FAST SEPARABLE FACTORIZATION AND APPLICATIONS

GRÉGOIRE LECERF

Abstract. In this paper we show that the separable decomposition of a univariate polynomial can be computed in softly optimal time, in terms of the number of arithmetic operations in the coefficient field. We also adapt the classical multi-modular strategy that speeds up the computations for many coefficient fields, and we analyze consequences of the new results to the squarefree and the irreducible factorizations.

Introduction

Let $\mathbb{A}$ be a unique factorization domain, let $\mathbb{L}$ denote its field of fractions, and let p be the characteristic of $\mathbb{L}$. Let us recall that a polynomial $G \in \mathbb{A}[y] \setminus \mathbb{A}$ is said to be *separable* if $\operatorname{Res}(G, G') \neq 0$ ($\operatorname{Res}(F, G)$ represents the resultant of F and G). Equivalently this means that G has no multiple root in the algebraic closure $\bar{\mathbb{L}}$ of $\mathbb{L}$. If G is irreducible, then it is further equivalent to $G' \neq 0$. By convention, any constant polynomial is considered to be separable.

Throughout this paper $\mathcal{B} := \{1, p, p^2, p^3, \ldots\}$ denotes the set of the powers of p, and $m \bmod p$ represents the remainder of m in the division by p. If F is a primitive polynomial in $\mathbb{A}[y]$ of degree $d \geq 1$, and if $p > 0$, then the *separable decomposition* of F, written $\operatorname{Sep}(F)$, is defined to be the set

$$\operatorname{Sep}(F) := \{(G_1, q_1, m_1), \ldots, (G_s, q_s, m_s)\} \subseteq (\mathbb{A}[y] \setminus \mathbb{A}) \times \mathcal{B} \times \mathbb{N}$$

satisfying the following properties:

(S_1) $F(y) = \prod_{i=1}^{s} G_i(y^{q_i})^{m_i}$;
(S_2) for all $i \neq j$ in $\{1, \ldots, s\}$, $G_i(y^{q_i})$ and $G_j(y^{q_j})$ are coprime;
(S_3) for all $i \in \{1, \ldots, s\}$, $m_i \bmod p \neq 0$;
(S_4) for all $i \in \{1, \ldots, s\}$, G_i is separable, primitive and of positive degree;
(S_5) for all $i \neq j$ in $\{1, \ldots, s\}$, $(q_i, m_i) \neq (q_j, m_j)$.

We follow the same terminology as in [GG03]: the separable decomposition is the set $\operatorname{Sep}(F)$, while the *separable factorization* is the process of computing this set.

If $p = 0$ then the separable decomposition of F is naturally defined to be the set generated by all the triples $(G, 1, m)$ such that G is a proper squarefree factor of F with multiplicity $m \geq 1$. The existence and the uniqueness (the G_i being defined up to units in $\mathbb{A}$) of the separable decomposition are recalled in Proposition 4 of Section 1. Note that the separable decomposition of F coincides with its squarefree decomposition whenever $p = 0$ or $p \geq d + 1$.

Example 1. With $\mathbb{A} := \mathbb{F}_3$ and $F := y^2(y+1)^3(y+2)^4 = y^9 + 2y^8 + 2y^3 + y^2$, we have that $\operatorname{Sep}(F) = \{(y, 1, 2), (y + 1, 3, 1), (y + 2, 1, 4)\}$.

Date: Preliminary version of June 6, 2007. Revised version of October 9, 2007.

2000 *Mathematics Subject Classification.* Primary 12Y05, 68W30; Secondary 11Y16, 12D05, 13P05.

Key words and phrases. Polynomial factorization, separable factorization, squarefree factorization, irreducible factorization.

This work was partly supported by the French Research Agency via the Gecko project (gecko.inria.fr).

Example 2. With $\mathbb{A} := \mathbb{F}_3[x]$ and $F := (y+2x)^7(y^3+2x)^3(y^6+x)$, we have that $\mathrm{Sep}(F) = \{(y+2x,1,7),(y+2x^3,9,1),(y^2+x,3,1)\}$.

Overview. The goal of this article is to demonstrate that reducing the squarefree and the irreducible factorization problems to separable polynomials is a good theoretical point of view that is also very efficient in practice.

For completeness, the existence and the uniqueness of the separable decomposition are given in the first section. The second section is then devoted to the separable factorization algorithm. Therein we show that the separable decomposition can be computed by means of arithmetic operations ($+, *, -$, exact division, equality test, and gcd) in the coefficient ring $\mathbb{A}$ alone. Our approach naturally extends Yun's classical squarefree factorization for characteristic 0 [GG03, Algorithm 14.21], and is softly optimal if $\mathbb{A}$ is a field. In Section 3 we adapt the multi-modular strategy that is worthful whenever $\mathbb{A}$ is a multivariate polynomial ring over a field with sufficiently many elements. Finally Section 4 is devoted to applications of the new results to the complexities of the squarefree and the irreducible factorizations. We have implemented our algorithms within the `Magma` computer algebra system [Mag]. Our package, called `sepfact`, is available for download from `http://www.math.uvsq.fr/~lecerf/software`. We will report on performances on the multi-modular approach in Section 3.4.

Related Works. The separable decomposition is classical in the study of the irreducible factorization because its helps reducing the problem to separable polynomials (see Section 4). Seminal works in polynomial factorization date back to Kronecker in 1882 [Kro82]. Forty years later Hermann [Her26], followed by van der Waerden [Wae30, Wae49], Fröhlich and Shepherdson [FS55, FS56] in the fifties, and then Seidenberg [Sei70, Sei74, Sei78] and Richman [Ric81, MR82] solved all the constructibility and calculability issues of irreducible factorization. The separable factorization is present in the modern treatment of constructive mathematics by Richman and his collaborators (see the book [MRR88]).

Application to the Squarefree Factorization. If $\mathbb{A}$ is a perfect field then our softly optimal separable factorization algorithm, namely Algorithm 3 of Section 2, essentially coincides with the squarefree factorization algorithm of [GG03, Exercise 14.30]. In general, when $\mathbb{A}$ is not perfect, to the best of our knowledge, the separable factorization first clearly appeared in computer algebra in a work by Gianni and Trager on the squarefree factorization [GT96]. Their algorithm is derived from Musser's one [Mus71], hence it has an inherent quadratic cost.

Application to the Irreducible Factorization. The computer algebra literature often suggests that the irreducible factorization should be reduced to squarefree polynomials ([DT81, Gat84, BM97],...). Although this point of view is well suited to characteristic 0, several problems arise in positive characteristic, and one must consider the separable decomposition instead of the squarefree one.

In fact, if $\mathbb{A} = \mathbb{K}[x]$ is a univariate polynomial ring over some effective field $\mathbb{K}$. It is classical that the irreducible factorization in $\mathbb{A}[y]$ reduces to the one in $\mathbb{K}[y]$ efficiently by means of the Hensel lifting strategy [GG03, Chapter 15]. If the characteristic is 0 or is sufficiently large, and if F is squarefree then this strategy begins with finding a specialization point $a \in \mathbb{K}$ such that $F(a,y)$ remains squarefree. Now if $p > 0$ this strategy behaves badly. For instance, consider $F := (y^p + x)(y + x^p)$ with with $\mathbb{K}$ being the algebraic closure of $\mathbb{F}_p$ (the field with p elements). It is clear that F is squarefree, and that $F(a,y)$ is not squarefree for all $a \in \mathbb{K}$. By symmetry the same holds even if x and y are swapped. On the other hand $\mathrm{Sep}(F) = \{(y + x^p, 1, 1), (y + x, p, 1)\}$, and for all $a \in \mathbb{K}$, the polynomials $y + a^p$ and $y + a$ are separable.

In the factorization algorithm of multivariate polynomials over finite fields designed by Bernardin and Monagan in [BM97], this specialization problem in Hensel lifting is solved by using the separable factorization in a hidden manner: it turns out that their squarefree factorization algorithm, borrowed from [Ber97], internally performs the separable one. However the explicit algorithmic use of the separable factorization as a general pretreatment to the irreducible one first appeared in a recent algorithm due to Steel [Ste05] that handles all coefficient field being explicitly finitely generated over its prime field. Such an algorithm had been previously proposed by Davenport and Trager in [DT81], but it contained some gaps and imprecision pointed out by Steel. It is important to mention that Steel showed in [Ste05, Remark 3.4(1)] that the separable decomposition could be obtained by modifying standard squarefree factorization algorithms. Our contributions in this paper essentially concern the precise design of these modifications and the cost analysis.

