering to compute ϱ in $\mathbb{K}[x]$ of degree $\leqslant n-1$ such that $\varrho = \varrho_i \bmod h_i$ for all $1 \leqslant i \leqslant t$.
6. Return ϱ and $\chi_1 \cdots \chi_t$.

PROPOSITION 3.2. *Algorithm 3.2 is correct and takes* $O(\mathsf{M}_{\mathbb{K}}(n) \log t) + \sum_{i=1}^{t} (\mathsf{Q}_{\mathbb{K}}(n_i) + \mathsf{C}_{\mathbb{K}}(n_i))$ *operations in* $\mathbb{K}$, *where* $n_i = \deg h_i$.

Proof. For all $1 \leqslant i \leqslant t$, the Cayley–Hamilton theorem gives us $\chi_i \circ g = 0 \bmod h_i$, which implies $f \circ g \bmod h_i = (f \operatorname{rem} \chi_i) \circ (g \operatorname{rem} h_i) \bmod h_i$, whence the correctness of $\varrho = f \circ g \operatorname{rem} h$. The correctness of the characteristic polynomial of g follows from the usual isomorphism of $\mathbb{K}$-algebras $\mathbb{K}[x]/(h(x)) \cong \mathbb{K}[x]/(h_1(x)) \times \cdots \times \mathbb{K}[x]/(h_t(x))$.

The costs of steps 1, 3, 5, and 6 are $O(\mathsf{M}_{\mathbb{K}}(n) \log t)$. Step 2 costs $\sum_{i=1}^{t} \mathsf{Q}_{\mathbb{K}}(n_i)$, and step 4 takes $\sum_{i=1}^{t} \mathsf{C}_{\mathbb{K}}(n_i)$ operations in $\mathbb{K}$. □

Example 3.3. For some families of polynomials the irreducible factorization is explicitly known. For instance, the following result is due to Serret [41, section III, chapitre III, pp. 158–162] (see also [10, pp. 24–27], [35, Theorem 3.2.18]):

> Let $\mathbb{F}_q$ be a finite field of characteristic p such that $q + 1 = 2^A u$ with $A \geqslant 2$ and u odd. Let $a \in \mathbb{F}_q$ be an element of order e. Let t be a multiple of 2^A having all its prime factors dividing e but not $(q-1)/e$. Then the polynomial $x^t - a$ factors into 2^{A-1} irreducible polynomials of degrees $t/2^{A-1}$. The irreducible factors may be described explicitly.

For example, with $q = p = 7$, $A = 3$, $u = 1$, $e = 2$, $a = 6$, $t = 16$, the polynomial $x^{16} + 1$ factors into irreducible polynomials of degree 2. Taking $e = 6$ instead leads to $x^{16} + 2$ and $x^{16} + 4$.

4. EXPLOITING FACTORIZATIONS OVER ALGEBRAIC EXTENSIONS

4.1. Degree reduction

Let $\mathbb{K}$ still be an effective field and assume that we wish to compute a modular composition $f \circ g \operatorname{rem} h$, where $f, g, h \in \mathbb{K}[x]$ and h is monic. Let us study what happens if the polynomials f and g to be composed have degrees larger than n. We clearly have

$$f \circ g \operatorname{rem} h \;=\; f \circ (g \operatorname{rem} h) \operatorname{rem} h$$

and we may compute $g \operatorname{rem} h$ using $O\big(\lfloor \frac{\deg g}{n} \rfloor \, \mathsf{M}_{\mathbb{K}}(n)\big)$ operations in $\mathbb{K}$ [12, Exercise 9.16]. Without loss of generality we may therefore assume that $\deg g < n$.

If $\deg f$ exceeds n, then it suffices to perform $\lfloor \deg f / n \rfloor$ modular compositions:

$$f \circ g \operatorname{rem} h = \left(\sum_{i=0}^{\lfloor \deg f / n \rfloor} (f_{in;(i+1)n} \circ g \operatorname{rem} h)\, (g^{in} \operatorname{rem} h) \right) \operatorname{rem} h.$$

This requires $\left\lfloor \frac{\deg f}{n} \right\rfloor$ additional compositions modulo h in size n, plus $O\big(\left\lfloor \frac{\deg f}{n} \right\rfloor \mathsf{M}_{\mathbb{K}}(n)\big)$ operations in $\mathbb{K}$.

Alternatively, given a polynomial θ with $\theta \circ g \operatorname{rem} h = 0$, the following formula provides us with a more efficient way to reduce the degree of f:

$$f \circ g \operatorname{rem} h = (f \operatorname{rem} \theta) \circ g \operatorname{rem} h.$$

Taking θ to be the characteristic polynomial χ of g, its computation usually admits a similar cost as modular composition. Therefore it is worth using this method unless $\deg f \leqslant n + o(n)$. This is actually one key ingredient for the upcoming algorithms for modular composition: in order to reduce a composition modulo h to compositions modulo a factor $\hbar$ of h, we in particular need to compute the characteristic polynomial of g modulo $\hbar$. At the end of the recursive calls, one should nevertheless keep in mind that we only need annihilating polynomials, so that we may also use minimal polynomials. Shoup has given a probabilistic $O\big(\sqrt{n}\,\mathsf{M}_{\mathbb{K}}(n) + n^2\big)$ algorithm for computing minimal polynomials [43], which is useful for actual implementations.

4.2. Normal factorizations

In the case when we wish to compute a composition modulo an irreducible polynomial $h \in \mathbb{K}[x]$, we cannot apply the algorithms from sections 3.1 and 3.2. Nevertheless, it might happen that h admits a non trivial factorization over an algebraic extension of $\mathbb{K}$. This generically happens when $\mathbb{K}$ is a finite field and $\deg h$ is composite. Indeed, we recall the following well known result.

PROPOSITION 4.1. *Let h be a monic irreducible polynomial in $\mathbb{F}_q[x]$ of degree n, and let m be an integer dividing n. Then there exist an irreducible polynomial $\mu(z) \in \mathbb{F}_q[z]$ of degree m, and a polynomial $H(z, x) \in \mathbb{F}_q[z, x]$ of bi-degree $(<m, n/m)$, monic in x, such that the irreducible factorization of $h(x)$ over $\mathbb{F}_q[z]/(\mu(z))$ is exactly $\prod_{\mu(\zeta)=0} H(\zeta, x)$.*

Proof. Since h is irreducible, $\mathbb{F}_q[y]/(h(y))$ is isomorphic to $\mathbb{F}_{q^n}$, which contains $\mathbb{F}_{q^m}$. We may thus take a generator $\alpha \in \mathbb{F}_q[y]/(h(y))$ of the image of $\mathbb{F}_{q^m}$ in $\mathbb{F}_q[y]/(h(y))$, and write $\mu(z) \in \mathbb{F}_q[z]$ its minimal polynomial over $\mathbb{F}_q$. We set $H(\alpha, x)$ to the monic minimal polynomial of the class β of y in $\mathbb{F}_q[y]/(h(y))$ over $\mathbb{F}_q[\alpha]$. It divides h, and all its conjugates $H\left(\alpha^{q^i}, x\right)$ do so for $i \in \{0, \ldots, m-1\}$. On the other hand since $H(\alpha, \beta) = 0$, then $H\left(\alpha^{q^i}, \beta^{q^i}\right) = 0$, any root of h is a root of one of the $H\left(\alpha^{q^i}, x\right)$, which proves the equality $h(x) = \prod_{\mu(\zeta)=0} H(\zeta, x)$. □

We call factorizations as in this proposition "normal factorizations". This concept can actually be defined over arbitrary fields, as follows. Let h be a monic separable polynomial in $\mathbb{K}[x]$, let m be a divisor of $n = \deg h$, and let $\mu \in \mathbb{K}[z]$ be a monic separable irreducible polynomial of degree m. We set $\mathbb{L} = \mathbb{K}[z]/(\mu(z))$, and write α for the class of z in $\mathbb{L}$. For all roots ζ of μ in $\bar{\mathbb{K}}$, we write σ_ζ for the map from $\mathbb{L}$ to $\bar{\mathbb{K}}$ that sends α to ζ. We say that h admits a *normal factorization* over $\mathbb{L}$ if there exists a bivariate polynomial $H(z, x)$ that is monic in x, of bi-degree $(<m, n/m)$, and such that h factors into $\prod_{\mu(\zeta)=0} H(\zeta, x)$ over $\bar{\mathbb{K}}$, with $H(\zeta_1, x)$ and $H(\zeta_2, x)$ coprime whenever $\zeta_1 \neq \zeta_2$. We call $H(\alpha, x)$ the *normal factor* of h over $\mathbb{L}$. Notice that the polynomials h and $H(\zeta, x)$ are not required to be irreducible here.

Example 4.2. With $\mathbb{K} = \mathbb{F}_2$, $h(x) = x^6 + x + 1 \in \mathbb{F}_2[x]$ is irreducible, and we have $\mathbb{F}_{2^6} \simeq \mathbb{F}_2[y]/(h(y))$. For $m = 2$ we take $\mu(z) = z^2 + z + 1$ and present $\mathbb{L} = \mathbb{F}_{2^2}$ as $\mathbb{F}_2[z]/(\mu(z))$. Then h factors over $\mathbb{L}$ as

$$h(x) = (x^3 + x^2 + \zeta_1 x + \zeta_1 + 1)\,(x^3 + x^2 + \zeta_2 x + \zeta_2 + 1),$$

where ζ_1, ζ_2 are the two roots of μ in $\mathbb{F}_{2^2}$. More precisely for ζ_1 we may take the class of z in $\mathbb{L}$, whereas $\zeta_2 = \zeta_1 + 1$. The normal factor of h is thus $H(z, x) = x^3 + x^2 + zx + z + 1$.

Example 4.3. When $\mathbb{K}=\mathbb{Q}$ the situation is different from the case of finite fields. For instance $h(x)=x^6+x+1$ is irreducible, but it remains irreducible over $\mathbb{Q}[\mathrm{i}]$, $\mathbb{Q}[\sqrt{2}]$, etc. Nevertheless, for a prescribed extension degree n, we may randomly pick an irreducible $\mu(z)$ of degree m and an irreducible $H(z,x)$ in $\mathbb{Q}[z]/(\mu(z))[x]$ of bi-degree $(<m, n/m)$, and build $h(x)$ as $\mathrm{Res}_z(H(z,x),\mu(z))$. If h is separable, then it is irreducible in $\mathbb{Q}[x]$, and H is a normal factor of h over $\mathbb{L}=\mathbb{Q}[z]/(\mu(z))$.

4.3. Single extensions

Assume that h admits a normal factorization as above. Then the Chinese remainder theorem yields a natural isomorphism

$$\mathbb{K}[x]/(h(x)) \;\cong\; \mathbb{K}[z,x]/(\mu(z),H(z,x))$$

and we may define $a(x)$ as the unique polynomial in $\mathbb{K}[x]_{<n}$ that satisfies $H(a(x),x)=0 \bmod h(x)$ and $\mu(a(x))=0 \bmod h(x)$. We may now adapt the algorithm from section 3.2 as follows:

Algorithm 4.1

Input. Polynomials h, μ, H, a as above, and f, g in $\mathbb{K}[x]_{<n}$.

Output. $f\circ g \operatorname{rem} h$, and the characteristic polynomial of g modulo h.

1. Compute the remainder $G(\alpha,x)=g(x) \operatorname{rem} H(\alpha,x)$ in $\mathbb{L}[x]$.
2. Compute the characteristic polynomial $\chi(\alpha,x)$ of $G(\alpha,x)$ modulo $H(\alpha,x)$ in $\mathbb{L}[x]$.
3. Compute $F(\alpha,x)=f(x) \operatorname{rem} \chi(\alpha,x)$ in $\mathbb{L}[x]$.
4. Perform the modular composition $\varrho(\alpha,x)=F(\alpha,G(\alpha,x)) \operatorname{rem} H(\alpha,x)$ in $\mathbb{L}[x]$.
5. Return $\varrho(a(x),x) \operatorname{rem} h(x)$ and $\mathrm{Res}_z(\chi(z,x),\mu(z))$.

PROPOSITION 4.4. *Algorithm 4.1 is correct, and takes*

$$\mathsf{Q}_{\mathbb{L}/\mathbb{K}}(n/m)+\mathsf{C}_{\mathbb{L}/\mathbb{K}}(n/m)+O(\mathsf{M}_{\mathbb{K}}(mn)\log m)$$

operations in $\mathbb{K}$.

Proof. We first observe that

$$(f\circ g)(x) \operatorname{rem} H(\alpha,x)=(f(x) \operatorname{rem} \chi(\alpha,x))\circ(g(x) \operatorname{rem} H(\alpha,x)) \operatorname{rem} H(\alpha,x)=\varrho(\alpha,x).$$

It follows that $(f\circ g)(x) \operatorname{rem} h(x)=\varrho(a(x),x) \operatorname{rem} h(x)$, whence $f\circ g \operatorname{rem} h$ is computed correctly. As to the characteristic polynomial of g, the argument is the same as for Algorithm 3.2, thanks to the Poisson formula $\mathrm{Res}_z(\chi(z,x),\mu(z))=(-1)^n\prod_{\mu(\zeta)=0}\chi(\zeta,x)$.

Now the multiplication of two polynomials in $\mathbb{L}[x]$ of degree n using Kronecker substitution requires $O(\mathsf{M}_{\mathbb{K}}(mn))$ operations in $\mathbb{K}$. This way, steps 1 and 3 take $O(\mathsf{M}_{\mathbb{K}}(mn))$ operations in $\mathbb{K}$. Steps 2 and 4 respectively cost $\mathsf{Q}_{\mathbb{L}/\mathbb{K}}(n/m)$ and $\mathsf{C}_{\mathbb{L}/\mathbb{K}}(n/m)$ operations in $\mathbb{K}$. The computation of $\varrho(a(x), x) \bmod h(x)$ in the last step may be done naively using $O(m\,\mathsf{M}_{\mathbb{K}}(n))$ operations in $\mathbb{K}$. The computation of $\mathrm{Res}_z(\chi(z,x),\mu(z))$ requires $O(\mathsf{M}_{\mathbb{K}}(mn)\log m)$ further operations by Proposition 2.2. □

COROLLARY 4.5. *With the above notations, and given a normal factorization of h for $m=O(\sqrt{n})$, $n/m=O(\sqrt{n})$, the modular composition $f\circ g \operatorname{rem} h$ can be computed using $O(\mathsf{M}_{\mathbb{K}}(n^{3/2})\log n)$ operations in* $\mathbb{K}$.

Proof. We simply apply Algorithm 4.1. For $\mathsf{Q}_{\mathbb{L}/\mathbb{K}}(n/m)$ we use $\chi(\alpha,y)=\mathrm{Res}_x(G(\alpha,x)-y,H(\alpha,x))$, which takes $O(\mathsf{M}_{\mathbb{K}}(n^2/m)\log n+(n/m)\,\mathsf{M}(m)\log m)$ operations in $\mathbb{K}$ by Proposition 2.2. We perform the computations in step 4 naively, which yields $\mathsf{C}_{\mathbb{L}/\mathbb{K}}(n/m)=O((n/m)\,\mathsf{M}_{\mathbb{K}}(n))$. □

5. COMPOSITION TOWERS

5.1. Effective towers

Corollary 4.5 already illustrates the potential of our ability to factor h non trivially over an extension field $\mathbb{L} \supseteq \mathbb{K}$. This idea can be pushed even farther whenever the factors $H(\zeta, x)$ with $\mu(\zeta) = 0$ can be factored recursively over a tower of extension fields of $\mathbb{L}$. In order to carry out this generalization, we first need to decide how to compute with elements in the successive fields of such a tower. Instead of privileging particular representations, we rely on the abstract concept of an *effective tower*.