For generalities on the irreducible factorization we refer the reader to [GG03], to Kaltofen's surveys [Kal82, Kal90, Kal92], and to [Gao01, BHKS04, Lec07a, Ste05, Lec06] for recent advances.

Further Related Works. Besides its impact towards the squarefree and the irreducible factorizations, the separable decomposition is useful in other contexts, as illustrated by Fortuna and Gianni with the computation of the Jordan normal form of a matrix [FG99]. Finally let us mention that the multi-modular approach developed in our Section 3 had already been treated by Gerhard for the special case when $\mathbb{A} = \mathbb{Z}$ (where the separable and squarefree decompositions coincide) [Ger01].

1. Separable Factorization

Throughout this section, $\mathbb{A}$ is a unique factorization domain of characteristic p, whose field of fractions is written $\mathbb{L}$, and F is a primitive polynomial in $\mathbb{A}[y]$ of degree $d \geq 1$.

We write $\mathrm{Sqr}(F)$ for the *squarefree decomposition* of F, that is the set of pairs (G, m) where G represents the squarefree factor of F with multiplicity $m \geq 1$ (i.e. the product of the irreducible factors of F with multiplicity m). We also define $\mathrm{Irr}(F)$ to be the *irreducible decomposition* of F, that is the set of the pairs (G, m), where G runs over the irreducible factors of F, and where $m \geq 1$ is the multiplicity of G in F, so that

$$F(y) = \prod_{(G,m)\in \mathrm{Irr}(F)} G^m.$$

The squarefree and irreducible factors are primitive and uniquely defined up to units in $\mathbb{A}$.

1.1. Separable and Inseparable Degrees.

We write $\deg^i(F)$ for the *inseparable degree* of F defined as the largest integer q in $\mathcal{B}$ such that $F \in \mathbb{A}[y^q] \setminus \mathbb{A}[y^{pq}]$.

Proposition 1. *Let F and G be primitive polynomials in $\mathbb{A}[y] \setminus \mathbb{A}$.*

a. *The inseparable degree of F is the largest power of p that divides all the multiplicities of the roots of F.*
b. $\deg^i(FG) \geq \min(\deg^i(F), \deg^i(G))$ *with equality whenever F and G are coprime.*

Proof. If $q \in \mathcal{B}$ divides the multiplicities of all the roots of F, then F clearly belongs to $\mathbb{A}[y^q]$. Conversely, if F rewrites into $G(y^q)$ with some $q \in \mathcal{B}$, then the multiplicity of a root of F must be a multiple of q. This proves part (a), which directly implies part (b). □

We define the *separable degree* of $F \in \mathbb{A}[y]$, written $\deg^s(F)$, as follows:

$$\deg^s(F) := \sum_{(G,m)\in \operatorname{Irr}(F)} \deg(G)/\deg^i(G) = \sum_{(G,m)\in \operatorname{Irr}(F)} \deg^s(G).$$

If G is irreducible then $\deg^i(G)$ (resp. $\deg^s(G)$) corresponds to the inseparable (resp. separable) degree of $\mathbb{L}[y]/(G(y))$ seen as a field extension of $\mathbb{L}$, so that $\deg(G) = \deg^i(G)\deg^s(G)$ holds. From this remark derives the following proposition:

Proposition 2. *Let F and G be primitive polynomials in $\mathbb{A}[y] \setminus \mathbb{A}$.*

a. $\deg^s(F)$ *equals the number of roots of F in $\bar{\mathbb{L}}$ without counting multiplicities.*
b. *F is separable if, and only if,* $\deg^s(F) = \deg(F)$.
c. *For all $q \in \mathcal{B}$,* $\deg^s(F(y^q)) = \deg^s(F)$.
d. $\deg^s(FG) \le \deg^s(F) + \deg^s(G)$, *with equality if, and only if, F and G are coprime.*
e. $\deg^s(F) = \displaystyle\sum_{(G,q,m)\in \operatorname{Sep}(F)} \deg(G)$.

1.2. **Deflation of Polynomials.** For any F in $\mathbb{A}[y] \setminus \mathbb{A}$, let us write $\tilde{F}$ for the *deflated polynomial* of F, that is uniquely defined by the following equality:

$$\tilde{F}(y^{\deg^i(F)}) := F(y).$$

Remark that the multiplicities of the roots of $\tilde{F}$ are the ones of F divided by $\deg^i(F)$, whence the terminology "deflation". Since $\tilde{F}$ belongs to $\mathbb{A}[y] \setminus \mathbb{A}[y^p]$, the following map Φ is well-defined:

$$\begin{aligned} \Phi : \mathbb{A}[y] \setminus \mathbb{A} &\to (\mathbb{A}[y] \setminus \mathbb{A}[y^p]) \times \mathcal{B} \\ F &\mapsto (\tilde{F}, \deg^i(F)). \end{aligned}$$

Now let $\mathcal{Q}$ be the set of the *qth powers of an irreducible polynomial* in $\mathbb{A}[y] \setminus \mathbb{A}$ with $q \in \mathcal{B}$, and let $\mathcal{S}$ be the set of the *separable irreducible polynomials* in $\mathbb{A}[y] \setminus \mathbb{A}$.

Proposition 3. *Φ is a bijection, and restricts to a bijection from $\mathcal{Q}$ onto $\mathcal{S} \times \mathcal{B}$.*

Proof. The fact that Φ is bijective is clear from the definitions. If $F \in \mathbb{A}[y]$ is irreducible, then $\tilde{F}$ is necessarily irreducible. From $\tilde{F} \notin \mathbb{A}[y^p]$ follows the separability of $\tilde{F}$. Since the deflated polynomial of F^q coincides with the one of $\tilde{F}^q$ for all $q \in \mathcal{B}$, Lemma 1 below implies that $\Phi(\mathcal{Q})$ is actually included in $\mathcal{S} \times \mathcal{B}$.

Conversely, let $(G, q) \in \mathcal{S} \times \mathcal{B}$. We shall show that $F(y) := G(y^q)$ belongs to $\mathcal{Q}$, and that $\deg^i(F) = q$. The latter equality is already clear since G is separable. Now let $h \le q$ be the largest element of $\mathcal{B}$ such that $G(y^h)$ is a hth power, and let $\tilde{G}$ be the corresponding hth root, so that we have $\tilde{G}^h(y) = G(y^h)$. By Lemma 1 below, $\tilde{G}$ is irreducible and separable. Proposition 1(b) thus implies that $\tilde{G}(y^{q/h})$ is irreducible, which concludes the proof since $F(y) = \tilde{G}(y^{q/h})^h$. □

Lemma 1. *Let F and G in $\mathbb{A}[y] \setminus \mathbb{A}$ be such that $G(y^p) = F(y)^p$.*

a. *G is primitive if, and only if, F is primitive.*
b. *G is separable if, and only if, F is separable.*
c. *If G is irreducible then F is irreducible. The converse holds if F or G is separable.*

Proof. Part (a) is clear. The multiplicities of $G(y^p)$ (resp. $F(y)^p$) are all p if, and only if, G (resp. F) is separable. Part (b) thus follows from Proposition 2. As for part (c), if G is irreducible, then for any irreducible proper factor A of F, we have that A^p divides F^p, hence that B divides G, where B is defined by $B(y^p) = A(y)^p$. It follows that B is a unit of $\mathbb{A}$, so is A. Conversely, assume that F is irreducible, and let A be a proper irreducible factor of G. Since $A(y^p)$ divides $F(y)^p$, the polynomial $A(y^p)$ rewrites into $F(y)^\alpha$ for some $\alpha \in \{0, \dots, p-1\}$. Differentiating thus yields $\alpha F' F^{\alpha-1} = 0$, whence $\alpha = 0$ whenever F is separable. □

1.3. **Existence and Uniqueness of the Separable Decomposition.** The proof of the following proposition makes use of the irreducible decomposition. The relationship between the irreducible and the separable decompositions established herein will be turned into an algorithm in Section 4. For a constructive proof we refer the reader to [MRR88, Chapter VI, Theorem 6.3].

Proposition 4. *Any primitive polynomial $F \in \mathbb{A}[y]$ admits a unique (up to units in $\mathbb{A}$) separable decomposition.*

Proof. Let $(F_1, e_1), \dots, (F_r, e_r)$ represent the irreducible decomposition of F in $\mathbb{A}[y]$. For each $i \in \{1, \dots, s\}$, let a_i be the largest power of p that divides e_i, and let $m_i := e_i/a_i$. We then define $(G_i, q_i) := \Phi(F_i^{a_i})$. The triples

$$(G_1, q_1, m_1), \dots, (G_s, q_s, m_s)$$

obtained in this way satisfy (S_1), (S_2) and (S_3) (as defined in the introduction), and property (S_4) follows from Proposition 3. In order to obtain (S_5) it suffices to gather the triples sharing the same second and third entries as follows: if $(A_1, q, m), \dots, (A_t, q, m)$ are such triples then merge them into $(A_1 \cdots A_r, q, m)$. Properties (S_1) to (S_4) are preserved, which concludes the existence proof.

Let $(G_1, q_1, m_1), \dots, (G_s, q_s, m_s)$ satisfy (S_1) to (S_5), and let $i \in \{1, \dots, s\}$. Any root in $\mathbb{L}$ of $G_i(y^{q_i})$ has multiplicity q_i, hence multiplicity $q_i m_i$ in F. Property (S_5) thus implies that the roots of $G_i(y^{q_i})$ are exactly those of F with multiplicity $q_i m_i$, which concludes the proof of the uniqueness. □

Corollary 1. *If $\mathbb{A}$ is contained in a unique factorization domain $\mathbb{B}$ then the separable factorization of F seen in $\mathbb{B}[y]$ coincides with the extension to $\mathbb{B}[y]$ of the separable factorization of F.*

2. Computation of the Separable Decomposition

This section in devoted to our fast separable factorization algorithm. This algorithm concerns any coefficient ring (that is a unique factorization domain) as soon as the ring operations (equality test, exact division, and gcd) are effective. For the cost analysis we will suppose that $\mathbb{A}$ is a field in order to benefit of the softly optimal gcd algorithm and other such fundamental operations.

2.1. **Computational Model.** Formally speaking, for our cost analysis, we use the *computation tree* model [BCS97, Chapter 4] from the *total complexity* point of view. This means that complexity estimates charge a constant cost for each arithmetic operation (+, −, ×, ÷) and the equality test. Yet all the constants in the base fields (or rings) of the trees are thought to be freely at our disposal.

Everywhere, a univariate polynomial of degree d is thought to be represented as the vector of its coefficients of size $d+1$. For each integer d, we assume that we are given a computation tree that computes the products of two polynomials of degree at most d with at most $\mathsf{M}(d)$ operations, independently of the base ring. As in [GG03, Chapter 8.3], for any positive integers d_1 and d_2, we assume that M satisfies the following properties: $\mathsf{M}(d_1 d_2) \le d_1^2 \mathsf{M}(d_2)$ and $\mathsf{M}(d_1)/d_1 \le \mathsf{M}(d_2)/d_2$

if $d_1 \leq d_2$. In particular, this implies the *super-additivity* of M, that is $\mathsf{M}(d_1) + \mathsf{M}(d_2) \leq \mathsf{M}(d_1 + d_2)$.

During the cost analyzes we will appeal to the following classical results without explicit references to them:

- the resultant and the extended greatest common divisor of two univariate polynomials of degree at most d over a field $\mathbb{L}$ can be computed with $\mathcal{O}(\mathsf{M}(d)\log(d))$ operations in $\mathbb{L}$ [GG03, Chapter 11];
- the product of r univariate polynomials $G_1, \ldots, G_r$ over a field $\mathbb{L}$ whose degree sum is d takes $\mathcal{O}(\mathsf{M}(d)\log(r))$ operations in $\mathbb{L}$ by means of the *subproduct tree* technique [GG03, Chapter 10];
- if $F \in \mathbb{L}[y]$ has degree d then the remainders of F modulo all the G_i can also be computed with $\mathcal{O}(\mathsf{M}(d)\log(r))$ operations in $\mathbb{L}$. This task is usually called the *simultaneous reduction*. The inverse problem, called *Chinese remaindering*, also costs $\mathcal{O}(\mathsf{M}(d)\log(r))$[GG03, Chapter 10].

We will use the classical $\tilde{\mathcal{O}}$ ("soft Oh") notation in the neighborhood of infinity as defined in [GG03, Chapter 25.7]. Informally speaking, "soft Oh"s are used for readability in order to hide logarithmic factors in cost estimates.

2.2. **The Induction Step.** The separable decomposition of F is to be obtained by induction via the following Lemma:

Lemma 2. *Assume that $p > 0$, and let F be a primitive polynomial in $\mathbb{A}[y]$. There exist unique (up to units in $\mathbb{A}$) polynomials S_0 and S_1 in $\mathbb{A}[y]$ such that the following properties hold:*

- *the irreducible factors of S_0 are separable and with multiplicity at most $p-1$;*
- *$F(y) = S_0(y)S_1(y^p)$.*

Proof. Let $S_0 \in \mathbb{A}[y]$ be defined by its irreducible factorization as follows:

$$\mathrm{Irr}(S_0) := \{(G, m \bmod p) \mid (G, m) \in \mathrm{Irr}(F),\ G \text{ is separable and } m \bmod p \neq 0\}.$$

Any irreducible factor G of F/S_0 is either inseparable or its multiplicity m in F is a multiple of p. In both cases G^m belongs to $\mathbb{A}[y^p]$, whence $F/S_0 \in \mathbb{A}[y^p]$. We can thus define S_1 to be the unique polynomial such that $S_1(y^p) = F(y)/S_0(y)$. We are done with the existence of the decomposition.

Let us now deal with the uniqueness. Let S_0 and S_1 be polynomials as in the lemma, and let G be an irreducible factor of S_0 with multiplicity $e \leq p-1$. Let m denote the multiplicity of G in F, so that $m - e$ is the multiplicity of G in $S_1(y^p)$. Then Proposition 1 implies that $\deg^i(G^{m-e}) \geq 1$, and thus that p divides $m - e$, since G is separable. Finally from $e \leq p-1$ we deduce that $e = m \bmod p$. □

Example 3. With Example 1 we have that $S_0 = (y+2)y^2$ and $S_1 = (y+1)(y+2)$.

Example 4. With Example 2 we have that $S_0 = y + 2x$ and $S_1 = (y^2 + x)(y + 2x^3)^2(y + 2x)^3$.

2.3. **Yun's Algorithm Revisited.** In this subsection S_0 and S_1 are the polynomials defined in Lemma 2. We first show that Yun's algorithm [GG03, Algorithm 14.21] called with F actually returns $\mathrm{Sqr}(S_0)$, that is also equivalent to $\mathrm{Sep}(S_0)$. Then we will proceed by induction in order to complete the separable factorization of F.

Algorithm 1. *Squarefree factorization of S_0.*

```
Input: a primitive polynomial $F \in \mathbb{A}[y]$ of degree $d$.
Output: $\mathrm{Sqr}(S_0)$ and $S_1$.
1. Let $l := 1$ and initialize $L$ with the empty list.
```

```
2. Compute $U := \gcd(F, F')$, $V := F/U$, and $W := F'/U$.
3. While $\deg(V) \geq 1$ do:
   a. Compute $H := \gcd(V, W - V')$, $W = (W - V')/H$, and $V := V/H$.
   b. If $\deg(H) \geq 1$ then append $(H, l)$ to $L$.
   c. Increment $l$ by 1.
4. Compute $S_0 := \prod_{(H,l) \in L} H^l$.
5. Compute $F/S_0$ and let $S_1$ be defined by $S_1(y^p) = (F/S_0)(y)$.
6. Return $L$ and $S_1$.
```

Lemma 3. *Algorithm 1 works correctly as specified. If $\mathbb{A}$ is a field then it takes $\mathcal{O}(\mathsf{M}(d)\log(d))$ operations in $\mathbb{A}$.*

Proof. The proof is essentially the same as the one of Yun's algorithm as done in [GG03, Theorem 14.23] except that [GG03, Lemma 14.22] has to be replaced by Lemma 4 below. However, for completeness, let us briefly recall this proof. Let V_l and W_l denote the respective values of V and W when entering step l of the while loop, and let H_l be the value of H computed during step l of the loop. Let n be the value of l when exiting the loop. By convention we also let V_n and W_n be the respective values of V and W when exiting the loop. We have to prove by induction on l from 1 to n that the following properties hold:

$$V_l = \prod_{\substack{(G,m)\in\mathrm{Sqr}(S_0)\\ m\geq l}} G, \qquad W_l = \sum_{\substack{(G,m)\in\mathrm{Sqr}(S_0)\\ m\geq l}} (m-l+1)\frac{V_l}{G}G'.$$