DEFINITION 5.1. *An* effective tower *over* $\mathbb{K}$ *is a tower of fields*

$$\mathbb{K} = \mathbb{K}_0 \subsetneq \mathbb{K}_1 \subsetneq \cdots \subsetneq \mathbb{K}_t$$

with the following properties:

- *Each field* $\mathbb{K}_i$ *comes with a specific way to represent its elements and algorithms for the field operations.*
- *For* $i = 1, \ldots, t$, *the field* $\mathbb{K}_i$ *is a finite separable algebraic extension of* $\mathbb{K}_{i-1}$, *and we have precomputed an element* $\alpha_i \in \mathbb{K}_i$ *along with its minimal polynomial* μ_i *over* $\mathbb{K}_{i-1}$, *such that* $m_i = \deg \mu_i \geqslant 2$ *and* $\mathbb{K}_i \cong \mathbb{K}_{i-1}[\alpha_i]$. *We set* $\bar{m} = \max(m_1, \ldots, m_t)$.
- *For* $i = 1, \ldots, t$, *we have algorithms for computing the natural bijection* Λ_i, *given by*

$$\begin{aligned} \mathbb{K}_{i-1}[z]_{<m_i} &\xrightarrow{\Lambda_i} \mathbb{K}_i \\ z &\longmapsto \alpha_i \end{aligned}$$

  and its inverse Λ_i^{-1}. *We call* Λ_i *and* Λ_i^{-1} *the* upward *and* downward *conversions at level i. The coefficientwise extensions of* Λ_i *and* Λ_i^{-1} *yield mappings* $\mathbb{K}_{i-1}[z]_{<m_i}[x] \to \mathbb{K}_i[x]$ *and* $\mathbb{K}_i[x] \to \mathbb{K}_{i-1}[z]_{<m_i}[x]$ *that we still denote by* Λ_i *and* Λ_i^{-1}.

We denote by $\mathsf{M}_{\mathbb{K}_i/\mathbb{K}}(n)$ the cost of multiplying two polynomials in $\mathbb{K}_i[x]_{<n}$ in terms of the number of required operations in $\mathbb{K}$. Similarly, we write $\mathsf{D}_{\mathbb{K}_i/\mathbb{K}}$ for the cost of inverting an element in $\mathbb{K}_i$ in terms of the number of operations in $\mathbb{K}$. We always assume that additions and subtractions can be done in linear time. We also let $\mathsf{L}_{\mathbb{K}_i/\mathbb{K}}$ upper bound the costs of both the upward and downward conversions at level i.

5.2. Composition towers

Let us now return to our particular modulus $h \in \mathbb{K}[x]$ and assume that its degree $n = \deg h$ admits the factorization $n = m_1 \cdots m_t$ with $m_i \geqslant 2$ for $i = 1, \ldots, t$. If $t = 2$, then Algorithm 4.1 shows how to reduce modular composition modulo h to composition modulo a polynomial over $\mathbb{K}_1$ of degree n/m_1, provided a normal factor of h over $\mathbb{K}_1$ exists and is given. In order to generalize this idea to the case when $t > 2$, it is useful to introduce the concept of a *composition tower*.

DEFINITION 5.2. *Let* $(\mathbb{K}_i)_{i \leqslant t}$ *be an effective tower. Let* $h \in \mathbb{K}[x]$ *be a monic separable polynomial of degree* $n = m_1 \cdots m_t$. *We say that* $(\mathbb{K}_i)_{i \leqslant t}$ *is a* composition tower *for h over* $\mathbb{K}$ *if the following properties are satisfied:*

- *We let* $H_0 = h \in \mathbb{K}_0[x]$, *and for each* $i = 1, \ldots, t$, *we have precomputed a monic normal factor* $H_i \in \mathbb{K}_i[x]$ *of* H_{i-1}. *We let* $n_i = \deg H_i = n/(m_1 \cdots m_i)$.
- *For* $i = 1, \ldots, t$, *we have the isomorphism*

$$\mathbb{K}_{i-1}[x]/(H_{i-1}(x)) \cong \mathbb{K}_{i-1}[z, x]/\left(\mu_i(z), \breve{H}_i(z, x)\right),$$

where $\breve{H}_i(z,x)$ is a shorthand for $\Lambda_i^{-1}(H_i)$ and we assume we have precomputed a polynomial $a_i \in \mathbb{K}_{i-1}[x]_{<n_i}$ such that

$$\begin{aligned} \breve{H}_i(a_i(x),x) &= 0 \mod H_{i-1}(x) \\ \mu_i(a_i(x)) &= 0 \mod H_{i-1}(x). \end{aligned}$$

5.3. Modular composition using composition towers

Given $h \in \mathbb{K}[x]$ monic and separable along with a composition tower $(\mathbb{K}_i)_{i\leqslant t}$, we may now apply Algorithm 4.1 recursively. Unrolling the recursive calls yields the following algorithm for modular composition.

Algorithm 5.1

Input. $f, g, h \in \mathbb{K}[x]$ of degrees $<n, <n, n$ and a composition tower $(\mathbb{K}_i)_{i\leqslant t}$ for h.

Output. $f \circ g \operatorname{rem} h$, and the characteristic polynomial of g modulo h.

1. Set $F_0 := f$ and $G_0 := g$.
2. For $i = 1, \ldots, t$, compute $G_i(x) := \Lambda_i(G_{i-1}) \operatorname{rem} H_i$ in $\mathbb{K}_i[x]$.
3. Let $\chi_t = x - G_t$ be the characteristic polynomial of G_t modulo H_t over $\mathbb{K}_t$.
4. For $i = t-1, \ldots, 0$ do
   Compute $\chi_i(x) := \operatorname{Res}_z(\Lambda_{i+1}^{-1}(\chi_{i+1}(x))(z,x), \mu_{i+1}(z))$.
   — *Notice that χ_i is the characteristic polynomial of G_i modulo H_i over $\mathbb{K}_i$.*
5. For $i = 1, \ldots, t$, compute $F_i := \Lambda_i(F_{i-1}) \operatorname{rem} \chi_i$ in $\mathbb{K}_i[x]$.
6. Let $\rho_t := F_t$
7. For $i = t-1, \ldots, 0$ do
   Compute $\rho_i(x) := \Lambda_{i+1}^{-1}(\rho_{i+1})(a_{i+1}(x), x) \operatorname{rem} H_i(x)$.
   — *Notice that $\rho_i = F_i \circ G_i \operatorname{rem} H_i$.*
8. Return ρ_0 and χ_0.

THEOREM 5.3. *Algorithm 5.1 is correct and takes*

$$O\left(\sum_{i=1}^{t} \mathsf{M}_{\mathbb{K}_{i-1}/\mathbb{K}}(m_i n_{i-1}) \log m_i + \sum_{i=1}^{t} \mathsf{M}_{\mathbb{K}_i/\mathbb{K}}(m_i n_i) + \sum_{i=1}^{t} m_i \mathsf{D}_{\mathbb{K}_{i-1}/\mathbb{K}} + 4 \sum_{i=1}^{t} n_i \mathsf{L}_{\mathbb{K}_i/\mathbb{K}} \right)$$

operations in $\mathbb{K}$.

Proof. Since $\deg G_i < \deg H_i = n_i$ for $i = 0, \ldots, n$, the computation of $\Lambda_i(G_{i-1}(x))$ in step 2 can be done in time $n_i \mathsf{L}_{\mathbb{K}_i/\mathbb{K}}$. Using the pre-inverse of $H_i(x)$, the computation of the remainder of its division by $H_i(x)$ can be done in time $O(\mathsf{M}_{\mathbb{K}_i/\mathbb{K}}(m_i n_i))$. Step 2 therefore amounts to

$$O\left(\sum_{i=1}^{t} \mathsf{M}_{\mathbb{K}_i/\mathbb{K}}(m_i n_i) \right) + \sum_{i=1}^{t} n_i \mathsf{L}_{\mathbb{K}_i/\mathbb{K}}$$

operations in $\mathbb{K}$ and similarly for step 5.

The computation of the resultant $\operatorname{Res}_z(\Lambda_{i+1}^{-1}(\chi_{i+1}(x))(z,x), \mu_{i+1}(z))$ in step 4 can be performed in time $O(\mathsf{M}_{\mathbb{K}_i/\mathbb{K}}(m_{i+1}^2 n_{i+1}) \log m_{i+1} + m_{i+1} \mathsf{D}_{\mathbb{K}_i/\mathbb{K}})$ by Proposition 2.2. It follows that the complete step 4 requires

$$O\left(\sum_{i=1}^{t} (\mathsf{M}_{\mathbb{K}_{i-1}/\mathbb{K}}(m_i n_{i-1}) \log m_i + m_i \mathsf{D}_{\mathbb{K}_{i-1}/\mathbb{K}}) \right) + \sum_{i=1}^{t} n_i \mathsf{L}_{\mathbb{K}_i/\mathbb{K}}$$

operations in $\mathbb{K}$. In step 7, the naive evaluation of $\Lambda_{i+1}^{-1}(\varrho_{i+1}(x))$ at $(a_{i+1}(x), x)$ modulo $H_i(x)$ using Horner's method requires $O(m_{i+1}\,\mathsf{M}_{\mathbb{K}_i/\mathbb{K}}(n_i))$ operations in $\mathbb{K}$. Consequently, step 7 requires

$$O\left(\sum_{i=1}^{t} m_i\,\mathsf{M}_{\mathbb{K}_{i-1}/\mathbb{K}}(n_{i-1})\right) + \sum_{i=1}^{t} n_i\,\mathsf{L}_{\mathbb{K}_i/\mathbb{K}}$$

operations in $\mathbb{K}$. The conclusion follows by adding up the above bounds for the costs of the individual steps. □

Remark 5.4. For certain applications, it might be useful to generalize the algorithm to the case when h is not necessarily irreducible. In that case, we assume that the tower $(\mathbb{K}_i)_{i\leqslant t}$ is only a "partial composition tower", meaning that we no longer require that $n_t = 1$. In Algorithm 5.1, we then need to make two adjustments:

- In step 3, we compute χ_t to be the characteristic polynomial of G_t modulo H_t over $\mathbb{K}_t$.
- In step 6, we compute $\varrho_t := F_t \circ G_t \operatorname{rem} H_t$ in $\mathbb{K}_t[x]$.

These computations lead to an additional term $\mathsf{Q}_{\mathbb{K}_t/\mathbb{K}}(n_t) + \mathsf{C}_{\mathbb{K}_t/\mathbb{K}}(n_t)$ in the complexity bound of Theorem 5.3.

6. TRIANGULAR TOWERS

Assume that we are given a tower

$$\mathbb{K} \subset \mathbb{K}[\alpha_1] \subset \cdots \subset \mathbb{K}[\alpha_1, \ldots, \alpha_t]$$

of finite fields such that $m_i = [\mathbb{K}[\alpha_1, \ldots, \alpha_i] : \mathbb{K}[\alpha_1, \ldots, \alpha_{i-1}]] \geqslant 2$ for each i. One obvious way to represent an element of $\mathbb{K}_i = \mathbb{K}[\alpha_1, \ldots, \alpha_i]$ is to write it as $u(\alpha_1, \ldots, \alpha_i)$, where $u \in \mathbb{K}[z_1, \ldots, z_i]$ is a polynomial with $\deg_{z_1} u < m_1, \ldots, \deg_{z_i} u < m_i$. An effective tower that uses this representation for the elements of the fields $\mathbb{K}_i$ is called a *triangular tower*. For such towers, the costs of the upward and downward conversions are zero. Throughout this section it will be convenient to make the relatively harmless assumption that $n \log n = O(\mathsf{M}_{\mathbb{A}}(n))$ for all effective rings $\mathbb{A}$. We always assume available the following precomputed data:

PRE-T1. For all $i \leqslant t$, the pre-inverse of μ_i.

6.1. Complexity analysis for triangular towers

LEMMA 6.1. *Let $(\mathbb{K}_i)_{i\leqslant t}$ be a triangular tower. Then*

$$\mathsf{M}_{\mathbb{K}_i/\mathbb{K}}(n) \;\leqslant\; O(7^i\,\mathsf{M}_{\mathbb{K}}(m_1 \cdots m_i n)),$$

for all $i \in \{1, \ldots, t\}$ and $n \in \mathbb{N}$.

Proof. Let us first show how to reduce polynomial multiplication over $\mathbb{K}_i$ to polynomial multiplication over $\mathbb{K}_{i-1}$. So consider two polynomials u and v in $\mathbb{K}_i[t]_{<n}$, represented as polynomials in $\mathbb{K}_{i-1}[z]_{<m_i}[t]_{<n}$ evaluated at $z = \alpha_i$. We may compute their product in $\mathbb{K}_i[t]_{<2n}$ as follows: we first substitute $t := z^{2m_i}$ in u and v, which yields two polynomials $\tilde{u}, \tilde{v} \in \mathbb{K}_{i-1}[z]_{<2m_i n}$. We next compute their product in $\tilde{w} \in \mathbb{K}_{i-1}[z]_{<4m_i n}$. Now $(u\,v)_k = (\tilde{w}_{2m_i k} + \cdots + \tilde{w}_{2m_i(k+1)-1}\, z^{2m_i-1}) \operatorname{rem} \mu_i$ for each $k \in \{0, \ldots, 2n-1\}$. Since each remainder can be computed using two multiplications of elements in $\mathbb{K}_{i-1}[z]_{<m_i}$ using the pre-inverse of μ_i, we obtain

$$\mathsf{M}_{\mathbb{K}_i/\mathbb{K}}(n) \;\leqslant\; \mathsf{M}_{\mathbb{K}_{i-1}/\mathbb{K}}(2m_i n) + 4n\,\mathsf{M}_{\mathbb{K}_{i-1}/\mathbb{K}}(m_i) + c_0 m_1 \cdots m_i n,$$

for a sufficiently large constant c_0 independent of i. On the other hand, applying Karatsuba's trick, there exists a constant c_1 such that $\mathsf{M}(2n) \leqslant 3\,\mathsf{M}(n) + c_1 n$ holds for all n.