In step 2 we have:

$$U = \gcd(S_0, S_0')S_1(y^p), \qquad V = \frac{S_0}{\gcd(S_0, S_0')}, \qquad W = \frac{S_0'}{\gcd(S_0, S_0')}.$$

Let E_0 denote the squarefree part of S_0. From the squarefree decomposition of S_0 we calculate:

$$\begin{aligned}\gcd(S_0, S_0') &= \gcd\left(\prod_{(G,m)\in\mathrm{Sqr}(S_0)} G^m, \sum_{(G,m)\in\mathrm{Sqr}(S_0)} m\frac{S_0}{G}G'\right)\\ &= \gcd\left(E_0, \sum_{(G,m)\in\mathrm{Sqr}(S_0)} m\frac{E_0}{G}G'\right)\prod_{(G,m)\in\mathrm{Sqr}(S_0)} G^{m-1}\\ &= \prod_{(G,m)\in\mathrm{Sqr}(S_0)} G^{m-1}.\end{aligned}$$

The latter equality makes use of Lemma 4 below. It thus follows that the induction hypothesis holds when $l = 1$. Let us now assume that the induction hypothesis holds up to some $l \in \{1, \ldots, n-1\}$. Thanks to Lemma 4 again we obtain that:

$$\begin{aligned}H_l = \gcd(V_l, W_l - V_l') = \gcd&\left(\prod_{\substack{(G,m)\in\mathrm{Sqr}(S_0)\\ m\geq l}} G, \sum_{\substack{(G,m)\in\mathrm{Sqr}(S_0)\\ m\geq l}} (m-l)\frac{V_l}{G}G'\right)\\ = \prod_{\substack{(G,m)\in\mathrm{Sqr}(S_0)\\ m=l}} G,&\end{aligned}$$

from which the formulas for V_{l+1} and W_{l+1} easily follow. Since n is the first integer such that $\deg(V_n) = 0$, we deduce that $n-1$ is the largest multiplicity among the irreducible factors of S_0. Finally we have shown that L contains all the squarefree factors of S_0 at the end of the loop.

Step 2 takes $\mathcal{O}(\mathsf{M}(d)\log(d))$ operations in $\mathbb{L}$. Step l of the while loop costs $\mathcal{O}(\mathsf{M}(\deg(V_l))\log(\deg(V_l)))$ since $\deg(W_l) \leq \deg(V_l) - 1$. By the super-additivity of M, step 3 thus amounts to $\mathcal{O}\left(\mathsf{M}\left(\sum_{l\geq 1}^{n-1}\deg(V_l)\right)\log(d)\right)$ operations, and the conclusion follows from $\prod_{l=1}^{n-1} V_l = S_0$. Once $\mathrm{Sep}(S_0)$ is computed, the computation of S_0 takes $\mathcal{O}(\mathsf{M}(d)\log(d))$ operations, and then we need an extra $\mathcal{O}(\mathsf{M}(d))$ in order to deduce S_1. □

Lemma 4. *Let $G_1, \ldots, G_s$ be separable primitive polynomials in $\mathbb{A}[y] \setminus \mathbb{A}$, let $G := G_1 \cdots G_s$, and let $(c_1, \ldots, c_s) \in \mathbb{A}^s$. If $G_1, \ldots, G_s$ are pairwise coprime then the following equality holds:*

$$\gcd\left(G, \sum_{i=1}^{s} c_i \frac{G}{G_i} G_i'\right) = \prod_{c_i=0} G_i.$$

Proof. A straightforward calculation gives:

$$\gcd\left(G, \sum_{i=1}^{s} c_i \frac{G}{G_i} G_i'\right) = \prod_{j=1}^{s} \gcd\left(G_j, \sum_{i=1}^{s} c_i \frac{G}{G_i} G_i'\right) = \prod_{j=1}^{s} \gcd\left(G_j, c_j \frac{G}{G_j} G_j'\right).$$

The latter gcd equals G_j if $c_j = 0$ and is 1 otherwise since G_j' is prime with G_j. □

2.4. **Main Algorithm.** In order to obtain the separable decomposition of F, we will compute $\mathrm{Sqr}(S_0)$ and S_1 by the preceding algorithm. Then we shall recursively compute $\mathrm{Sep}(S_1)$, and finally merge these two decompositions into $\mathrm{Sep}(F)$. Let us start with explaining how this merging can be done fast. If

$$\begin{aligned} \mathrm{Sqr}(S_0) &= \{(G_1, m_1), \ldots, (G_r, m_r)\}, \\ \mathrm{Sep}(S_1) &= \{(H_1, q_1, n_1), \ldots, (H_s, q_s, n_s)\}, \end{aligned}$$

then the next algorithm computes all the $U_{i,j} := \gcd(G_i, H_j(y^{pq_j}))$ for $i \in \{1, \ldots, r\}$ and $j \in \{1, \ldots, s\}$. We will see that $(U_{i,j}, 1, m_i + pq_j n_j)$ is an element of $\mathrm{Sep}(F)$ whenever $\deg(U_{i,j}) \geq 1$. In order to compute all these gcd fast we appeal to simultaneous reductions. For this purpose we write

$$A \operatorname{pmod} B := \mathrm{lc}(B)^{\max(0, \deg(A) - \deg(B))} A \bmod B$$

for the *pseudo-remainder* of A divided by B. Here $\mathrm{lc}(B)$ represents the leading coefficient of B. Recall that the exact division in $\mathbb{A}$ is supposed to be effective.

Algorithm 2. *Merging* $\mathrm{Sqr}(S_0)$ *and* $\mathrm{Sep}(S_1)$.

```
Input: a primitive polynomial $F \in \mathbb{A}[y]$ of degree $d$, $\mathrm{Sqr}(S_0)$ and $\mathrm{Sep}(S_1)$.
Output: $\mathrm{Sep}(F)$.
1. If $S_1 \in \mathbb{A}$ then return $\{(G_1, 1, m_1), \ldots, (G_r, 1, m_r)\}$.
   Otherwise initialize $L$ with the empty list.
2. For all $i \in \{1, \ldots, r\}$ do
   a. For all $j \in \{1, \ldots, s\}$ compute $R_{i,j}(y) := \tilde{G}_i(y) \operatorname{pmod} H_j(y^{q_j})$, where $\tilde{G}_i$
      is the deflated polynomial of $G_i(y)^p$, so that $\tilde{G}(y^p) = G_i(y)^p$.
   b. For all $j \in \{1, \ldots, s\}$ compute $S_{i,j}(y) := \gcd(R_{i,j}(y), H_j(y^{q_j}))$.
   c. For all $j \in \{1, \ldots, s\}$ compute $T_{i,j}(y) := G_i(y) \operatorname{pmod} S_{i,j}(y^p)$.
   d. For all $j \in \{1, \ldots, s\}$ compute $U_{i,j}(y) := \gcd(T_{i,j}(y), S_{i,j}(y^p))$ and append
      $(U_{i,j}, 1, m_i + pq_j n_j)$ to $L$ if $\deg(U_{i,j}) \geq 1$.
   e. Compute $V_i(y) := \prod_{j=1}^{s} U_{i,j}(y)$.
   f. Compute $U_{i,0}(y) := G_i(y)/V_i(y)$ and append $(U_{i,0}, 1, m_i)$ to $L$ if
      $\deg(U_{i,0}) \geq 1$.
3. For all $j \in \{1, \ldots, s\}$ do
   a. Compute $W_j(y^{q_j}) := \prod_{i=1}^{r} S_{i,j}(y)^{q_j}$.
```

b. Compute $U_{0,j}(y) := H_j(y)/W_j(y)$ and append $(U_{0,j}, pq_j, n_j)$ to L if $\deg(U_{0,j}) \geq 1$.