If $i=1$ then we have $\mathsf{M}_{\mathbb{K}_1/\mathbb{K}}(n) \leqslant \mathsf{M}_{\mathbb{K}}(2m_1n)+4\mathsf{M}_{\mathbb{K}}(m_1n)+c_0m_1n$, which yields

$$\mathsf{M}_{\mathbb{K}_1/\mathbb{K}}(n) \leqslant 7\,\mathsf{M}_{\mathbb{K}}(m_1n)+(c_0+c_1)m_1n.$$

We claim that

$$\mathsf{M}_{\mathbb{K}_i/\mathbb{K}}(n) \leqslant 7^i\mathsf{M}_{\mathbb{K}}(m_1\cdots m_in)+i\,7^{i-1}(c_0+c_1)m_1\cdots m_in.$$

The proof is done by induction assuming the inequality holds for $i-1$:

$$\begin{aligned}
\mathsf{M}_{\mathbb{K}_i/\mathbb{K}}(n) &\leqslant 7^{i-1}(3\,\mathsf{M}_{\mathbb{K}}(m_1\cdots m_in)+c_1m_1\cdots m_in)+2(i-1)\,7^{i-2}(c_0+c_1)m_1\cdots m_in\\
&\quad +4\,(7^{i-1}\,\mathsf{M}_{\mathbb{K}}(m_1\cdots m_in)+(i-1)\,7^{i-2}(c_0+c_1)m_1\cdots m_in)\\
&\quad +c_0m_1\cdots m_in\\
&\leqslant 7^i\mathsf{M}_{\mathbb{K}}(m_1\cdots m_in)\\
&\quad +(7^{i-1}c_1+2(i-1)\,7^{i-2}(c_0+c_1)+4(i-1)\,7^{i-2}(c_0+c_1)+c_0)\,m_1\cdots m_in\\
&\leqslant 7^i\mathsf{M}_{\mathbb{K}}(m_1\cdots m_in)+(7^{i-1}+2(i-1)\,7^{i-2}+4(i-1)\,7^{i-2})(c_0+c_1)m_1\cdots m_in\\
&\leqslant 7^i\mathsf{M}_{\mathbb{K}}(m_1\cdots m_in)+i\,7^{i-1}(c_0+c_1)m_1\cdots m_in.
\end{aligned}$$

□

Remark 6.2. We do not claim the latter constant 7 to be optimal in the latter lemma, but it is sufficient for our purposes.

LEMMA 6.3. *Let $(\mathbb{K}_i)_{i\leqslant t}$ be a triangular tower. Then inverting an element in $\mathbb{K}_i$ may be done with $O(7^i\mathsf{M}_{\mathbb{K}}(m_1\cdots m_i)\log\bar{m})$ operations in $\mathbb{K}$.*

Proof. By Proposition 2.2 the inverse of an element in $\mathbb{K}_i$ takes $m_i\,\mathsf{D}_{\mathbb{K}_{i-1}/\mathbb{K}}+O(\mathsf{M}_{\mathbb{K}_{i-1}/\mathbb{K}}(m_i)\log m_i)$ operations in $\mathbb{K}$. Combined with the previous lemma, we obtain

$$\mathsf{D}_{\mathbb{K}_i/\mathbb{K}}\leqslant m_i\,\mathsf{D}_{\mathbb{K}_{i-1}/\mathbb{K}}+c\,7^{i-1}\mathsf{M}_{\mathbb{K}}(m_1\cdots m_i)\log m_i.$$

for some sufficiently large constant c. This yields the claimed bound. □

PROPOSITION 6.4. *Let $(\mathbb{K}_i)_{i\leqslant t}$ be a triangular composition tower for $h\in\mathbb{K}[x]$ with $\deg h=n=m_1\cdots m_t$. Given $f,g\in\mathbb{K}[x]_{<n}$, we may then compute $f\circ g \operatorname{rem} h$ and the characteristic polynomial of g modulo h using*

$$O(7^t\mathsf{M}_{\mathbb{K}}(\bar{m}n)\log\bar{m})$$

operations in $\mathbb{K}$.

Proof. By Lemma 6.1 we have

$$\begin{aligned}
&\sum_{i=1}^{t}\mathsf{M}_{\mathbb{K}_{i-1}/\mathbb{K}}(m_in_{i-1})\log m_i+\sum_{i=1}^{t}\mathsf{M}_{\mathbb{K}_i/\mathbb{K}}(m_in_i)\\
&=O\left(\sum_{i=1}^{t}7^{i-1}\mathsf{M}_{\mathbb{K}}(m_in)\log m_i+\sum_{i=1}^{t}7^i\mathsf{M}_{\mathbb{K}}(m_in)\right)=O(7^t\mathsf{M}_{\mathbb{K}}(\bar{m}n)\log\bar{m}).
\end{aligned}$$

Then Lemma 6.3 gives

$$\sum_{i=1}^{t}m_i\,\mathsf{D}_{\mathbb{K}_{i-1}/\mathbb{K}}=O\left(\sum_{i=1}^{t}m_i7^{i-1}\mathsf{M}_{\mathbb{K}}(m_1\cdots m_{i-1})\log\bar{m}\right)=O(7^t\mathsf{M}_{\mathbb{K}}(n)\log\bar{m}).$$

The conclusion follows from Theorem 5.3. □

6.2. Smooth degrees over finite fields

Recall that an integer n is said to be *b-smooth* whenever all its prime factors are at most b.

LEMMA 6.5. *Let $\varepsilon > 0$. If n is n^ε-smooth, then there exist $m_1, \ldots, m_t$ such that $n = m_1 \cdots m_t$ and $n^{\varepsilon/2} < m_i \leqslant n^\varepsilon$ for all $1 \leqslant i \leqslant t-1$, where $t < 2/\varepsilon + 1$.*

Proof. It suffices to gather prime factors, counted with multiplicities, into $t-1$ products in the range $(n^{\varepsilon/2}, n^\varepsilon]$. Then $n^{\varepsilon(t-1)/2} < m_1 \cdots m_{t-1} \leqslant n$ implies $\varepsilon(t-1) < 2$. □

COROLLARY 6.6. *Let $\varepsilon > 0$. If n is n^ε-smooth and given a suitable triangular decomposition tower for h of degree n, then one composition or one characteristic polynomial modulo h may be computed using $\tilde{O}\left(\varepsilon\, n^{1+\varepsilon+\frac{2\log 7}{\varepsilon \log n}}\right)$ operations in $\mathbb{K}$.*

Proof. We appeal to the previous lemma to construct the integer sequence $m_1, \ldots, m_t$ for which we precompute a triangular decomposition tower for h. The cost of Proposition 6.4 simplifies to

$$O(7^t\, \mathsf{M}_{\mathbb{K}}(n^{1+\varepsilon}) \log n^\varepsilon) = \tilde{O}\left(\varepsilon\, n^{1+\varepsilon+\frac{2\log 7}{\varepsilon \log n}}\right).$$ □

Notice that $\varepsilon = \eta(n) = \sqrt{\frac{2\log 7}{\log n}}$ minimizes the latter exponent, which leads to the cost $\tilde{O}(n^{1+2\eta(n)}) = n^{1+o(1)}$ provided that n is $n^{\eta(n)}$-smooth.

Remark 6.7. It is well known that the number of n^ε-smooth integers below an integer n is asymptotically equal to $n\,\rho(1/\varepsilon) + O(n/\log n)$, where ρ is the Dickman–de Bruijn function defined by $t\,\rho'(t) + \rho(t-1) = 0$ with initial condition $\rho(t) = 1$ for all $t \in [0,1]$ (see [39] for the original proof). For $\varepsilon = 1/2$, we have $\rho(2) \approx 0.31$. Then ρ decreases rapidly with $\rho(3) \approx 0.049$, $\rho(4) \approx 0.0049$, etc. Another important consequence of Proposition 6.4 is the following: for more than 30% of large values of n, modular compositions and characteristic polynomials for irreducible h of degree n over $\mathbb{F}_q$ may be computed using $\tilde{O}(n^{3/2})$ operations in $\mathbb{F}_q$.

6.3. Cyclic modulus of prime degree over a finite field

An interesting application of Proposition 6.4 concerns cyclic moduli $h(x) = x^n - 1$ in $\mathbb{F}_q[x]$, where n is a prime number different from the characteristic p.

LEMMA 6.8. *If n is a prime number different from p, and if l divides $n-1$, then the degrees of the irreducible factors of $x^n - 1$ in $\mathbb{F}_{q^l}[x]$ divide $(n-1)/l$.*

Proof. Assume that l divides $n-1$, and write $m = (n-1)/l$. It is well known that the polynomial $x^{(q^l)^m} - x$ is the product of the monic irreducible polynomials of $\mathbb{F}_{q^l}$ whose degrees divide m. We obtain $\gcd\left(x^{(q^l)^m} - x, x^n - 1\right) = \gcd\left(x^{q^{n-1} \bmod n} - x, x^n - 1\right) = x^n - 1$, by using Fermat's little theorem, which asserts that $q^{n-1} = 1 \bmod n$. □

COROLLARY 6.9. *Let $\varepsilon > 0$. If n is a prime number different from p, and if $n-1$ is n^ε-smooth, then we may precompute suitable triangular decomposition towers for all irreducible factors of h, so one composition or characteristic polynomial modulo $h \in \mathbb{F}_q[x]$ may be obtained using $\tilde{O}\left(\varepsilon\, n^{1+\varepsilon+\frac{2\log 7}{\varepsilon \log n}}\right)$ operations in $\mathbb{F}_q$.*

Proof. The modulus $h(x) = x^n - 1$ is separable, and the precomputations first involve the irreducible factorization of $h(x)$ into $h_1, \ldots, h_s$, whose respective degrees $n_1, \ldots, n_s$ divide $n-1$.

Since $n-1$ is n^ε-smooth then each n_i is also n^ε-smooth, hence the cost of Proposition 6.4 for h_i simplifies to

$$\tilde{O}\left(\varepsilon_i\, n_i^{1+\varepsilon_i+\frac{2\log 7}{\varepsilon_i \log n_i}}\right) = \tilde{O}\left(\varepsilon\, n^\varepsilon\, n_i^{1+\frac{2\log 7}{\varepsilon \log n}}\right),$$

where $\varepsilon_i = \varepsilon \log n / \log n_i$. The total cost for all h_i is thus $\tilde{O}\Big(\varepsilon\, n^{1+\varepsilon+\frac{2\log 7}{\varepsilon \log n}}\Big)$. Then we appeal to Proposition 3.2 which leads to the following cost for one composition or characteristic polynomial modulo h: $O(\mathsf{M}(n)\log n) + \sum_{i=1}^{s} (\mathsf{Q}_{\mathbb{K}}(n_i) + \mathsf{C}_{\mathbb{K}}(n_i))$. □

7. PRIMITIVE TOWERS

Proposition 6.4 shows that the overhead of triangular set arithmetic rapidly grows with the height t of the tower. In this section we consider an alternative representation for elements in the fields $\mathbb{K}_i$. This representation allows for faster multiplication inside the fields $\mathbb{K}_i$, but the upward and downward conversions may become more expensive. One major goal of this section is to provide a more precise analysis of the cost of these conversions and to isolate particular situations in which they can be computed fast.

7.1. Primitive towers

An effective tower $(\mathbb{K}_i)_{i\leqslant t}$ with $\mathbb{K}_i \cong \mathbb{K}_{i-1}[\alpha_i]$ is said to be *primitive* if $\mathbb{K}_i = \mathbb{K}[\alpha_i]$ for each i. In that case, we assume that we precomputed the minimal polynomial ν_i of each α_i over $\mathbb{K}$. It follows that

$$\mathsf{M}_{\mathbb{K}_i/\mathbb{K}}(n) = O(\mathsf{M}_{\mathbb{K}}(m_1 \cdots m_i n)) \tag{7.1}$$

$$\mathsf{D}_{\mathbb{K}_i/\mathbb{K}} = O(\mathsf{M}_{\mathbb{K}}(m_1 \cdots m_i)\log(m_1 \cdots m_i)), \tag{7.2}$$

for $i = 0, \ldots, n$. On the other hand, the upward and downward conversions are more expensive than in the case of triangular towers. The following consequence of (7.1), (7.2) and Theorem 5.3 will be of frequent use.

LEMMA 7.1. *For a primitive composition tower for h with $\mathbb{K}_t = \mathbb{K}[x]/(h(x))$, Algorithm 5.1 is correct and takes*

$$O(\mathsf{M}_{\mathbb{K}}(\bar{m} n)\log n) + 4\sum_{i=1}^{t} n_i \mathsf{L}_{\mathbb{K}_i/\mathbb{K}}$$

operations in $\mathbb{K}$.

Proof. We have

$$\sum_{i=1}^{t} \mathsf{M}_{\mathbb{K}_{i-1}/\mathbb{K}}(m_i n_{i-1})\log m_i = O(\mathsf{M}_{\mathbb{K}}(\bar{m} n)\log n)$$

$$\sum_{i=1}^{t} \mathsf{M}_{\mathbb{K}_i/\mathbb{K}}(m_i n_i) = O(\mathsf{M}_{\mathbb{K}}(\bar{m} n)\log n)$$

$$\sum_{i=1}^{t} m_i \mathsf{D}_{\mathbb{K}_{i-1}/\mathbb{K}} = \sum_{i=1}^{t} m_i \mathsf{M}_{\mathbb{K}}(m_1 \cdots m_{i-1})\log(m_1 \cdots m_{i-1}) = O(\mathsf{M}_{\mathbb{K}}(n)\log n),$$

so the conclusion follows from Theorem 5.3. □

7.2. Arbitrary primitive elements

For computing with arbitrary primitive towers, we recall that we always assume available the following precomputed data:

PRE-P1. For all $i \geqslant j$, the minimal polynomial of α_i over $\mathbb{K}_j$,

PRE-P2. For all $i \geqslant j \geqslant 1$, the minimal polynomial of α_i over $\mathbb{K}_{j-1}[\alpha_j]$,

PRE-P3. For all $i \geqslant j \geqslant 1$, the polynomial expression of α_j in terms of α_i over $\mathbb{K}_{j-1}$.

Assume that $\mathbb{K}_i = \mathbb{K}[\alpha_i]$ for some arbitrary primitive element α_i. For a natural morphism $\mathbb{A} \to \mathbb{B}$ for $\mathbb{K}$-algebras $\mathbb{A}$ and $\mathbb{B}$, let $\mathsf{C}_{\mathbb{K}}(\mathbb{A} \to \mathbb{B})$ denote the cost of applying the morphism once in terms of the number of required operations in $\mathbb{K}$.

LEMMA 7.2. *Modulo precomputations, we have for all* $1 \leqslant j < i \leqslant t$,

$$\mathsf{C}_{\mathbb{K}}(\mathbb{K}_j[\alpha_i] \to \mathbb{K}_{j-1}[\alpha_i]) = m_{j+1} \cdots m_i \, \mathsf{C}_{\mathbb{K}}(\mathbb{K}_j \to \mathbb{K}_{j-1}[\alpha_j]) + O(m_j \mathsf{M}_{\mathbb{K}}(m_1 \cdots m_i)) \tag{7.3}$$

$$\mathsf{C}_{\mathbb{K}}(\mathbb{K}_{j-1}[\alpha_i] \to \mathbb{K}_j[\alpha_i]) = m_{j+1} \cdots m_i \, \mathsf{C}_{\mathbb{K}}(\mathbb{K}_{j-1}[\alpha_j] \to \mathbb{K}_j) + O(m_j \mathsf{M}_{\mathbb{K}}(m_1 \cdots m_i)). \tag{7.4}$$

Proof. Let $\tilde{u}(\alpha_j, \alpha_i) \in \mathbb{K}_j[\alpha_i]$ with $\tilde{u} \in \mathbb{K}[z_j]_{<m_1 \cdots m_j}[z_i]_{<m_{j+1} \cdots m_i}$. We may convert $\tilde{u}(\alpha_j, \alpha_i)$ into $u(\alpha_j, \alpha_i) \in \mathbb{K}_{j-1}[\alpha_j, \alpha_i]$ with $u \in \mathbb{K}_{j-1}[z_j]_{<m_j}[z_i]_{<m_{j+1} \cdots m_i}$ using $m_{j+1} \cdots m_i \, \mathsf{C}_{\mathbb{K}}(\mathbb{K}_j \to \mathbb{K}_{j-1}[\alpha_j])$ operations in $\mathbb{K}$. Then we use the precomputed polynomial $b \in \mathbb{K}_{j-1}[z_i]_{<m_j \cdots m_i}$ with $\alpha_j = b(\alpha_i)$, and also the minimal polynomial $w \in \mathbb{K}_{j-1}[z_i]$ of α_i over $\mathbb{K}_{j-1}$, which has degree $m_j \cdots m_i$. We now compute $\rho(x) = u(b(x), x) \operatorname{rem} w(x)$ using Horner's method. This requires $O(m_j \mathsf{M}_{\mathbb{K}_{j-1}/\mathbb{K}}(m_j \cdots m_i)) = O(m_j \mathsf{M}_{\mathbb{K}}(m_1 \cdots m_i))$ operations in $\mathbb{K}$. The evaluation $\rho(\alpha_i)$ is the natural image of $\tilde{u}(\alpha_j, \alpha_i)$ in $\mathbb{K}_{j-1}[\alpha_i]$. This proves (7.3).

For the opposite direction, consider $u(\alpha_i) \in \mathbb{K}_{j-1}[\alpha_i]$ and reinterpret $u(\alpha_i)$ as an element of $\mathbb{L}[\alpha_i]$ with $\mathbb{L} = \mathbb{K}_{j-1}[\alpha_j]$ and $u \in \mathbb{L}[z_i]_{<m_j \cdots m_i}$. We use the precomputed minimal polynomial $\theta \in \mathbb{L}[z_i]$ of α_i over $\mathbb{L}$, which has degree $m_{j+1} \cdots m_i$. We next compute $w = u \operatorname{rem} \theta \in \mathbb{L}[z_i]$. This requires $O(m_j \mathsf{M}_{\mathbb{L}/\mathbb{K}}(m_{j+1} \cdots m_i)) = O(m_j \mathsf{M}_{\mathbb{K}}(m_1 \cdots m_i))$ operations in $\mathbb{K}$. We finally convert $w(\alpha_i)$ coefficientwise into an element $\tilde{w}(\alpha_i)$ of $\mathbb{K}_j[\alpha_i]$. This requires $m_{j+1} \cdots m_i \, \mathsf{C}_{\mathbb{K}}(\mathbb{K}_{j-1}[\alpha_j] \to \mathbb{K}_j)$ operations in $\mathbb{K}$ and yields the natural image of $\tilde{u}(\alpha_i)$ in $\mathbb{K}_j[\alpha_i]$. This completes the proof of (7.4). □

LEMMA 7.3. *Modulo precomputations, we have for all* $i \leqslant t$ *and* $j, k \in \{0, \ldots, i\}$,

$$\mathsf{C}_{\mathbb{K}}(\mathbb{K}_j[\alpha_i] \to \mathbb{K}_k[\alpha_i]) = O((m_i + 2 m_{i-1} + \cdots + 2^{i-1} m_1) \mathsf{M}_{\mathbb{K}}(m_1 \cdots m_i)).$$

Proof. It will be convenient to use the following abbreviations:

$$\begin{aligned} \sigma_{j,k} &= m_j + \cdots + m_k \\ \pi_{j,k} &= m_j \cdots m_k \\ \Sigma_{j,k} &= 2^{k-j} m_j + \cdots + 2 m_{k-1} + m_k. \end{aligned}$$

Let c be a constant such that

$$\mathsf{C}_{\mathbb{K}}(\mathbb{K}_j[\alpha_i] \to \mathbb{K}_{j-1}[\alpha_i]) \leqslant \pi_{j+1,i} \, \mathsf{C}_{\mathbb{K}}(\mathbb{K}_j \to \mathbb{K}_{j-1}[\alpha_j]) + c \, m_j \mathsf{M}_{\mathbb{K}}(\pi_{1,i}) \tag{7.5}$$

$$\mathsf{C}_{\mathbb{K}}(\mathbb{K}_{j-1}[\alpha_i] \to \mathbb{K}_j[\alpha_i]) \leqslant \pi_{j+1,i} \, \mathsf{C}_{\mathbb{K}}(\mathbb{K}_{j-1}[\alpha_j] \to \mathbb{K}_j) + c \, m_j \mathsf{M}_{\mathbb{K}}(\pi_{1,i}) \tag{7.6}$$

in the previous lemma and let us show by induction over i that

$$\mathsf{C}_{\mathbb{K}}(\mathbb{K}_j[\alpha_i] \to \mathbb{K}_k[\alpha_i]) \leqslant c \, \Sigma_{1,i} \mathsf{M}_{\mathbb{K}}(\pi_{1,i}). \tag{7.7}$$

For $i = 1$, we have $\mathbb{K}_1[\alpha_1] = \mathbb{K}_0[\alpha_1]$, so all possible conversions are trivial and (7.7) holds. Now assume that the result holds until $i - 1 \geqslant 0$. Let us first consider the case when $j < k < i$. Then (7.6) yields

$$\begin{aligned} \mathsf{C}_{\mathbb{K}}(\mathbb{K}_j[\alpha_i] \to \mathbb{K}_k[\alpha_i]) &\leqslant \sum_{l=j+1}^{k} \mathsf{C}_{\mathbb{K}}(\mathbb{K}_{l-1}[\alpha_i] \to \mathbb{K}_l[\alpha_i]) \\ &\leqslant \sum_{l=j+1}^{k} \pi_{l+1,i} \, \mathsf{C}_{\mathbb{K}}(\mathbb{K}_{l-1}[\alpha_l] \to \mathbb{K}_l) + c \, \sigma_{j+1,k} \mathsf{M}_{\mathbb{K}}(\pi_{1,i}) \\ &\leqslant \sum_{l=j+1}^{k} c \, \pi_{l+1,i} \Sigma_{1,l} \mathsf{M}_{\mathbb{K}}(\pi_{1,l}) + c \, \sigma_{j+1,k} \mathsf{M}_{\mathbb{K}}(\pi_{1,i}) \\ &\leqslant \sum_{l=j+1}^{k} c \, \Sigma_{1,l} \mathsf{M}_{\mathbb{K}}(\pi_{1,i}) + c \, \sigma_{j+1,k} \mathsf{M}_{\mathbb{K}}(\pi_{1,i}) \\ &= c \, \mathsf{M}_{\mathbb{K}}(\pi_{1,i}) \left(\sum_{l=j+1}^{k} \Sigma_{1,l} + \sigma_{j+1,k} \right) \\ &\leqslant c \, \mathsf{M}_{\mathbb{K}}(\pi_{1,i}) \Sigma_{1,i}. \end{aligned}$$

Since $\mathbb{K}_i[\alpha_i]=\mathbb{K}_i=\mathbb{K}_0[\alpha_i]$, this also deals with the case when $k<j=i$. If $k<j<i$, then (7.5) yields

$$\mathsf{C}_{\mathbb{K}}(\mathbb{K}_j[\alpha_i]\to\mathbb{K}_k[\alpha_i]) \;\leqslant\; c\,\Sigma_{1,i}\,\mathsf{M}_{\mathbb{K}}(\pi_{1,i})$$

in a similar way. This also deals with the case when $j<k=i$. We conclude by induction. □

COROLLARY 7.4. *Modulo precomputations, we have for all $i\leqslant t$,*

$$\mathsf{L}_{\mathbb{K}_i/\mathbb{K}} \;=\; O((m_i+2\,m_{i-1}+\cdots+2^{i-1}m_1)\,\mathsf{M}_{\mathbb{K}}(m_1\cdots m_i)).$$

COROLLARY 7.5. *Let $(\mathbb{K}_i)_{i\leqslant t}$ be a primitive composition tower for $h\in\mathbb{K}[x]$ with $\deg h=n$. Given $f,g\in\mathbb{K}[x]_{<n}$, we may then compute one composition or characteristic polynomial modulo h using*

$$O(\mathsf{M}_{\mathbb{K}}(\bar{m}\,n)\,(2^t+\log n))$$

operations in $\mathbb{K}$.

Proof. From Corollary 7.4 we deduce

$$\sum_{i=1}^{t} n_i\,\mathsf{L}_{\mathbb{K}_i/\mathbb{K}}=O\left(\sum_{i=1}^{t} m_i+2\,m_{i-1}+\cdots+2^{i-1}m_1\right)\mathsf{M}_{\mathbb{K}}(n)=O(2^t\,\mathsf{M}_{\mathbb{K}}(\bar{m}\,n)),$$

so the conclusion follows from Lemma 7.1. □

Comparing to Proposition 6.4, using primitive towers thus turns out to be more efficient than using triangular towers, although it requires more precomputations. Therefore the costs for the two particular cases studied in sections 6.2 and 6.3 may be revisited and slightly improved.

7.3. Nested towers

We say that a primitive tower $(\mathbb{K}_i)_{i\leqslant t}$ with $\mathbb{K}_i=\mathbb{K}[\alpha_i]$ is a *nested tower*, if there exist $\tau_i,\kappa_i\in\mathbb{K}[z]$ with $\deg\tau_i=m_i$, $\deg\kappa_i=k_i\leqslant m_i$, $\kappa_i(\alpha_i)\neq 0$, τ_i and κ_i coprime, and

$$\tau_i(\alpha_i) \;=\; \alpha_{i-1}\,\kappa_i(\alpha_i)\quad(i=2,\ldots,t).$$

Setting $\tau_1(z)=\nu_1(z)$ and $\kappa_1(z)=z$, this also means that

$$\nu_i \;=\; \kappa_i^{m_1\cdots m_{i-1}}\,\nu_{i-1}(\tau_i/\kappa_i)\qquad(i=1,\ldots,t),$$

and has degree $m_1\cdots m_i$. For this specific type of towers we require the following precomputations:

PRE-N1. $\kappa_i^{2^j}$, $\tau_i^{2^j}$ and $\kappa_i^{-2^j} \bmod \tau_i^{2^j}$ for $2\leqslant i\leqslant t$ and $0\leqslant j<s$, where 2^s is the next power of two of $m_1\cdots m_{i-1}-1$.

LEMMA 7.6. *Given a nested tower $(\mathbb{K}_i)_{i\leqslant t}$, we have*

$$\mathsf{L}_{\mathbb{K}_i/\mathbb{K}} \;=\; O(\mathsf{M}_{\mathbb{K}}(m_1\cdots m_i)\log(m_1\cdots m_i))$$

for all $i\in\{1,\ldots,t\}$.

Proof. Consider an element $u(\alpha_{i-1},\alpha_i)\in\mathbb{K}_{i-1}[\alpha_i]$ with $u\in\mathbb{K}[t]_{<m_1\cdots m_{i-1}}[z]_{<m_i}$. Then we may compute $\tilde{w}(z)=\kappa_i(z)^{m_1\cdots m_{i-1}}\,u(\tau_i(z)/\kappa_i(z),z)$ using $O(m_i\,\mathsf{M}_{\mathbb{K}}(m_1\cdots m_{i-1})\log(m_1\cdots m_{i-1}))$ operations in $\mathbb{K}$, and we have $\tilde{w}(\alpha_i)=\kappa_i(\alpha_i)^{m_1\cdots m_{i-1}}\,\Lambda_i(u(\alpha_{i-1},z))$. This computation follows the natural "divide and conquer" strategy: if $\varphi\in\mathbb{K}[t]_{<l}$ then we split it into $\varphi(t)=\varphi_0(t)+t^{\lfloor l/2\rfloor}\,\varphi_1(t)$ with $\deg\varphi_0<\lfloor l/2\rfloor$, we compute recursively $A(z)=\kappa_i(z)^{\lfloor l/2\rfloor-1}\,\varphi_0(\tau_i(z)/\kappa_i(z))$ and $\kappa_i(z)^{l-\lfloor l/2\rfloor}\varphi_1(\tau_i(z)/\kappa_i(z))$, and deduce

$$\kappa_i(z)^l\,\varphi\left(\frac{\tau_i(z)}{\kappa_i(z)}\right)=\kappa_i(z)^{l-\lfloor l/2\rfloor+1}\,A(z)+\tau_i(z)^{\lfloor l/2\rfloor}\,B(z).$$

Then we compute the inverse of $\kappa_i(\alpha_i)^{m_1\cdots m_{i-1}}$ with $\mathsf{M}_{\mathbb{K}}(m_1\cdots m_i)\log(m_1\cdots m_i)$ additional operations in $\mathbb{K}$.

Conversely, let $w(\alpha_i)\in\mathbb{K}_i$ with $w\in\mathbb{K}[z]_{<m_1\cdots m_i}$. We compute $\tilde{w}(z)=\kappa_i^{2^s}(z)\,w(z)\operatorname{rem}\nu_i(z)$. We are looking for an expansion of the form

$$\tilde{w} \;=\; \kappa_i^{2^s}u_0+u_1\kappa_i^{2^s-1}\tau_i+\cdots+u_{2^s}\tau_i^{2^s-1}\,,$$

where the $u_i\in\mathbb{K}[z]_{<m_i}$. The is done by "divide and conquer":

- Compute $\tilde{w}_0=\kappa_i^{-2^{s-1}}\tilde{w}\bmod\tau_i^{2^{s-1}}$, and recursively compute the expansion
$$w_0=\kappa_i^{2^{s-1}}u_0+u_1\kappa_i^{-2^{s-1}-2}\tau_i+\cdots+u_{2^{s-1}-1}\tau_i^{-2^{s-1}}.$$
- Compute $\tilde{w}_1=\left(\tilde{w}-\tilde{w}_0\kappa_i^{2^{s-1}}\right)\operatorname{quo}\tau_i^{2^{s-1}}$, and recursively compute the expansion
$$\tilde{w}_1=\kappa_i^{2^{s-1}}u_{2^s}+u_1\kappa_i^{-2^{s-1}-2}\tau_i+\cdots+u_{2^s-1}\tau_i^{-2^{s-1}}.$$

In final we have $\tilde{w}=\kappa_i^{2^{s-1}}\tilde{w}_0+\tau_i^{2^{s-1}}\tilde{w}_1$, as required. Thanks to the precomputations this expansion requires $O(\mathsf{M}_{\mathbb{K}}(m_1\cdots m_i)\log(m_1\cdots m_{i-1}))$ operations in $\mathbb{K}$. Then we observe that $w(\alpha_i)=\kappa_i^{-2^s}(\alpha_i)\,\tilde{w}(\alpha_i)=u_0(\alpha_i)+u_1(\alpha_i)\,\alpha_{i-1}+\cdots+u_{2^s-1}(\alpha_i)\,\alpha_{i-1}^{2^s-1}=u(\alpha_{i-1},\alpha_i)$, where $u\in\mathbb{K}[t]_{<2^s}[x]_{<m_i}$ and $u(\alpha_{i-1},z)=\Lambda_i^{-1}(w(\alpha_i))$. A final reduction by $\nu_{i-1}(t)$ takes $O(m_i\,\mathsf{M}(m_1\cdots m_{i-1}))$ operations in $\mathbb{K}$. □

COROLLARY 7.7. *Let $(\mathbb{K}_i)_{i\leqslant t}$ be a nested composition tower for $h\in\mathbb{K}[x]$ with $\deg h=n$. Then we may compute one composition or characteristic polynomial modulo h using*

$$O((\mathsf{M}_{\mathbb{K}}(\bar{m}\,n)+t\,\mathsf{M}_{\mathbb{K}}(n))\log n)$$

operations in $\mathbb{K}$.