4. Return L.

Lemma 5. *Algorithm 2 works correctly as specified. If $\mathbb{A}$ is a field then it takes $\mathcal{O}(\mathsf{M}(d)\log(d))$ operations in $\mathbb{A}$.*

Proof. As for the correctness, by Proposition 4, we have to check that the returned list L satisfies properties (S_1) to (S_5). For all $i \in \{1, \ldots, r\}$ and all $j \in \{1, \ldots, s\}$ we have that

$$R_{i,j}(y^p) = G_i(y)^p \operatorname{pmod} H_j(y^{pq_j}), \quad S_{i,j}(y^p) = \gcd(G_i(y)^p, H_j(y^{pq_j})),$$

hence that:

$$\begin{aligned} U_{i,j}(y) &= \gcd(G_i(y), S_{i,j}(y^p)) = \gcd(G_i(y), \gcd(G_i(y)^p, H_j(y^{pq_j}))) \\ &= \gcd(G_i(y), H_j(y^{pq_j})). \end{aligned}$$

At the end of step 2, for all $i \in \{1, \ldots, r\}$, it is clear that $G_i = \prod_{j=0}^{s} U_{i,j}$, and that the $U_{i,j}$ are separable and pairwise coprime.

In step 3 we have that

$$S_{i,j}(y)^{q_j} = \gcd(\tilde{G}_i(y)^{q_j}, H_j(y^{q_j})^{q_j}) = \gcd(\tilde{G}_i(y)^{q_j}, H_j(y^{q_j}))$$

since $\tilde{G}_i$ is separable by Lemma 1. It follows that $U_{0,j}$ is well-defined, and that

$$W_j(y^{pq_j}) = \prod_{i=1}^{r} S_{i,j}(y^p)^{q_j} = \prod_{i=1}^{r} \gcd(G_i(y)^{pq_j}, H_j(y^{pq_j})) = \prod_{i=1}^{r} U_{i,j}(y)^{pq_j}.$$

We deduce that $H_j(y^{pq_j}) = U_{0,j}(y^{pq_j}) \prod_{i=1}^{r} U_{i,j}(y)^{pq_j}$ holds for all $j \in \{1, \ldots, s\}$. Therefore F rewrites into:

$$\begin{aligned} F(y) = S_0(y)S_1(y^p) &= \left(\prod_{i=1}^{r}\prod_{j=0}^{s} U_{i,j}(y)^{m_i}\right) \prod_{j=1}^{s} \left(U_{0,j}(y^{pq_j})^{n_j} \prod_{i=1}^{r} U_{i,j}(y)^{pq_j n_j}\right) \\ &= \left(\prod_{i=1}^{r}\prod_{j=1}^{s} U_{i,j}(y)^{m_i + pq_j n_j}\right) \left(\prod_{i=1}^{r} U_{i,0}(y)^{m_i}\right) \left(\prod_{j=1}^{s} U_{0,j}(y^{pq_j})^{n_j}\right). \end{aligned}$$

The decomposition returned in L thus satisfies (S_1). The other properties are immediately satisfied by construction. We are done with the correctness.

Let us now analyze the cost of the algorithm. If $p \geq d+1$ then $S_1 \in \mathbb{A}$, and the algorithm does nothing. From now on we can assume that $p \leq d$. The following costs are direct consequences of the classical results recalled in Section 2.1, the super-additivity of M, the inequality $r \leq p-1$, and $\deg(S_0) + p\deg(S_1) = d$. Step 2a performs pth powering and simultaneous reductions. Its total cost amounts to

$$\begin{aligned} &\mathcal{O}\left(\sum_{i=1}^{r} \deg(G_i)\log(p) + \sum_{i=1}^{r}\Big(\mathsf{M}(\deg(S_1))\log(s) + \mathsf{M}(\max(\deg(G_i), \deg(S_1)))\Big)\right) \\ &\quad \subseteq \mathcal{O}\Big(\deg(S_0)\log(d) + p\mathsf{M}(S_1)\log(d) + \mathsf{M}(\deg(S_0))\Big) \subseteq \mathcal{O}(\mathsf{M}(d)\log(d)). \end{aligned}$$

The total cost of the gcd computed in steps 2b belongs to

$$\begin{aligned} &\mathcal{O}\left(\sum_{i=1}^{r}\sum_{j=1}^{s} \mathsf{M}(\deg(H_j(y^{q_j})))\log(\deg(H_j(y^{q_j})))\right) \\ &\quad \subseteq \mathcal{O}\left(p\mathsf{M}(\deg(S_1))\log(d)\right) \subseteq \mathcal{O}(\mathsf{M}(d)\log(d)). \end{aligned}$$

The total cost of the simultaneous reductions of steps 2c amounts to

$$\mathcal{O}\left(\sum_{i=1}^{r}\left(\mathsf{M}(\deg(G_i)) + \mathsf{M}\left(p\sum_{j=1}^{s}\deg(S_{i,j})\right)\log(d)\right)\right)$$
$$\subseteq \mathcal{O}\left(\mathsf{M}(\deg(S_0)) + \mathsf{M}(p\deg(S_1))\log(d)\right) \subseteq \mathcal{O}(\mathsf{M}(d)\log(d)).$$

The total cost of steps 2d belongs to

$$\mathcal{O}\left(\sum_{i=1}^{r}\sum_{j=1}^{s}\mathsf{M}(p\deg(S_{i,j}))\log(d)\right) \subseteq \mathcal{O}(\mathsf{M}(d)\log(d)).$$

The subproduct trees and divisions in steps 2e and 2f cost

$$\mathcal{O}\left(\sum_{i=1}^{r}\mathsf{M}(\deg(G_i))\log(d)\right) \subseteq \mathcal{O}(\mathsf{M}(d)\log(d)).$$

Finally the cost of step 3 is in

$$\mathcal{O}\left(\sum_{j=1}^{s}\mathsf{M}(\deg(H_j(y^{q_j})))\log(d)\right) \subseteq \mathcal{O}(\mathsf{M}(d)\log(d)).$$

□

Example 5. With Example 1 we enter Algorithm 2 with $\mathrm{Sqr}(S_0) = \{(y+2,1),(y,2)\}$ and $\mathrm{Sep}(S_1) = \{((y+1)(y+2),1,1)\}$, and obtain the following values of $U_{i,j}$:

$i\backslash j$	0	1
0		$(y+1)$
1	1	$(y+2)$
2	y	1

Example 6. With Example 2 we enter Algorithm 2 with $\mathrm{Sqr}(S_0) = \{(y+2x,1)\}$ and $\mathrm{Sep}(S_1) = \{(y^2+x,1,1),(y+2x^3,1,2),(y+2x^3,3,1)\}$, and obtain the following values of $U_{i,j}$:

$i\backslash j$	0	1	2	3
0		y^2+x	1	$y+2x^3$
1	1	1	$(y+2x)$	1

We are now ready to complete the separable factorization algorithm:

Algorithm 3. *Separable factorization.*

Input: a primitive polynomial $F \in \mathbb{A}[y]$ of degree d.
Output: $\mathrm{Sep}(F)$.

1. Compute $\mathrm{Sqr}(S_0)$ and S_1 with Algorithm 1.
2. Recursively call the present algorithm in order to compute $\mathrm{Sep}(S_1)$.
3. Call Algorithm 2 with $\mathrm{Sqr}(S_0)$ and $\mathrm{Sep}(S_1)$ on order to obtain $\mathrm{Sep}(F)$.

Proposition 5. *Algorithm 3 works correctly as specified. If $\mathbb{A}$ is a field then it takes $\mathcal{O}(\mathsf{M}(d)\log(d))$ operations in $\mathbb{A}$.*

Proof. The proof is a direct consequence of Lemmas 3 and 5, and of the following bound: $\sum_{k\geq 0}\mathsf{M}(d/p^k)\log(d/p^k) \in \mathcal{O}(\mathsf{M}(d)\log(d))$. □

3. Multi-modular Algorithm

If $\mathbb{A}$ is a multivariate polynomial ring over a field $\mathbb{K}$ then the separable factorization algorithm of the preceding section performs several pseudo-divisions and gcds. In practice, assuming that $\mathbb{K}$ has sufficiently many elements, such gcds computations behave well when computed via the multi-modular algorithm [GG03, Chapter 6], but the pseudo-divisions in Algorithm 3 provoke an expression swell for non-monic polynomials. In order to remedy this behavior, in this section, we propose a multi-modular separable factorization algorithm that even saves all the intermediate evaluations and interpolations involved within the gcd computations. In fact, and for a general unique factorization domain $\mathbb{A}$, we will compute the separable factorization of F modulo sufficiently many maximal ideals of $\mathbb{A}$, then the Chinese remainder theorem will allow us to reconstruct the result over $\mathbb{A}$. Of course it is necessary to require that $\mathbb{A}$ contains sufficiently many maximal ideals. Let us recall that the special case for when $\mathbb{A} = \mathbb{Z}$ has already been treated in [Ger01].