Proof. Lemma 7.6 implies

$$\sum_{i=1}^{t} n_i\,\mathsf{L}_{\mathbb{K}_i/\mathbb{K}}=O\left(\sum_{i=1}^{t} n_i\,\mathsf{M}_{\mathbb{K}}(m_1\cdots m_i)\log(m_1\cdots m_{i-1})\right)=O(t\,\mathsf{M}_{\mathbb{K}}(n)\log n).$$

The result now follows from Lemma 7.1. □

The above corollary makes nested composition towers extremely attractive from a complexity point of view. The existence of such towers only depends on $\mathbb{K}$ and the degrees $m_1,\ldots,m_t$, but not on h. A practical way to construct nested towers is to pick random monic $\tau_1,\ldots,\tau_t$ of degrees $m_1,\ldots,m_t$ and to check that $\tau_1\circ\cdots\circ\tau_i$ is irreducible for each $i\in\{1,\ldots,t\}$. We repeat this process for random choices of $\tau_1,\ldots,\tau_t$ until we find a suitable tower. From a heuristic point of view, we will show below that the probability that we eventually obtain a nested tower is non zero in most cases of interest. From a theoretical point of view, the existence problem of nested towers remains an interesting problem.

In order to analyze the probability that random choices of $\tau_1,\ldots,\tau_t$ provide us with a nested tower, we rely on

- the fact that a random polynomial over $\mathbb{K}$ of degree d is irreducible with probability $\approx 1/d$;
- the heuristic assumption that $\tau_1\circ\cdots\circ\tau_i$ is again random for $i\in\{2,\ldots,t\}$.

In this framework, the probability that $\tau_1\circ\cdots\circ\tau_i$ is irreducible for each $i\in\{1,\ldots,t\}$ is given by

$$P \;=\; \frac{1}{m_1^t\,m_2^{t-1}\cdots m_{t-1}^2\,m_t}.$$

On the other hand, if $\mathbb{K}$ has cardinality q, then we have

$$N \;=\; q^{m_1+\cdots+m_t}$$

possible choices for the tuple $(\tau_1, \ldots, \tau_i)$. For a fixed value of n, we maximize P by taking $m_1 \leqslant m_2 \leqslant \cdots \leqslant m_t$. Setting $\mu = (m_1 + \cdots + m_t)/t$, we then have

$$\begin{aligned} P &\geqslant \mu^{-\binom{t}{2}} \\ N &= q^{t\mu}. \end{aligned}$$

The existence of a nested tower over $\mathbb{K}$ with extension degrees $m_1, \ldots, m_t$ is likely whenever $PN \gg 1$. Taking logarithms, this happens as soon as

$$\log q > \frac{t \log \mu}{2\,\mu}.$$

The algorithm for finding $\tau_1 \circ \cdots \circ \tau_i$ needs $O(m_1^t\, m_2^{t-1} \cdots m_t)$ runs before finding a suitable tower. An obvious optimization is to make a better use of successful guesses of $\tau_1, \ldots, \tau_i$ for which $\tau_1 \circ \cdots \circ \tau_j$ is irreducible for all $j \in \{1, \ldots, i\}$: instead of starting everything over after one unsuccessful guess of τ_{i+1}, we try at least $c\, m_1 \cdots m_{i+1}$ times for some fixed constant c. The expected number of guesses then drops to $O(n)$.

Remark 7.8. Taking $\mathbb{K} = \mathbb{F}_2$ and $m_1 = m_2 = m_3 = 5$, it can be checked that there do not exist any polynomials $\tau_1, \tau_2, \tau_3 \in \mathbb{K}[x]$ of degree 5 such that τ_1, $\tau_1 \circ \tau_2$ and $\tau_1 \circ \tau_2 \circ \tau_3$ are all irreducible.

It is an interesting question whether there exist finite fields $\mathbb{K}$ and sequences $m_1, m_2, \ldots$ for which it is possible to construct monic $\tau_1, \tau_2, \ldots \in \mathbb{K}[x]$ of degrees $m_1, m_2, \ldots$ such that $\tau_1 \circ \cdots \circ \tau_i$ is irreducible for each i. The literature contains specific constructions of nested towers over finite fields based on [8, Lemma 1] which relates the irreducibility of $\kappa_i^{m_1 \cdots m_{i-1}}\, \nu_{i-1}(\tau_i / \kappa_i)$ to $\tau_i(z) - \beta\, \kappa_i(z)$ where $\nu_{i-1}(\beta) = 0$. We refer the reader to [35, chapter 3, section 2] for a nice survey. Let us exemplify two useful constructions.

Example 7.9. Following [31, Theorem 4], if $q = 1 \bmod 4$ is a prime power, and $\nu_1(z) = z^2 + b\, z + c \in \mathbb{F}_q$, where $b \neq 0$, and c is a non-zero square and $b^2 - 4\, c$ is a non-square in $\mathbb{F}_q$. Then we may build a nested tower with $\tau_i(z) = x^2 + c$, $\kappa_i(z) = 2\, z$, $[\mathbb{K}_i : \mathbb{K}] = 2^i$. For instance we may take $q = 5$, $b = 3$, and $c = 4$.

Example 7.10. This following construction is also due Kyuregyan [30, Theorem 7]. Let $\nu_1(z)$ be an irreducible polynomial of degree $m_1 \geqslant 1$ over $\mathbb{F}_q$, where m_1 is even if $q = 3 \bmod 4$. Let $b \in \mathbb{F}_q$ such that $\nu_1(-b/2)$ is a non-square in $\mathbb{F}_q$. Then we may take $\tau_i(z) = z^2 + b\, z + b^2/4 - b/2$ and $\kappa_i(z) = 1$ for all $i \geqslant 2$. In this way we have $[\mathbb{K}_i : \mathbb{K}] = m_1\, 2^{i-1}$. For instance with $q = p = 7$, we may take $\nu_1(z) = z^4 + z + 1$, $b = 3$, $\tau_i(z) = z^2 + 3\, z + 6$. Notice that computations with nested towers simplifies a bit in this situation where $\kappa_i = 1$ for all $i \geqslant 2$.

Example 7.11. One may wonder whether the above examples admit generalizations for which the τ_i are of odd degree $\geqslant 3$. A non trivial candidate example of this kind is the sequence $\nu_k := (x^3 + x^2 + 1)^{\circ k}$. We verified ν_k to be irreducible over $\mathbb{F}_2$ for $k \leqslant 11$, but does this hold for all k?

7.4. Composed towers

Over finite fields, the situation when the m_i are pairwise coprime may be exploited to construct special towers, relying on the following well-known lemma:

LEMMA 7.12. *Let $\mathbb{K}$ be a finite field and consider two monic irreducible polynomials ν and λ in $\mathbb{K}[x]$ whose degrees are coprime. Then the* composed product $\nu \odot \lambda \in \mathbb{K}[z]$, *defined by*

$$(\nu \odot \lambda)(z) = \prod_{\nu(\zeta)=0} \prod_{\lambda(\xi)=0} (z - \zeta\, \xi) = \prod_{\nu(\zeta)=0} \zeta^{\deg \lambda}\, \lambda(\zeta^{-1} z) = \prod_{\lambda(\xi)=0} \xi^{\deg \nu}\, \nu(\xi^{-1} z),$$

is irreducible in $\mathbb{K}[x]$ *and of degree* $\deg \nu \deg \lambda$.

Proof. See for instance [4]. □

Remark 7.13. Given primitive elements α and β of coprime degrees d and e over $\mathbb{K}$, an alternative way to state the lemma is that $\alpha\,\beta$ is again a primitive element of degree $d\,e$ over $\mathbb{K}$.

Remark 7.14. Composed products can be computed in softly linear time [2], by means of the Newton–Girard identities (see also [15] for handling these identities in small characteristic).

Let $(\mathbb{K}_i)_{i\leqslant t}$ be a primitive tower and let α_i, m_i and ν_i be as usual. We say that $(\mathbb{K}_i)_{i\leqslant t}$ is a *composed tower* if the m_i are pairwise coprime and if there exist monic irreducible polynomials $\lambda_1, \ldots, \lambda_t \in \mathbb{K}[z]$ of degrees $m_1, \ldots, m_t$ such that $\nu_1 = \lambda_1$ and $\nu_i = \nu_{i-1} \odot \lambda_i$ for $i = 2, \ldots, t$.

In that case, the minimal polynomial μ_i of α_i over $\mathbb{K}_{i-1}$ is given by $\mu_1(z) = \nu_1(z)$ and $\mu_i(z) = \alpha_{i-1}^{m_i}\, \lambda_i(\alpha_{i-1}^{-1}\, z)$ for $i \geqslant 2$: if μ_i were reducible over $\mathbb{K}_{i-1}$ then m_i would have a proper gcd with $[\mathbb{K}_{i-1} : \mathbb{K}] = m_1 \cdots m_{i-1}$ which is impossible (see Proposition 4.1).

By construction, we thus have $\nu_{i-1}(\alpha_{i-1}) = 0$ and $\mu_i(\alpha_i) = 0$. For each $i \geqslant 2$, let ξ_i be a root of λ_i and $\alpha_i = \xi_i\, \alpha_{i-1}$, so that $\lambda_i(\alpha_{i-1}^{-1}\, \alpha_i) = 0$. For such composed towers, we assume that the following precomputations have been done for $i \in \{1, \ldots, t\}$:

PRE-C1. $z^{-1} \bmod \lambda_i(z)$,

PRE-C2. the trace map of $\mathbb{K}[z]/\lambda_i(z)$ over $\mathbb{K}$ (as a vector of $\mathbb{K}^{m_i}$),

PRE-C3. $\frac{\nu_i(z)}{\nu_{i-1}(t^{-1} z)}$ and its inverse modulo $\nu_{i-1}(t^{-1} z)$ and $\lambda_i(t)$.

LEMMA 7.15. *Let* $(\mathbb{K}_i)_{i\leqslant t}$ *be a composed tower. Then we have*

$$\mathsf{L}_{\mathbb{K}_i/\mathbb{K}} \;=\; O(\mathsf{M}_{\mathbb{K}}(m_1 \cdots m_{i-1}\, m_i^2)),$$

for all $i \in \{1, \ldots, t\}$.

Proof. Let $u(\alpha_i) \in \mathbb{K}[\alpha_i]$ with $u \in \mathbb{K}[z]_{<m_1 \cdots m_i}$. We wish to compute $w(\alpha_{i-1}, z) = \Lambda_i^{-1}(u(\alpha_i))$ with $w \in \mathbb{K}[t]_{<m_1 \cdots m_{i-1}}[z]_{<m_i}$. For this purpose we first calculate

$$\tilde{w}(t, z) \;=\; (u(t\, z) \bmod \lambda_i(t)) \bmod \nu_{i-1}(z)$$

using $O(m_1 \cdots m_i\, \mathsf{M}_{\mathbb{K}}(m_i) + m_i\, \mathsf{M}_{\mathbb{K}}(m_1 \cdots m_i))$ operations in $\mathbb{K}$. Then

$$w(\alpha_{i-1}, z) \;=\; \tilde{w}(\alpha_{i-1}^{-1}\, z, \alpha_{i-1}) \bmod \mu_i(z)$$

can be computed using $O(m_i\, \mathsf{M}(m_1 \cdots m_{i-1}))$ additional operations.

Conversely, let $w(\alpha_{i-1}, z) \in \mathbb{K}[\alpha_{i-1}][z]$ with $w \in \mathbb{K}[t]_{<m_1 \cdots m_{i-1}}[z]_{<m_i}$. The direct image $u(\alpha_i) = \Lambda_i(w(\alpha_{i-1}, z))$ with $u \in \mathbb{K}[z]_{<m_1 \cdots m_i}$ is obtained *via* Chinese remaindering:

$$u(z) = \sum_{\lambda_i(\xi)=0} w(\xi^{-1} z, z)\, \frac{\nu_i(z)}{\nu_{i-1}(\xi^{-1} z)} \left(\left(\frac{\nu_i(z)}{\nu_{i-1}(\xi^{-1} z)} \right)^{-1} \bmod \nu_{i-1}(\xi^{-1} z) \right).$$

In fact we just verify that $u(\alpha_i) = w(\xi_i^{-1}\, \alpha_i, \alpha_i) = w(\alpha_{i-1}, \alpha_i)$. So if we let $\mathbb{L}_i = \mathbb{K}[t]/(\lambda_i(t))$, then we may calculate

$$u(z) = \mathrm{Tr}_{\mathbb{L}_i/\mathbb{K}} \left(w(t^{-1} z, z)\, \frac{\nu_i(z)}{\nu_{i-1}(t^{-1} z)} \left(\left(\frac{\nu_i(z)}{\nu_{i-1}(t^{-1} z)} \right)^{-1} \bmod \nu_{i-1}(t^{-1} z) \right) \right),$$

which takes $O(\mathsf{M}_{\mathbb{K}}(m_1 \cdots m_{i-1}\, m_i^2))$ operations in $\mathbb{K}$, when using our assumption that $\frac{\nu_i(z)}{\nu_{i-1}(t^{-1} z)}$ and its inverse modulo $\nu_{i-1}(t^{-1} z)$ and $\lambda_i(t)$ have been precomputed. □

COROLLARY 7.16. *Let $(\mathbb{K}_i)_{i\leqslant t}$ be a composed composition tower for $h\in\mathbb{K}[x]$ with $\deg h=n$. Then, given $f,g\in\mathbb{K}[x]_{<n}$, we may compute $f\circ g \operatorname{rem} h$ using*

$$O(\mathsf{M}_{\mathbb{K}}(\bar{m}\,n)\log n)$$

operations in $\mathbb{K}$.

Proof. Lemma 7.15 implies

$$\sum_{i=1}^{t} n_i\,\mathsf{L}_{\mathbb{K}_i/\mathbb{K}} = O\left(\sum_{i=1}^{t} n_i\,\mathsf{M}_{\mathbb{K}}(m_1\cdots m_{i-1}m_i^2)\right) = O(t\,\mathsf{M}_{\mathbb{K}}(\bar{m}\,n)).$$

We conclude by Lemma 7.1. □

Given a number $c>0$, we say that an integer $n>1$ is *c-super-smooth* if for every prime power m that divides n, we have $m\leqslant 2\log^c n$. For instance, the product of the first τ prime numbers grows as $e^{(1+o(1))\tau\log\tau}$ (see for instance [16, chapter 22]) and is therefore 1-super-smooth for sufficiently large τ. Similarly, the number $\operatorname{lcm}(1,\ldots,k)$ is 1-super-smooth for every $k\geqslant 3$. For a fixed modulus of super-smooth degree, the following corollary shows that modular composition can be done in quasi-linear time in this specific situation:

COROLLARY 7.17. *Let $h\in\mathbb{K}[x]$ be a fixed irreducible polynomial of c-super-smooth degree n over a finite field $\mathbb{K}$, and assume a composed composition tower has been precomputed for h. Then, given any $f,g\in\mathbb{K}[x]_{<n}$, we may compute $f\circ g \operatorname{rem} h$ using $O(n(\log n)^{2+c+o(1)})$ operations in $\mathbb{K}$.*

Proof. We apply the previous corollary with $\bar{m}\leqslant 2\log^c n$ and $\mathsf{M}_{\mathbb{K}}(n)=O(n\log n\log\log n)$. □

7.5. Artin–Schreier towers

Using the composed tower approach, we are left with the question how to deal with algebraic extensions of prime power degree $n=r^k$. In the case when r coincides with the characteristic p of the field $\mathbb{K}=\mathbb{F}_q=\mathbb{F}_{p^d}$, one may use Artin–Schreier towers instead, as outlined below.

An *Artin–Schreier* polynomial over a field $\mathbb{K}$ of characteristic $p>0$, is an irreducible polynomial of $\mathbb{K}[x]$ of the form x^p-x-a. An *Artin–Schreier tower* of height t over $\mathbb{F}_q$ is a tower of field extensions $\mathbb{F}_q\subset\mathbb{F}_{q^p}\subset\cdots\subset\mathbb{F}_{q^{p^t}}$ where each extension $\mathbb{F}_{q^{p^{i+1}}}$ is explicitly constructed from an Artin–Schreier polynomial over $\mathbb{F}_{q^{p^i}}$.

In terms of our tower $\mathbb{F}_q=\mathbb{F}_p[\alpha_0]=\mathbb{K}\subset\mathbb{K}_1\subset\mathbb{K}_2\subset\cdots\subset\mathbb{K}_t$, one supposes that the trace of α_0 over $\mathbb{F}_p$ is non zero and one takes

$$\begin{aligned}
\mu_1(z) &= z^p-z-\alpha_0 && \\
\mu_i(z) &= z^p-z-\alpha_1 && (i=2, p=2, d \text{ odd})\\
\mu_i(z) &= z^p-z-\alpha_{i-1}^{2p-1} && (\text{all other cases}).
\end{aligned}$$

In [9, Theorem 2], it is shown that this defines a primitive tower. The polynomials ν_i may even be computed fast according to [9, Theorems 12]. For such towers we count the number of operations in $\mathbb{F}_p$ instead of in $\mathbb{F}_q$, and the cost of the upward and downward conversion are as follows.

LEMMA 7.18. [9, Theorem 13] *Let $(\mathbb{K}_i)_{i\leqslant t}$ be an Artin–Schreier tower. Then, modulo precomputations, we have*

$$\mathsf{L}_{\mathbb{K}_i/\mathbb{F}_p} = O(p^{i+1}\,d\log_p^2(p^i d)+p\,\mathsf{M}_{\mathbb{F}_p}(p^i d))$$

for all $i \in \{1, \ldots, t\}$.

COROLLARY 7.19. *Let* $(\mathbb{K}_i)_{i \leqslant t}$ *be an Artin–Schreier composition tower for* $h \in \mathbb{K}[x]$ *with* $\deg h = n = p^t$. *Then, given* $f, g \in \mathbb{K}[x]_{<n}$, *we may compute* $f \circ g \operatorname{rem} h$ *using*

$$O(p\,d\,n \log_p^2(d\,n) \log_p n + \mathsf{M}_{\mathbb{F}_p}(p\,d\,n) \log n)$$

operations in $\mathbb{F}_p$.

Proof. Lemma 7.18 implies

$$\sum_{i=1}^{t} n_i \mathsf{L}_{\mathbb{K}_i/\mathbb{F}_p} = O\left(\sum_{i=1}^{t} p^{t-i} (p^{i+1} d \log_p^2(p^i d) + p\,\mathsf{M}_{\mathbb{F}_p}(p^i d)) \right) = O(t\,(p^{t+1} d \log_p^2(p^t d) + p\,\mathsf{M}_{\mathbb{F}_p}(p^t d))).$$

The result therefore follows from Lemma 7.1. □

8. BUILDING COMPOSITION TOWERS

Let $\mathbb{K} = \mathbb{F}_q$ be a given finite field and let h be a given monic irreducible polynomial of degree $n = m_1 \cdots m_t$ over $\mathbb{K}$. In this section we consider the task of constructing composition towers for h. In fact, we may distinguish three different problems of increasing complexity:

1. Building an algebraic tower $(\mathbb{K}_i)_{i \leqslant t}$ with the prescribed extension degrees $m_i = [\mathbb{K}_i : \mathbb{K}_{i-1}]$;
2. Building a composition tower for *some* monic irreducible $h \in \mathbb{K}[x]$ of degree n;
3. Building a composition tower for the prescribed modulus h.

Moreover, one may study these problems for each type of towers that we have encountered so far. From now on, we drop the study of triangular towers, for simplicity and because primitive towers are more efficient anyway. Given an effective tower, we notice that the construction of a primitive tower still requires computing the minimal polynomials ν_i over $\mathbb{K}$ of the α_i. In general, it is not known how to do this in softly linear time without using the modular composition algorithm by Kedlaya and Umans.

The traditional solution to the first problem involves computing irreducible polynomials μ_i of "small" degrees m_i over "large" field extensions $\mathbb{K}_{i-1}$. The number of operations in $\mathbb{K}_{i-1}$ then grows with $m_i \log q^{m_1 \cdots m_{i-1}}$, even with the fastest known algorithm of Shoup [42]. In total this leads to at least a quadratic cost in n to build an effective tower. In [13, Proposition 4.6] von zur Gathen and Seroussi proved the lower bound $\Omega(\log q)$ for factoring polynomials of degree 2 over $\mathbb{F}_q$ in the arithmetic circuit model. Consequently the quadratic cost in n might be difficult to decrease in general. Theorem 8.2 below shows how to achieve a cost that is quasi-linear in $n\,\bar{m}\,2^t$ in the case when $n = m_1 \cdots m_t$.

In this section we mainly focus on the efficient construction of primitive composition towers for prescribed m_1, …, m_t. We first describe the general algorithm and then explain possible speed-ups for nested, composed, and Artin-Schreier towers. Altogether, this yields an efficient answer to the second problem. Notice that composition towers of this type are sufficient if we merely need a representation for the finite field $\mathbb{F}_{q^n}$ such that polynomials in $\mathbb{F}_q[x]$ can be evaluated efficiently at points in $\mathbb{F}_{q^n}$.

The third problem is more difficult. So far we have not been able to apply the techniques of this paper to obtain more efficient solutions to this problem, even when n is very smooth or in the extremely favourable case of Artin–Schreier towers. In practice, a straightforward strategy for the construction of composition towers for h is to factor h over all intermediate fields $\mathbb{K}_1, \ldots, \mathbb{K}_t$ using standard available algorithms. This is discussed at the end of the section.

8.1. Building primitive composition towers

The naive construction of a primitive composition tower with prescribed extension degrees m_i proceeds by induction. Assume the tower is built up to height $t-1$. We first construct an irreducible polynomial $\mu_t(z)$ in $\mathbb{K}_{t-1}[z]$ and set $\mathbb{K}_t = \mathbb{K}_{t-1}[z]/(\mu_t(z))$. If μ_t is "sufficiently random", then the class α_t of z in $\mathbb{K}_t$ is a primitive element over $\mathbb{K}$. Its minimal polynomial ν_t over $\mathbb{K}$ may simply be obtained as $\nu_t(z) = \mathrm{Res}_x(\breve{\mu}_t(x, z), \nu_{t-1}(x))$ where $\breve{\mu}_t(x, z) \in \mathbb{K}[x, z]$ is the natural preimage with bidegree $(<m_1 \cdots m_{t-1}, m_t)$ that satisfies $\breve{\mu}_t(\alpha_{t-1}, z) = \mu_t(z)$. The latter resultant requires roughly $(m_1 \cdots m_{t-1})^2$ operations in $\mathbb{K}$ when using the best known algorithms. To complete the decomposition tower of height t, it remains to compute the sequence of normal factors of $\nu_t(x)$ over $\mathbb{K}_1, \ldots, \mathbb{K}_{t-1}$. Factorization algorithms based on modular composition [29] can achieve this in time $(m_1 \cdots m_{t-1})^{1.5}$, roughly speaking.

In fact, we will show how to build primitive composition towers far more efficiently. We still proceed by induction. The composition towers naturally share the same underlying effective subtowers of $(\mathbb{K}_i)_{i \leqslant t}$. More precisely, the subtower of height l consists of the fields $(\mathbb{K}_i)_{i \leqslant l}$ and forms a composition tower for ν_l; the successive normal factors are written $H_{l,0} = \nu_l$, $H_{l,1}$, …, $H_{l,l} = x - \alpha_l$, and the auxiliary polynomials a_i of Definition 5.2 are written $a_{l,1}, \ldots, a_{l,l}$.

We begin with constructing an irreducible polynomial $\mu_1(z) \in \mathbb{K}[z]$ of degree m_1, so the first primitive tower is made of $\mu_1, \nu_1 = \mu_1$, and it is a composition tower for $H_{1,0}(x) = \nu_1(x)$ with normal factor $H_{1,1}(x) = x - \alpha_1$. The auxiliary polynomial $a_{1,1}(x) = x$ satisfies $\mu_1(a_{1,1}(x)) = 0 \bmod H_{1,0}(x)$.

For the second composition tower we construct an irreducible polynomial $\mu_2(z) \in \mathbb{K}_1[z]$ of degree m_2, which defines α_2 as the class of z in $\mathbb{K}_2 = \mathbb{K}_1[z]/(\mu_2(z))$. If μ_2 is "sufficiently random", then α_2 is a primitive element of $\mathbb{K}_2$ over $\mathbb{K}$. We obtain the minimal polynomial of α_2 over $\mathbb{K}$ as $\nu_2(z) = \mathrm{Res}_x(\breve{\mu}_2(x, z), \nu_1(x))$, where $\breve{\mu}_2(\alpha_1, z) = \mu_2(z)$ and $\breve{\mu}_2 \in \mathbb{K}[x]_{<m_1}[z]$. The normal factor of $H_{2,0}(x) = \nu_2(x)$ over $\mathbb{K}_1$ is $H_{2,1}(x) = \mu_2(x)$, and the one of $H_{2,1}$ over $\mathbb{K}_2$ is $H_{2,2}(x) = x - \alpha_2$. We clearly have $a_{2,2}(x) = x$ and we obtain $a_{2,1}(x)$ from the subresultant of degree 1 in x of $\breve{\mu}_2(x, z)$ and $\nu_1(x)$ (it necessarily exists because α_2 is a primitive element of $\mathbb{K}_2$ over $\mathbb{K}$ implies that ν_2 is separable; this will be detailed below in the general situation). In this way we obtain a decomposition tower of height 2.

The third tower again requires building an irreducible polynomial $\mu_3(z) \in \mathbb{K}_2[z]$. This defines α_3 and $\mathbb{K}_3$ so we have $\mu_3(\alpha_3) = 0$. We assume that α_3 generates $\mathbb{K}_3$ over $\mathbb{K}$ and we wish to obtain the normal factorizations of its minimal polynomial $\nu_3(z) = \mathrm{Res}_x(\breve{\mu}_3(x, z), \nu_2(x))$. Over $\mathbb{K}_3$ and $\mathbb{K}_2$ the normal factors are respectively $H_{3,3}(x) = x - \alpha_3$ and $H_{3,2}(x) = \mu_3(x)$. For $H_{3,1}(x)$, we use the second composition tower: we compute $\breve{\mu}_3(z, x) = \Lambda_2^{-1}(\mu_3(x))$ such that $\breve{\mu}_3(z, x) \in \mathbb{K}_1[z]_{<m_2}[x]$ and $\breve{\mu}_3(\alpha_2, x) = \mu_3(x)$, then $H_{3,1}(x) = \mathrm{Res}_z(\breve{\mu}_3(z, x), \mu_2(z)) \in \mathbb{K}_1[x]$. Then we obtain $H_{3,0}(x) = \mathrm{Res}_z\big(\breve{H}_{3,1}(z, x), \mu_1(z)\big)$, where $\breve{H}_{3,1}(z, x) = \Lambda_1^{-1}(H_{3,1}(x))$. The auxiliary polynomials $a_{3,1}$ and $a_{3,2}$ are obtained from the corresponding first subresultants.

For general heights, we use the following algorithm for the construction of composition towers.

Algorithm 8.1

Input. Primitive composition towers $(\mathbb{K}_j)_{j \leqslant i}$ for ν_i for $i \leqslant t-1$; $\mu_t(z) \in \mathbb{K}_{t-1}[z]$ irreducible of degree m_t such that any root of μ_t has degree $m_1 \cdots m_t$ over $\mathbb{K}$.

Output. The primitive composition tower $(\mathbb{K}_i)_{i \leqslant t}$ for ν_t with $\mathbb{K}_{t-1}[\alpha_t] = \mathbb{K}_{t-1}[z]/(\mu_t(z))$, and where ν_t is the minimal polynomial of α_t over $\mathbb{K}$.

Notation. The normal factors of $(\mathbb{K}_i)_{i \leqslant l}$ are written $H_{l,0}, \ldots, H_{l,l}$.

1. Set $H_{t,t}(x) = x - \alpha_t$, $H_{t,t-1}(x) = \mu_t(x)$, $a_{t,t}(x) = x$.
2. For l from $t-1$ down to 1 do
   a. Compute $H_{t,l-1}(x) = \mathrm{Res}_z\big(\breve{H}_{t,l}(z, x), \mu_l(z)\big)$ over $\mathbb{K}_{l-1}$, where $\breve{H}_{t,l}(z, x) = \Lambda_l^{-1}(H_{t,l}(x))$;
   b. Compute the subresultant of degree 1 in z of $\breve{H}_{t,l}(z, x)$ and $\mu_l(z)$ written $A(x)\, z + B(x)$, and then set $a_{t,l}(x) = -A(x)^{-1} B(x) \bmod H_{t,l-1}(x)$.

3. Set α_t to the class of z in $\mathbb{K}_{t-1}[z]/(\mu_t(z))$, and let $\nu_t(z) = H_{t,0}(z)$.
   Notice that precomputations PRE-P1 correspond to polynomials $H_{t,1}, \ldots, H_{t,t}$; For PRE-P2 we have already computed $\Lambda_1^{-1}(H_{t,1}), \ldots, \Lambda_{t-1}^{-1}(H_{t,t-1})$, and $\Lambda_t^{-1}(H_{t,t})$ is simply $x - \alpha_t$; PRE-P3 correspond to $a_{t,1}, \ldots, a_{t,t}$.
4. Return the composition tower $(\mathbb{K}_i)_{i \leqslant t}$ for $H_{t,0}(x)$ made from $(\mu_i)_{i \leqslant t}$, $(\nu_i)_{i \leqslant t}$, $(H_{t,i})_{\leqslant t}$, $(a_{t,i})_{i \leqslant t}$, and the other precomputed auxiliary data.

PROPOSITION 8.1. *Algorithm 8.1 is correct and takes*

$$O\left(\sum_{l=1}^{t-1} n_l \mathsf{L}_{\mathbb{K}_l/\mathbb{K}} + \mathsf{M}_{\mathbb{K}}(\bar{m}\, n) \log n\right)$$

operations in $\mathbb{K}$.

Proof. By decreasing induction on l we prove that $\mathbb{K}_l, \ldots, \mathbb{K}_t$ is a composition tower for $H_{t,l}$ which has degree $m_{l+1} \cdots m_t$. This is clear for $l = t$ and $l = t-1$. Assume the induction hypothesis holds for l. At the end of step 2.a $H_{t,l-1}(x)$ has degree $m_l \cdots m_{t+1}$ and is the minimal polynomial of α_t over $\mathbb{K}_{l-1}$. Since $H_{t,l}$ is a normal factor of ν_t over $\mathbb{K}_l$, the degree of the ideal generated by $\big(\breve{H}_{t,l}(z,x), \mu_l(z)\big)$ is $m_l \cdots m_{t+1}$.