Throughout this section, F still denotes a primitive polynomial in $\mathbb{A}[y]$, and $\operatorname{lc}(F) \in \mathbb{A}$ represents the leading coefficient of F. If $\mathfrak{m}$ is a maximal ideal of $\mathbb{A}$, then we write $F \bmod \mathfrak{m}$ for the residue class of F in $\mathbb{A}/\mathfrak{m}[y]$. For convenience, we say that $\mathfrak{m}$ is *lucky* if $\deg(F \bmod \mathfrak{m}) = \deg(F)$, and if the image of $\operatorname{Sep}(F)$ in $\mathbb{A}/\mathfrak{m}[y]$ coincides with $\operatorname{Sep}(F \bmod \mathfrak{m})$, which precisely means that

$$\operatorname{Sep}(F \bmod \mathfrak{m}) = \{(G \bmod \mathfrak{m}, q, m) \mid (G, q, m) \in \operatorname{Sep}(F)\}.$$

Otherwise we say that $\mathfrak{m}$ is *unlucky*.

We will consider both deterministic and probabilistic points of views. Our probabilistic algorithms are seen as usual algorithms that depends on some parameters given as extra input. They will be of type Las Vegas, which means that the output is always correct. On the other hand the algorithm may stop prematurely without returning any answer. In the latter case we say that the given values of the parameters are unlucky. Our cost bounds concern both lucky and unlucky behaviors.

3.1. Characterization of Lucky Ideals.

Let $L = \{(G_1, q_1, m_1), \ldots, (G_s, q_s, m_s)\}$ be a subset of $(\mathbb{A}[y] \setminus \mathbb{A}) \times \mathcal{B} \times \mathbb{N}$ with all the G_i being primitive, and such that properties (S_1), (S_3) and (S_5) hold, and let

$$E(L) := \prod_{(G,q,m)\in L} G(y^q), \quad D(L) := \sum_{(G,q,m)\in L} \frac{E(L)}{G(y^q)} G'(y^q),$$
$$\Delta(L) := \operatorname{Res}(E(L), D(L)).$$

Lemma 6. *$\Delta(L) \neq 0$ if, and only if, $L = \operatorname{Sep}(F)$.*

Proof. By the multiplicativity of the resultant we have that (the following equalities hold up to nonzero factors in $\mathbb{A}$):

$$\begin{aligned}\Delta(L) &= \prod_{(G,q,m)\in L} \operatorname{Res}\left(G(y^q), \frac{E(L)}{G(y^q)} G'(y^q)\right)\\ &= \prod_{(G,q,m)\in L} \operatorname{Res}\left(G(y^q), \frac{E(L)}{G(y^q)}\right) \prod_{(G,q,m)\in L} \operatorname{Res}(G(y^q), G'(y^q)).\end{aligned}$$

Therefore (S_2) and (S_4) are satisfied if, and only if, $\Delta(L) \neq 0$. □

Lemma 7. *A maximal ideal $\mathfrak{m}$ is lucky if, and only if, $\operatorname{lc}(F)\Delta(\operatorname{Sep}(F)) \notin \mathfrak{m}$.*

Proof. For convenience, let us write $\operatorname{Sep}(F) \bmod \mathfrak{m}$ for

$$\{(G \bmod \mathfrak{m}, q, m) \mid (G, q, m) \in \operatorname{Sep}(F)\}.$$

If $\mathrm{lc}(F) \notin \mathfrak{m}$ then none of the leading coefficients of the separable factors G of F belongs to $\mathfrak{m}$, whence $\Delta(\mathrm{Sep}(F)) \bmod \mathfrak{m}$ is proportional to $\Delta(\mathrm{Sep}(F) \bmod \mathfrak{m})$. The conclusion thus follows from the previous lemma and from the uniqueness of the separable factorization (stated in Proposition 4). □

Proposition 6. *For all maximal ideal* $\mathfrak{m}$ *we have that* $\deg^s(F \bmod \mathfrak{m}) \leq \deg^s(F)$, *with equality if, and only if,* $\mathfrak{m}$ *is lucky.*

Proof. Let us first prove the lemma for when $F = G(y^q)^m$ with $q \in \mathcal{B}$, $m \bmod p \neq 0$, and G being separable. In this case Proposition 2 provides us with $\deg^s(F \bmod \mathfrak{m}) = \deg^s(G \bmod \mathfrak{m}) \leq \deg(G) = \deg^s(F)$, and

$$\deg^s(G \bmod \mathfrak{m}) = \sum_{(H,e,n) \in \mathrm{Sep}(G \bmod \mathfrak{m})} \deg(H).$$

From $G \bmod \mathfrak{m} = \prod_{(H,e,n) \in \mathrm{Sep}(G \bmod \mathfrak{m})} H(y^e)^n$, we deduce that the latter inequality is an equality if, and only if, $\mathfrak{m}$ is lucky for G.

Let us now deal with the general case. From Proposition 2 we have that:

$$\begin{aligned}\deg^s(F \bmod \mathfrak{m}) &\leq \sum_{(G,q,m) \in \mathrm{Sep}(F)} \deg^s(G \bmod \mathfrak{m}) \\ &\leq \sum_{(G,q,m) \in \mathrm{Sep}(F)} \deg^s(G) = \deg^s(F).\end{aligned}$$

The former inequality is an equality if, and only if, the $G(y^q) \bmod \mathfrak{m}$ are pairwise coprime. The latter inequality is an equality if, and only if, $\mathfrak{m}$ is lucky for all the G, which concludes the proof. □

3.2. **Algorithm.** Every finite set $\mathcal{M}$ of maximal ideals of $\mathbb{A}$ determines a projection

$$\begin{aligned}\rho_{\mathcal{M}} : \mathbb{A}[y] &\to \prod_{\mathfrak{m} \in \mathcal{M}} \mathbb{A}/\mathfrak{m}[y] \\ G &\mapsto (G \bmod \mathfrak{m} \mid \mathfrak{m} \in \mathcal{M}).\end{aligned}$$

Assuming that $\rho_{\mathcal{M}}$ is effective, we say that a map $\sigma_{\mathcal{M}}$ from $\prod_{\mathfrak{m} \in \mathcal{M}} \mathbb{A}/\mathfrak{m}[y]$ to $A[y]$ is an *effective section* of $\rho_{\mathcal{M}}$ if it is effective, and if $\rho_{\mathcal{M}}\sigma_{\mathcal{M}}$ is the identity map. For algorithmic purposes we shall assume that testing if an element of $\mathbb{A}$ actually belongs to an ideal of $\mathbb{A}$ is effective. For instance, in the case of bivariate polynomials treated in the next subsection, we will take $\mathbb{A} := \mathbb{K}[x]$ and $\mathcal{M}$ will be made of ideals of the form $(x - a)$ with $a \in \mathbb{K}$. Then $\rho_{\mathcal{M}}$ corresponds to the evaluation at the points of $\mathcal{M}$ while $\sigma_{\mathcal{M}}$ will be taken as the usual interpolation.

Algorithm 4. *Multi-modular separable factorization algorithm.*

Input: a primitive polynomial F in $\mathbb{A}[y]$, and $\mathcal{N}$ a finite subset of maximal ideals.

Output: $\mathrm{Sep}(F)$.

1. Initialize $\mathcal{M}$ with the subset of the ideals of $\mathcal{N}$ that do not contain $\mathrm{lc}(F)$.
2. For all $\mathfrak{m} \in \mathcal{M}$ compute $L^{\mathfrak{m}} := \mathrm{Sep}(F \bmod \mathfrak{m})$.
3. Compute $n := \max_{\mathfrak{m} \in \mathcal{M}} \deg^s(F \bmod \mathfrak{m})$ by means of the formula in Proposition 2(e), and remove from $\mathcal{M}$ all the ideals $\mathfrak{m}$ such that $\deg^s(F \bmod \mathfrak{m}) < n$.
4. If all the separable decompositions associated to the elements of $\mathcal{M}$ have different degree patterns, then stop the execution. Otherwise, this means that there exists a sequence of triples $(d_i, q_i, m_i)_{i \in \{1,\ldots,s\}}$ such that: $q_i m_i$ increases strictly, $d_i \geq 1$ for all i, and for all $\mathfrak{m} \in \mathcal{M}$ the list $L^{\mathfrak{m}}$ can be

reordered into $[(G_1^{\mathfrak{m}}, q_1, m_1), \ldots, (G_s^{\mathfrak{m}}, q_s, m_s)]$, with $\deg(G_i^{\mathfrak{m}}) = d_i$. Once these data are reorganized in this way, for an effective section $\sigma_{\mathcal{M}}$, compute:

$$L := \left[\left(\sigma_{\mathcal{M}}\left(\frac{\operatorname{lc}(F) \bmod \mathfrak{m}}{\operatorname{lc}(G_i^{\mathfrak{m}})} G_i^{\mathfrak{m}} \mid \mathfrak{m} \in \mathcal{M}\right), q_i, m_i\right) \mid i \in \{1, \ldots, s\}\right].$$

5. Replace each element (G, q, m) of L by $(\bar{G}, q, m)$, where $\bar{G}$ is computed as the primitive part of G.
6. If the coefficients of $\prod_{(G,q,m)\in L} G(y^q)^m$ and F are not all in the image of $\sigma_{\mathcal{M}}$ then stop the execution.
7. If the ratio of $\operatorname{lc}(F)$ with $\prod_{(G,q,m)\in L} \operatorname{lc}(G)^m$ is not a unit in $\mathbb{A}$ then stop the execution.
8. Return L.

Proposition 7. *Algorithm 4 either stops prematurely or returns a correct answer.*

Proof. Assume that the algorithm finishes normally. Let $\mathcal{L}$ be the value of $\mathcal{M}$ obtained at the end of step 3. By construction we have that

$$F/\operatorname{lc}(F) \bmod \mathfrak{m} = \prod_{(G,q,m)\in L} G(y^q)^m/\operatorname{lc}(G)^m \bmod \mathfrak{m}, \text{ for all } \mathfrak{m} \in \mathcal{L}.$$

Since $\operatorname{lc}(F)$ is guaranteed by step 7 to equal $\prod_{(G,q,m)\in L} \operatorname{lc}(G)^m$ up to a unit factor in $\mathbb{A}$ (that we discard for simplicity), we deduce that

$$F \bmod \mathfrak{m} = \prod_{(G,q,m)\in L} G(y^q)^m \bmod \mathfrak{m}, \text{ for all } \mathfrak{m} \in \mathcal{L}.$$

Since all the coefficients of F and $\prod_{(G,q,m)\in L} G(y^q)^m$ are in the image of $\sigma_{\mathcal{L}}$, the injectivity of $\sigma_{\mathcal{L}}$ imply that $F = \prod_{(G,q,m)\in L} G(y^q)^m$, that is property (S_1). The other properties (S_2) to (S_5) are clearly satisfied, which yields the correctness of the output. □

3.3. **Application to Bivariate Polynomials.** Let us now analyze the cost of the preceding multi-modular approach for when $\mathbb{A} := \mathbb{K}[x]$. For $\mathcal{N}$ we only consider ideals of the form $(x-a)$ with $a \in \mathbb{K}$. For all $\mathcal{M} = \{(x-a_1), \ldots, (x-a_n)\} \subseteq \mathcal{N}$, we set $\sigma_{\mathcal{M}}(b_1, \ldots, b_n)$ to be the unique polynomial B of degree at most $n-1$ such that $B(a_i) = b_i$ for all $i \in \{1, \ldots, n\}$. In order to avoid confusion we write $d_x := \deg_x(F)$ for the partial degree in x of $F \in \mathbb{K}[x][y]$, and $d_y := \deg_y(F)$ for its degree in y.

Proposition 8. *If $\mathbb{K}$ has cardinality at least $d_x(2d_y+1)+1$ then* $\operatorname{Sep}(F)$ *can be computed with $\mathcal{O}(d_y(d_y\mathsf{M}(d_x)\log(d_x) + d_x\mathsf{M}(d_y)\log(d_y)))$ or $\tilde{\mathcal{O}}(d_x d_y^2)$ operations in $\mathbb{K}$.*

Proof. We call Algorithm 4 with $\mathcal{N}$ being a set of $d_x(2d_y+1)+1$ maximal ideals in $\mathbb{K}[x]$ of the form $(x-a)$. Let $\mathcal{L}$ be the value of $\mathcal{M}$ when entering step 4. Since the number of roots of $\operatorname{lc}(F)\Delta(\operatorname{Sep}(F)) \in \mathbb{K}[x]$ is at most $2d_x d_y$, Proposition 6 implies that $\mathcal{L}$ is made of lucky elements and has cardinality at least d_x+1. On the other hand, for all $(G, q, m) \in \operatorname{Sep}(F)$, Lemma 8 below provides us with $\deg_x(\operatorname{lc}(F)G/\operatorname{lc}(G)) \le d_x$. It thus follows that $L = \operatorname{Sep}(F)$ when exiting step 5, and that the remaining steps of are useless.

Let us now analyze the cost. The evaluation of F in x at $\mathcal{O}(d_x d_y)$ points takes $\mathcal{O}(d_y^2\mathsf{M}(d_x)\log(d_x))$ operations in $\mathbb{K}$. Then steps 1 and 2 cost $\mathcal{O}(d_x d_y \mathsf{M}(d_y)\log(d_y))$ operations by Proposition 5. In step 4 it is sufficient to restrict to a subset of $\mathcal{M}$ containing only d_x+1 elements. The interpolations can thus be done with $\mathcal{O}(d_y\mathsf{M}(d_x)\log(d_x))$ operations. Finally the computation of the primitive parts amounts to $\mathcal{O}(d_y\mathsf{M}(d_x)\log(d_x))$ more operations in $\mathbb{K}$. □

Lemma 8. *If G divides F in $\mathbb{K}[x][y]$ then* $\deg_x(\operatorname{lc}(F)G/\operatorname{lc}(G)) \leq \deg_x(F)$.

Proof. The inequality follows from $\deg_x(\operatorname{lc}(F)G/\operatorname{lc}(G)) = \deg_x(F) + \deg_x(\operatorname{lc}(F)) - \deg_x(\operatorname{lc}(G)) - \deg_x(F/G)$ and $\deg_x(\operatorname{lc}(F/G)) \leq \deg_x(F/G)$. □

The main drawback of the algorithm underlying Proposition 8 is the computation of separable factorizations of $F \bmod \mathfrak{m}$ for too many $\mathfrak{m}$ than actually needed to interpolate the separable factors. In the next algorithm we appeal to a classical probabilistic strategy to decrease the size of $\mathcal{N}$ at the price of introducing a casual failure. The probabilities of success are to be formulated in terms of the following function:

$$\begin{aligned}\mathcal{E}(M,N) &:= \frac{1}{1+\left(\frac{M}{N-M}\right)^2}, \text{ for } N \geq 2M \\ &:= 0, \text{ for } N < 2M.\end{aligned} \tag{1}$$

Note that $\mathcal{E}(M,N) \geq 1/2$ whenever $N \geq 2M$.

Proposition 9. *If $\mathcal{N}$ is a set made of $\mathcal{O}(d_x)$ maximal ideals of the form $(x-a)$ with $a \in \mathbb{K}$, then Algorithm 4 performs $\mathcal{O}(d_y\mathsf{M}(d_x)\log(d_x) + d_x\mathsf{M}(d_y)\log(d_y))$ or $\tilde{\mathcal{O}}(d_xd_y)$ operations in $\mathbb{K}$. If at least d_x+1 elements of $\mathcal{N}$ are lucky then the algorithm finishes with the correct result. The density of the subsets $\mathcal{N}$ of a fixed set made of N maximal ideals of the preceding form having cardinality $2(d_x+1)$ and containing at least d_x+1 lucky ideals is at least $\mathcal{E}(2d_xd_y, N)$.*

Proof. The proof of the first paragraph can be done *mutatis mutandis* as in the proof of Proposition 8. Remark that step 6 only consists in testing if the cardinality of $\mathcal{M}$ is at least d_x+1, and if $\sum_{(G,q,m)\in L} qm \deg_x(G) \leq d_x$, which is negligible.

The probability estimate follows from Lemma 9 below, since we have already seen that the number of unlucky ideals is at most $2d_xd_y$ by Lemma 7. □

In other words the latter proposition tells us that the separable decomposition of a bivariate polynomial can be computed in softly optimal time by a probabilistic Las Vegas algorithm with a uniformly bounded probability of failure. If the cardinality of $\mathbb{K}$ is not sufficiently large then we can compute the separable decomposition of F in a sufficiently large extension of $\mathbb{K}$ instead, by Corollary 1.