Since $H_{t,l-1}$ is separable it belongs to the Groebner basis of $\big(\breve{H}_{t,l}(z,x), \mu_l(z)\big)$ for the lexicographic order induced by $z > x$. In particular a polynomial with leading monomial z belongs to $\big(\breve{H}_{t,l}(z,x), \mu_l(z)\big)$. This proves that the subresultant of degree 1 of $\breve{H}_{t,l}(z,x)$ and $\mu_l(z)$ is non zero, and that a_l is well defined, which implies the requested conditions:

$$H_{t,l}(a_{t,l}(x), x) = 0 \bmod H_{t,l-1}(x) \qquad \text{and} \qquad \mu_l(a_{t,l}(x)) = 0 \bmod H_{t,l-1}(x).$$

The induction hypothesis is thus satisfied for $l-1$.

As to the complexity analysis, we first notice that $\deg H_{t,l} = m_{l+1} \cdots m_t$. The conversions in step 2.a therefore take

$$O\left(\sum_{l=1}^{t-1} m_{l+1} \cdots m_t \mathsf{L}_{\mathbb{K}_l/\mathbb{K}}\right)$$

operations in $\mathbb{K}$. The resultant in step 2.a and the subresultant in step 2.b require

$$O(\mathsf{M}_{\mathbb{K}_{l-1}/\mathbb{K}}(m_l^2\, m_{l+1} \cdots m_t) \log m_l + m_l \mathsf{D}_{\mathbb{K}_{l-1}/\mathbb{K}})$$

further operations, by Proposition 2.2, where $\mathsf{D}_{\mathbb{K}_{l-1}/\mathbb{K}} = O(\mathsf{M}(m_1 \cdots m_{l-1}) \log(m_1 \cdots m_{l-1}))$. The inversion of A modulo $H_{t,l-1}$ takes $O(\mathsf{M}_{\mathbb{K}_{l-1}/\mathbb{K}}(m_l \cdots m_t) \log(m_l \cdots m_t))$ more operations. □

THEOREM 8.2. *Let* $\mathbb{K} = \mathbb{F}_q$. *A primitive composition tower of degrees* $m_1, \ldots, m_t$ *may be built using*

$$O(\bar{m}\, \mathsf{M}_{\mathbb{K}}(\bar{m}\, n)\, (2^t + \log n) \log n + \bar{m}\, \mathsf{M}_{\mathbb{K}}(n) \log q)$$

expected operations in $\mathbb{K}$.

Proof. The tower is constructed by natural successive applications of the Algorithm 8.1. By Corollary 7.4 we have

$$\sum_{l=1}^{t-1} n_l \mathsf{L}_{\mathbb{K}_l/\mathbb{K}} = O\left(\bar{m} \sum_{l=1}^{t-1} 2^l n_l \mathsf{M}_{\mathbb{K}}(m_1 \cdots m_l)\right) = O(2^t \bar{m}\, \mathsf{M}_{\mathbb{K}}(n)),$$

so the cost of the latter proposition simplifies to

$$O(2^t \bar{m}\, \mathsf{M}_{\mathbb{K}}(n) + \mathsf{M}_{\mathbb{K}}(\bar{m}\, n) \log n).$$

It remains to take the construction of the μ_i into account for $1 \leqslant i \leqslant t$. By Theorem A.4 a random polynomial μ_i can be computed using an expected number of $O(m_i^2 \log(m_1 \cdots m_i))$ compositions modulo ν_{i-1} plus

$$O(m_i \mathsf{C}_{\mathbb{K}_{i-1}/\mathbb{K}}(m_i) \log(m_1 \cdots m_i) + m_i \mathsf{M}_{\mathbb{K}}(m_1 \cdots m_i) \log q)$$

operations in $\mathbb{K}$. Thanks to Corollary 7.5 each composition with v_{i-1} may be done using $\mathsf{M}(\bar{m}\, m_1 \cdots m_{i-1})\,(2^{i-1} + \log(m_1 \cdots m_{i-1}))$ operations in $\mathbb{K}$. On the other hand we simply take $\mathsf{C}_{\mathbb{K}_{i-1}/\mathbb{K}}(m_i) = O(\mathsf{M}(m_1 \cdots m_{i-1} m_i^2))$. The sum over i yields

$$O(\bar{m}\,\mathsf{M}_{\mathbb{K}}(\bar{m}\,n)\,(2^t + \log n)\log n + \bar{m}\,\mathsf{M}_{\mathbb{K}}(n)\log q).$$

With a small uniformly bounded probability, the roots of μ_i do not have maximal order $m_1 \cdots m_i$ (see [12, chapter 14]). If we run into such a bad μ_i then it is easily detected by the algorithm since one v_i is not separable. In such a case we build an other μ_i at random and the average number of failure is bounded. □

8.2. Particular cases

Let us now investigate the consequences of Proposition 8.1 in the particular cases of nested, composed and Artin–Schreier towers. We assume that the minimal polynomials μ_i are already known. Of course, there are cases in which the computation of these minimal polynomials is actually the main bottleneck. In our analysis, we actually construct all intermediate subtowers along with the composition tower itself. If one just needs the highest tower, then there are cases that can be optimized by exploiting the specificities of the types of towers under consideration.

Nested towers.

COROLLARY 8.3. *Let $\mathbb{K} = \mathbb{F}_q$ and assume given $(\tau_i)_{1\leqslant i\leqslant t}$ and $(\kappa_i)_{1\leqslant i\leqslant t}$ defining a nested tower as defined in section 7.3. Then the nested composition tower for v_t may be built using*

$$O(t\,\mathsf{M}_{\mathbb{K}}(n)\log n + \mathsf{M}_{\mathbb{K}}(\bar{m}\,n)\log n)$$

operations in $\mathbb{K}$.

Proof. The tower is again constructed by natural successive applications of the Algorithm 8.1, except that we replace the precomputations in step 3 by those specified in PRE-N1. These precomputations amount to $O(\mathsf{M}_{\mathbb{K}}(n)\log n)$.

By Corollary 7.6 we have

$$\sum_{l=1}^{t-1} n_i\,\mathsf{L}_{\mathbb{K}_i/\mathbb{K}} = O\left(\sum_{l=1}^{t-1} n_l\,\mathsf{M}_{\mathbb{K}}(m_1 \cdots m_l)\log(m_1 \cdots m_l)\right) = O(t\,\mathsf{M}_{\mathbb{K}}(n)\log n),$$

so the cost of Proposition 8.1 simplifies to $O(t\,\mathsf{M}_{\mathbb{K}}(n)\log n + \mathsf{M}_{\mathbb{K}}(\bar{m}\,n)\log n)$. □

Composed towers.

COROLLARY 8.4. *Let $\mathbb{K} = \mathbb{F}_q$ and assume given $(\lambda_i)_{1\leqslant i\leqslant t}$ defining a nested tower as defined in section 7.4. Then the composed composition tower for v_t may be built using*

$$O(\mathsf{M}_{\mathbb{K}}(\bar{m}\,n)\log n)$$

operations in $\mathbb{K}$.

Proof. The tower is again constructed by natural successive applications of the Algorithm 8.1, except that we replace the precomputations in step 3 by those specified in PRE-N1. These precomputations amount to $O(\mathsf{M}_{\mathbb{K}}(n)\log n)$.

By Corollary 7.6 we have

$$\sum_{l=1}^{t-1} n_l\,\mathsf{L}_{\mathbb{K}_l/\mathbb{K}} = O\left(\sum_{l=1}^{t-1} n_l\,\mathsf{M}_{\mathbb{K}}(m_1 \cdots m_{l-1} m_l^2)\right) = O(t\,\mathsf{M}_{\mathbb{K}}(\bar{m}\,n)),$$

so the cost of Proposition 8.1 simplifies to $O(\mathsf{M}_{\mathbb{K}}(\bar{m}n)\log n)$. □

Artin–Schreier towers.

COROLLARY 8.5. *Let $\mathbb{K}=\mathbb{F}_q$ and assume given $(\mu_i)_{1\leqslant i\leqslant t}$ defining an Artin–Schreier tower as in section 7.5. Then the Artin–Schreier composition tower for ν_t may be built using*

$$O(t\,p\,n\,d\log^2(n\,d)+\mathsf{M}_{\mathbb{F}_p}(p\,n\,d)\log n)$$

operations in $\mathbb{F}_p$.

Proof. The tower is again constructed by natural successive applications of the Algorithm 8.1, expect that we may discard the precomputations in step 3. By Lemma 7.18 we have

$$\sum_{l=1}^{t-1} n_l\,\mathsf{L}_{\mathbb{K}_l/\mathbb{F}_p}=O(t\,(p\,n\,d\log_p^2(n\,d)+p\,\mathsf{M}_{\mathbb{F}_p}(n\,d))),$$

so the cost of Proposition 8.1 becomes

$$O(t\,p\,n\,d\log^2(n\,d)+\mathsf{M}_{\mathbb{F}_p}(p\,n\,d)\log n).$$ □

8.3. Building composition towers from roots

Let $(\mathbb{K}_i)_{i\leqslant t}$ be an algebraic tower. So far we have shown how to built composition towers for ν_t. Let us now explain how to deduce composition towers for a given $h\in\mathbb{K}[x]$ monic and irreducible of degree n. By standard algorithms, we first compute a root ζ of h in $\mathbb{K}_t$ (for example with Rabin's algorithm; see [12, chapter 14]). Unfortunately we do not know how to exploit a composition tower to compute ζ more efficiently. We are interested in finding a monic normal factor H_i of h over each field $\mathbb{K}_i$ in the tower. In the extreme case when $i=t$, we take $H_t=x-\zeta$.

PROPOSITION 8.6. *Given a primitive composition tower for ν_t, given an irreducible $h\in\mathbb{K}[x]$, together with a root $\zeta\in\mathbb{K}_t$ of h, we may compute the minimal polynomials H_i of ζ over $\mathbb{K}_i$ along with the a_i from Definition 5.2 for $i\in\{0,\ldots,t\}$, with cost*

$$O\left(\sum_{i=1}^{t} n_i\,\mathsf{L}_{\mathbb{K}_i/\mathbb{K}}+\mathsf{M}_{\mathbb{K}}(\bar{m}n)\log n\right).$$

Proof. We have $H_t=x-\zeta$. We compute H_i and a_i by descending induction on i from t down to 1. First we have

$$H_{i-1}(x) \;=\; \mathrm{Res}_z\big(\breve{H}_i(z,x),\mu_i(z)\big) \;\in\; \mathbb{K}_{i-1}[x],$$

where $\breve{H}_i=\Lambda_i^{-1}(H_i)\in\mathbb{K}_{i-1}[z]_{<m_i}[x]$. The first subresultant $A(x)\,z+B(x)$ of $\breve{H}_i(z,x)$ and $\mu_i(z)$ is well defined and $A(x)$ is invertible modulo $H_{i-1}(x)$. Therefore we have

$$a_i(x)=-A(x)^{-1}\,B(x)\bmod H_{i-1}(x).$$

According to Proposition 2.2, the resultant and the first subresultant can be computed using $O(\mathsf{M}_{\mathbb{K}_{i-1}/\mathbb{K}}(n_{i-1}\,m_i)\log m_i)$ operations in $\mathbb{K}$. Then a_i is deduced with $O(\mathsf{M}_{\mathbb{K}}(n_{i-1})\log n_{i-1})$ operations in $\mathbb{K}$. Summing over i, the result follows. □

Remark 8.7. Inversely, given a composition tower $(\mathbb{K}_i)_{i\leqslant t}$, we notice that the polynomial H_t is of the form $x-\zeta$ with $h(\zeta)=0$. In other words, the computation of a composition tower for h is as least as hard as finding a root of h in $\mathbb{K}_t$.

9. CONCLUSION

We have shown that modular composition and characteristic polynomials for a fixed irreducible modulus $h \in \mathbb{F}_q[x]$ can be computed fast if $n = \deg h$ is smooth. From a practical point of view, our algorithms lead to speed-ups with respect to state of the art methods as soon as n is composite and not reasonably large. From a theoretical point of view, the algorithms by Kedlaya and Umans generally outperform the new algorithms. One notable exception occurs when n is very smooth, in which case we were able to prove a quasi-linear complexity bound: see Corollary 7.17.

Our new algorithms are only more efficient for fixed moduli h. Nevertheless, the cost of the required precomputations as a function of h is of a similar order of magnitude as one composition modulo h without our methods. In other words, if we need to compute s compositions modulo the same modulus h, then our new methods are already of interest for small values of s. This problem of multiple modular compositions occurs in various applications:

Factorization over finite fields. A standard application of modular composition over finite fields is the irreducible factorization of univariate polynomials [27, 29]. In general, the cost of the precomputations is too expensive for our algorithms to be interesting. Nevertheless, our new algorithms are the most efficient ones in the case when we need to factor a polynomial of small degree over a finite field $\mathbb{F}_{p^n}$ such that p is prime and n is sufficiently smooth: see the appendix below.

Roots over large finite fields. One particular instance of factorization over finite fields is the extraction of t-th roots (i.e. finding the roots of polynomials of the form $x^t - a$). In that case, one may use [11, Theorem 1.1] in order to reduce the problem of root extraction to modular composition and the computation of minimal polynomials. The advantage of this method with respect to a direct use of [26] is that it typically remains fast for larger values of t.

Conversions. It frequently happens that one has to work with different representations of elements of a finite field. For instance, given distinct irreducible polynomials φ and ψ of degree n over $\mathbb{F}_q$, elements of $\mathbb{F}_{q^n}$ can both be represented as elements of $\mathbb{F}_q[x]/(\varphi)$ and $\mathbb{F}_q[x]/(\psi)$. Converting between these two different representations boils down to composition modulo φ or ψ. Given a non trivial divisor d of n, yet another representation of elements of $\mathbb{F}_{q^n}$ was needed in [21]. Let α be a primitive element of $\mathbb{F}_{q^n}$ and $\beta \in \mathbb{F}_{q^n}$ an element whose minimal polynomial has degree d. Then $\mathbb{F}_q[\alpha]$ and $\mathbb{F}_q[\beta][\alpha]$ are both isomorphic to $\mathbb{F}_{q^n}$ and correspond to two different representations. If α and β can be chosen "nicely", then conversions between these representations can be computed fast, using similar methods as in sections 7.3, 7.4 and 7.5. The general case reduces to this special case when we chose α and β using modular composition.

Frobenius maps. Thanks to von zur Gathen and Shoup's algorithm [14], the Frobenius maps $a \mapsto a^{q^i}$ can be computed efficiently when fast modular composition is available. More precisely, assume we wish to compute a^{q^i} in $\mathbb{F}_{q^n} = \mathbb{F}_q[x]/(h(x))$. If $i = 1$, then we use binary powering to get the preimage $b(x)$ of a^q. Then, by induction on i, we may compute the canonical preimage $c(x)$ of $a^{q^{\lfloor i/2 \rfloor}}$, so $d = c \circ c \operatorname{rem} h$ is the preimage of $a^{q^{2\lfloor i/2 \rfloor}}$. If i is even, then we are done. Otherwise, we compute $d \circ c \operatorname{rem} h$ to obtain the preimage of $a^{q^{2\lfloor i/2 \rfloor + 1}} = a^{q^i}$. Overall this method performs $O(\log n)$ compositions modulo h, plus $O(\mathsf{M}_{\mathbb{K}}(n) \log q)$ operations in $\mathbb{K}$.

Many intriguing questions remain to be answered. One major open problem is whether our new techniques can be used to build composition towers for a prescribed irreducible modulus h of smooth degree n in quasi-linear time. This would give us an unconditional algorithm for composition modulo h of quasi-linear time complexity. Another natural question is whether there exists an efficient way to reduce general modular composition to composition modulo irreducible polynomials of smooth degree. This question may be related to generalizations of our algorithms to modular composition for multivariate polynomials. Besides these fundamental issues, we finally think that there remains a lot of room for more minor improvements of the techniques in this paper and for working out various applications in more detail.

APPENDIX A. FACTORIZATION OVER FINITE FIELDS

This appendix is devoted to a fast algorithm for pseudo-trace computations designed by Kaltofen and Shoup in [26], together with its application to polynomial factorization over finite fields. We recall the algorithm for completeness and in order to make it work more generally over any finite field $\mathbb{F}_q$.

Algorithm A.1

Input. $h \in \mathbb{F}_q[z]$ irreducible of degree m; $f(z,x) \in \mathbb{F}_q[z,x]$ of bidegree $(<m, n)$ monic in x; $A(z,x) \in \mathbb{F}_q(z,x)$ of bidegree $(<m, <n)$; an integer $l \geqslant 1$; $Z_1 \in \mathbb{F}_q[z]_{<m}$ such that $Z_1(z) = z^q \bmod h(z)$; $X_1 \in \mathbb{F}_q(z,x)$ of bidegree $(<m, <n)$ such that $X_1(z,x) = x^q \bmod (h(z), f(z,x))$; $A_1 \in \mathbb{F}_q(z,x)$ of bidegree $(<m, <n)$ such that $A_1(z,x) = A(z,x)^q \bmod (h(z), f(z,x))$.

Output. $Z_l \in \mathbb{F}_q[z]_{<m}$ such that $Z_l(z) = z^{q^l} \bmod h(z)$; $X_l \in \mathbb{F}_q(z,x)$ of bidegree $(<m, <n)$ such that $X_l(z,x) = x^{q^l} \bmod (h(z), f(z,x))$; $A_l \in \mathbb{F}_q(z,x)$ of bidegree $(<m, <n)$ such that $A_l(z,x) = A(z,x)^q + A(z,x)^{q^2} + \cdots + A(z,x)^{q^l} \bmod (h(z), f(z,x))$.

1. If $l = 1$ then return Z_1, X_1, A_1.
2. Let $h = \lfloor l/2 \rfloor$, and recursively compute Z_h, X_h, A_h.
3. Compute $Z_{2h} = Z_h \circ Z_h \operatorname{rem} h$.
4. Compute $X_{2h} = X_h(Z_h, X_h) \operatorname{rem} f(z,x)$ over $\mathbb{F}_q[z]/(h(z))$.
5. Compute $A_{2h} = A_h + A_h(Z_h, X_h) \operatorname{rem} f(z,x)$ over $\mathbb{F}_q[z]/(h(z))$.
6. If l is even the return Z_{2h}, X_{2h}, A_{2h}.
7. Compute and return $Z_l = Z_{2h} \circ Z_1 \operatorname{rem} h$, $X_l = X_{2h}(Z_1, X_1) \operatorname{rem} f(z,x)$ over $\mathbb{F}_q[z]/(h(z))$, and $A_l = A_1 + A_{2h}(Z_1, X_1) \operatorname{rem} f(z,x)$ over $\mathbb{F}_q[z]/(h(z))$.

PROPOSITION A.1. *Let $\mathbb{K} = \mathbb{F}_q$ and $\mathbb{L} = \mathbb{F}_q[z]/(h(z))$. Algorithm A.1 is correct and takes $O(n \log l)$ compositions modulo h and*

$$O(\mathsf{C}_{\mathbb{L}/\mathbb{K}}(n) \log l)$$

additional operations in $\mathbb{K}$.

Proof. The proof proceeds by induction on l. The algorithm is clearly correct if $l = 1$. Assume that it is correct for $l \geqslant 1$. By linearity of the q-th power we have $Z_{2h} = z^{q^{2h}} \bmod h(z)$,

$$X_h(Z_h, X_h) \equiv x^{q^{2h}} \bmod (h(z), f(z,x)),$$

$$A_h(Z_h, X_h) \equiv A_h\left(z^{q^h}, x^{q^h}\right) \equiv A(z,x)^{q^{h+1}} + A(z,x)^{q^{h+2}} + \cdots + A(z,x)^{q^{2h}} \bmod (h(z), f(z,x)).$$

This already proves the correctness by induction when l is even. Otherwise $l = 2h+1$ and similar computations yield $Z_{2h} \circ Z_1 = (z^q)^{q^{2h}} = Z_l \bmod h(z)$,

$$X_{2h}(Z_1, X_1) \equiv x^{q^{2h+1}} \bmod (h(z), f(z,x)),$$

$$A_{2h}(Z_1, X_1) \equiv A(z,x)^{q^2} + A(z,x)^{q^3} + \cdots + A(z,x)^{q^{2h+1}} \bmod (h(z), f(z,x)).$$

This proves the correctness.

The cost for obtaining Z_l from Z_h is essentially one or two compositions modulo h. If we write $A_h(z,x) = \sum_{i=0}^{n-1} a_i(z)\, x^i$ with $a_i \in \mathbb{K}[z]_{<m}$, then we are led to compute $\tilde{a}_i = a_i \circ Z_h \operatorname{rem} h$ for all $0 \leqslant i \leqslant n-1$ and then $\sum_{i=0}^{n-1} \tilde{a}_i(z)\, X_h^i \bmod f(z,x)$ over $\mathbb{L}$. Overall A_{2h} requires $O(n)$ compositions modulo h plus $O(\mathsf{C}_{\mathbb{L}/\mathbb{K}}(n))$ operations in $\mathbb{K}$. The same bound holds for X_{2h}. If l is odd, then step 7 takes again $O(n)$ compositions modulo h plus $O(\mathsf{C}_{\mathbb{L}/\mathbb{K}}(n))$ operations in $\mathbb{K}$. Finally, the depth of the recursion is $O(\log l)$. □

COROLLARY A.2. *Let $\mathbb{K}=\mathbb{F}_q$, let h be a monic irreducible polynomial in $\mathbb{K}[z]$ of degree m, and let $\mathbb{L}=\mathbb{K}[z]/(h(z))$. For any monic polynomial $f\in\mathbb{L}[x]$ of degree n, and any $a\in\mathbb{L}[x]_{<n}$, the trace $\sum_{i=0}^{l} a^{q^i} \operatorname{rem} f$ may be computed using $O(n\log l)$ compositions modulo h plus*

$$O(\mathsf{C}_{\mathbb{L}/\mathbb{K}}(n)\log l+\mathsf{M}_{\mathbb{K}}(mn)\log q)$$

operations in $\mathbb{K}$.

Proof. If suffices to compute Z_1, X_1, A_1 for Algorithm A.1 using $O((\mathsf{M}_{\mathbb{L}/\mathbb{K}}(n)+\mathsf{M}_{\mathbb{K}}(m))\log q)$ operations in $\mathbb{K}$, in order to apply the preceding proposition. □

THEOREM A.3. *Let $\mathbb{K}=\mathbb{F}_q$, let h be a monic irreducible polynomial in $\mathbb{K}[z]$ of degree m, and let $\mathbb{L}=\mathbb{K}[z]/(h(z))$. Given n and the set of its prime factors of cardinality $\varpi(n)=O(\log n)$, we may check whether a monic polynomial $f\in\mathbb{L}[x]$ of degree n is irreducible using $O(n\,\varpi(n)\log(mn))$ compositions modulo h plus*

$$O(\varpi(n)\,\mathsf{C}_{\mathbb{L}/\mathbb{K}}(n)\log(mn)+\mathsf{M}_{\mathbb{K}}(mn)\log q)$$

operations in $\mathbb{K}$.

Proof. The polynomial f is irreducible if, and only if, $x^{q^{mn}}=x \bmod f(x)$ and $x^{q^{mn/\pi}}\neq x \bmod f(x)$ for all prime divisor π of n. We may thus apply the above corollary. □

THEOREM A.4. *Let $\mathbb{K}=\mathbb{F}_q$, let h be a monic irreducible polynomial in $\mathbb{K}[z]$ of degree m, and let $\mathbb{L}=\mathbb{K}[z]/(h(z))$. A random irreducible polynomial of degree n over $\mathbb{L}$ may be computed using an expected number of $O(n^2\log(mn))$ compositions modulo h plus*

$$O(n\,\mathsf{C}_{\mathbb{L}/\mathbb{K}}(n)\log(mn)+n\,\mathsf{M}_{\mathbb{K}}(mn)\log q)$$

operations in $\mathbb{K}$.

Proof. We appeal to Ben-Or's algorithm [12, Algorithm 14.40]. □

BIBLIOGRAPHY

[1] D. J. Bernstein. Composing power series over a finite ring in essentially linear time. *J. Symbolic Comput.*, 26(3):339–341, 1998.

[2] A. Bostan, Ph. Flajolet, B. Salvy, and É. Schost. Fast computation of special resultants. *J. Symbolic Comput.*, 41(1):1–29, 2006.

[3] A. Bostan, G. Lecerf, and É. Schost. Tellegen's principle into practice. In Hoon Hong, editor, *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*, ISSAC '03, pages 37–44, New York, NY, USA, 2003. ACM.

[4] J. V. Brawley and L. Carlitz. Irreducibles and the composed product for polynomials over a finite field. *Discrete Math.*, 65(2):115–139, 1987.

[5] R. P. Brent and H. T. Kung. Fast algorithms for manipulating formal power series. *J. ACM*, 25(4):581–595, 1978.

[6] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic complexity theory*, volume 315 of *Grundlehren der Mathematischen Wissenschaften*. Springer-Verlag, 1997.

[7] D. G. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Infor.*, 28(7):693–701, 1991.

[8] S. D. Cohen. On irreducible polynomials of certain types in finite fields. *Proc. Camb. Phil. Soc.*, 66(2):335–344, 1969.

[9] L. De Feo and É. Schost. Fast arithmetics in Artin-Schreier towers over finite fields. *J. Symbolic Comput.*, 47(7):771–792, 2012.

[10] L. E. Dickson. *Linear Groups: With an Exposition of the Galois Field Theory*. Dover Publication Inc., 1958.

[11] J. Doliskani and É. Schost. Taking roots over high extensions of finite fields. *Math. Comp.*, 83(285):435–446, 2014.

[12] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, New York, 2 edition, 2003.

[13] J. von zur Gathen and G. Seroussi. Boolean circuits versus arithmetic circuits. *Inform. and Comput.*, 91(1):142–154, 1991.

[14] J. von zur Gathen and V. Shoup. Computing Frobenius maps and factoring polynomials. *Comput. Complexity*, 2(3):187–224, 1992.

[15] B. Grenet, J. van der Hoeven, and G. Lecerf. Deterministic root finding over finite fields using Graeffe transforms. *Appl. Alg. Eng. Comm. Comp.*, 27(3):237–257, 2016.

[16] G. H. Hardy and E. M. Wright. *An introduction to the theory of numbers*. Oxford University Press, Oxford, 6 edition, 2008.

[17] D. Harvey, J. van der Hoeven, and G. Lecerf. Faster polynomial multiplication over finite fields. *J. ACM*, 63(6), 2017. Article 52.

[18] J. van der Hoeven. Relax, but don't be too lazy. *J. Symbolic Comput.*, 34(6):479–542, 2002.

[19] J. van der Hoeven. Fast composition of numeric power series. Technical Report 2008-09, Université Paris-Sud, Orsay, France, 2008.

[20] J. van der Hoeven. Faster Chinese remaindering. Technical report, CNRS & École polytechnique, 2016. http://hal.archives-ouvertes.fr/hal-01403810.

[21] J. van der Hoeven and R. Larrieu. The Frobenius FFT. Technical report, CNRS & École polytechnique, 2017. http://hal.archives-ouvertes.fr/hal-01453269.

[22] J. van der Hoeven and G. Lecerf. Composition modulo powers of polynomials. Technical report, HAL, 2017. http://hal.archives-ouvertes.fr/hal-01455722.

[23] J. van der Hoeven and G. Lecerf. Modular composition via complex roots. Technical report, HAL, 2017. http://hal.archives-ouvertes.fr/hal-01455731.

[24] Xiaohan Huang and V. Y. Pan. Fast rectangular matrix multiplication and applications. *J. Complexity*, 14(2):257–299, 1998.

[25] F. Johansson. A fast algorithm for reversion of power series. *Math. Comp.*, 84:475–484, 2015.

[26] E. Kaltofen and V. Shoup. Fast polynomial factorization over high algebraic extensions of finite fields. In *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation*, ISSAC '97, pages 184–188, New York, NY, USA, 1997. ACM.

[27] E. Kaltofen and V. Shoup. Subquadratic-time factoring of polynomials over finite fields. *Math. Comp.*, 67(223):1179–1197, 1998.

[28] K. S. Kedlaya and C. Umans. Fast modular composition in any characteristic. In *FOCS'08: IEEE Conference on Foundations of Computer Science*, pages 146–155, Washington, DC, USA, 2008. IEEE Computer Society.

[29] K. S. Kedlaya and C. Umans. Fast polynomial factorization and modular composition. *SIAM J. Comput.*, 40(6):1767–1802, 2011.

[30] M. K. Kyuregyan. Recurrent methods for constructing irreducible polynomials over F_q of odd characteristics. *Finite Fields Appl.*, 9(1):39–58, 2003.

[31] M. K. Kyuregyan. Recurrent methods for constructing irreducible polynomials over F_q of odd characteristics, II. *Finite Fields Appl.*, 12(3):357–378, 2006.

[32] P. Lairez and T. Vaccon. On p-adic differential equations with separation of variables. In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC '16, pages 319–323, New York, NY, USA, 2016. ACM.

[33] F. Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ISSAC '14, pages 296–303, New York, NY, USA, 2014. ACM.

[34] G. Lecerf. On the complexity of the Lickteig-Roy subresultant algorithm. Technical report, CNRS & École polyechnique, 2017.

[35] G. L. Mullen and D. Panario. *Handbook of Finite Fields*. Discrete Mathematics and Its Applications. Chapman and Hall/CRC, 2013.

[36] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[37] M. S. Paterson and L. J. Stockmeyer. On the number of nonscalar multiplications necessary to evaluate polynomials. *SIAM J.Comput.*, 2(1):60–66, 1973.

[38] M. O. Rabin. Probabilistic algorithms in finite fields. *SIAM J. Comput.*, 9(2):273–280, 1980.

[39] V. Ramaswami. On the number of positive integers less than x and free of prime divisors greater than x^c. *Bull. Amer. Math. Soc.*, 55:1122–1127, 1949.

[40] P. Ritzmann. A fast numerical algorithm for the composition of power series with complex coefficients. *Theoret. Comput. Sci.*, 44:1–16, 1986.

[41] J.-A. Serret. *Cours d'algèbre supérieure*, volume 2 of *Les grands classiques Gauthier-Villars*. Jacques Gabay, 4 edition, 1992. Réimpression du tome second de la 4^e édition du *Cours d'algèbre supérieure* de Joseph-Alfred Serret, publiée par Gauthier-Villars en 1879.

[42] V. Shoup. Fast construction of irreducible polynomials over finite fields. *J. Symbolic Comput.*, 17(5):371–391, 1994.

[43] V. Shoup. Efficient computation of minimal polynomials in algebraic extensions of finite fields. In *Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation*, ISSAC '99, pages 53–58, New York, NY, USA, 1999. ACM.