Lemma 9. *Let $\mathcal{M} \subseteq \mathcal{N}$ be two sets of respective cardinalities M and N. For any $n \leq M$, the density of subsets of $\mathcal{N}$ of cardinality $2n$ having at most n elements in $\mathcal{M}$ is at least $\mathcal{E}(M,N)$.*

Proof. The case when $N < 2M$ is immediate, so we can assume that $N \geq 2M$ from now. The number of subsets of $\mathcal{N}$ of cardinality $2n$ with $k \leq n$ elements in $\mathcal{M}$ is $\binom{M}{k}\binom{N-M}{2n-k}$. We set:

$$A := \sum_{k=0}^{n} \frac{\binom{M}{k}\binom{N-M}{2n-k}}{\binom{N}{2n}} \qquad \text{and} \qquad B := \sum_{k=n+1}^{2n} \frac{\binom{M}{k}\binom{N-M}{2n-k}}{\binom{N}{2n}},$$

so that $A + B = 1$ holds. We have to prove that $A > \mathcal{E}(M,N)$. Letting

$$C(M,N,n,k) := \frac{M\cdots(M-k+1)(N-M)\cdots(N-M-(2n-k)+1)}{N\cdots(N-2n+1)},$$

A and B rewrite into:

$$A = \sum_{k=0}^{n} \binom{2n}{k} C(M,N,n,k) \qquad \text{and} \qquad B = \sum_{k=n+1}^{2n} \binom{2n}{k} C(M,N,n,k).$$

Table 1. Multi-modular speed-up

(d_x, d_y)	$\mathbb{F}_{5^{10}}(x)[y]$ (s)	$\mathbb{F}_{5^{10}}[x][y]$ (s)	pattern
(15,10)	0.010	0.040	(5, 1, 1), (1, 5, 1)
(30,20)	0.120	0.100	(10, 1, 1), (2, 5, 1)
(63,41)	3.300	0.470	(21, 1, 1), (4, 5, 1)
(126,107)	246.9	5.100	(42, 1, 1), (8, 5, 1), (1, 25, 1)

The terms in A vanish whenever $k \leq 2n - N + M - 1$, while the ones in B are zero whenever $k \geq M + 1$. Since $2n - N + M \leq 2n - M$ we have that

$$\begin{aligned} A &= \sum_{k=\max(0,2n-N+M)}^{n-1} \binom{2n}{k} C(M,N,n,k) + \frac{\binom{M}{n}\binom{N-M}{n}}{\binom{N}{2n}} \\ &> \sum_{k=\max(0,2n-M)}^{n-1} \binom{2n}{k} C(M,N,n,k), \text{ and} \end{aligned}$$

$$B = \sum_{k=\max(0,2n-M)}^{n-1} \binom{2n}{k} C(M,N,n,2n-k).$$

For all $k \in \{\max(0, 2n - M), \ldots, n - 1\}$, since $N - M \geq M$ we obtain that

$$\frac{C(M,N,n,k)}{C(M,N,n,2n-k)} = \frac{M\cdots(M-k+1)(N-M)\cdots(N-M-(2n-k)+1)}{M\cdots(M-(2n-k)+1)(N-M)\cdots(N-M-k+1)}$$

$$= \frac{(N-M-k)\cdots(N-M-(2n-k)+1)}{(M-k)\cdots(M-(2n-k)+1)} \geq \left(\frac{N-M}{M}\right)^{2(n-k)} \geq \left(\frac{N-M}{M}\right)^2,$$

whence $A > \left(\frac{N-M}{M}\right)^2 B$, which concludes the proof. □

When $N = 2M$ we recover [GG03, Exercise 6.31], and the density bound is sharp for n and N in the neighborhood of infinity. However, in general, numerical experiments show that our bound could be refined.

3.4. **Timings.** We have implemented the algorithm underlying Proposition 9. In Table 1 we report on our experiments for $\mathbb{K} := \mathbb{F}_{5^{10}}$ on a Pentium (M) 1.8GHz processor with `Magma V2.11-14`. Timings are given in seconds. The column $\mathbb{K}(x)[y]$ means that we ran Algorithm 3 straightforwardly with $\mathbb{A} = \mathbb{K}(x)$, while the second column contains timings for the multi-modular algorithm. In the last column called "pattern" we indicate $((\deg_y(G), q, m) \mid (G, q, m) \in \mathrm{Sep}(F))$. As expected, we observe that the multi-modular approach remedy the expression swell occurring in the straightforward method.

4. Applications

This section is devoted to the reductions of the squarefree and the irreducible factorizations to separable polynomials. These reductions are classical in characteristic 0, hence we focus on positive characteristic p. Let us recall that, on the contrary to the separable factorization, the squarefree and the irreducible ones can not in general be computed by means of arithmetic operations in the coefficient ring or field $\mathbb{A}$ alone. Indeed Fröhlich and Shepherdson have even shown that testing if an element is a pth power is not decidable in general [FS56, Section 7] (see also the example in [Gat84, Remark 5.10]). Therefore we need to extend our computational model with *pth root extraction*. Precisely, we assume that we can test if an element of $\mathbb{A}$ is a pth power in $\mathbb{A}$, and that we can compute the corresponding pth root whenever it exists.

4.1. **Squarefree Factorization.** The relationship between the separable and the squarefree factorizations has been precisely studied by Gianni and Trager in [GT96, Section 4]. We are to revisit their results with a goal towards complexity improvements.

Besides pth root extractions in $\mathbb{A}$, we need to enlarge the computational model with a routine called *"basic split"* that performs the following task: from any separable primitive polynomial $F \in \mathbb{A}[y]$, compute $F_i \in \mathbb{A}[y]$, and $F_s \in \mathbb{A}^p[y]$ of maximal possible degree such that $F = F_s F_i$. Here $\mathbb{A}^p$ stands for the pth powers of $\mathbb{A}$. For convenience we assume that the cost of "basic split" is bounded by a super-additive function written S.

From [GT96, Proposition 16] we know that pth root extraction and "basic split" are implied by *Seidenberg's condition* P, which was introduced in [Sei70]. If $\mathbb{A}$ is a field, by [MRR88, Theorem 3.1] the latter condition is equivalent to the extraction of pth roots in any explicitly finitely generated algebraic field extension of $\mathbb{A}$, which is further equivalent to the squarefree factorization over any explicitly finitely generated algebraic field extension of $\mathbb{A}$ by [GT96, Theorem in p. 13].

The "basic split" of F mostly corresponds to the squarefree factorization of $F(y^p)$. Precisely, after rewriting $F_s(y^p)$ into $\tilde{F}_s(y)^p$, we obtain that $F(y^p) = F_i(y^p)\tilde{F}_s(y)^p$, whence $\operatorname{Sqr}(F) = \{(F_i(y^p), 1), (\tilde{F}_s, p)\}$. Based on this "basic split" we reach the general case via the following lemma:

Lemma 10. *If F is a separable primitive polynomial in $\mathbb{A}[y]$, and if $k \geq 1$, then the squarefree factorization of $F(y^{p^k})$ costs $\mathcal{O}(\mathsf{S}(k\deg(F)))$ plus $\mathcal{O}(k\deg(F))$ extractions of pth roots.*

Proof. We follow [GT96, Proposition 13]. The case $k = 1$ is clear. If $k \geq 2$ then we write $F(y^{p^k}) = F_i(y^{p^k})\tilde{F}_s(y^{p^{k-1}})^p$. The latter decomposition takes $\mathsf{S}(\deg(F))$ plus $\mathcal{O}(\deg(F))$ pth root extractions. By [GT96, Corollary 4] $F_i(y^{p^k})$ is squarefree and we have that

$$\operatorname{Sqr}(F(y^{p^k})) = \{(F_i(y^{p^k}), 1)\} \cup \{(G, pq) \mid (G, q) \in \operatorname{Sqr}(\tilde{F}_s(y^{p^{k-1}}))\}.$$

By Lemma 1, $\tilde{F}_s$ is separable so that we can recursively compute the squarefree decomposition of $\tilde{F}_s(y^{p^{k-1}})$. The cost follows from the super-additivity of S. □

Algorithm 5. *Squarefree factorization.*

Input: a primitive polynomial $F \in \mathbb{A}[y]$ of degree d.
Output: $\operatorname{Sqr}(F)$.

1. Compute the separable decomposition $\{(G_1, q_1, m_1), \ldots, (G_s, q_s, m_s)\}$ of F.
2. For all $i \in \{1, \ldots, s\}$ compute the squarefree decomposition of $G_i(y^{q_i})$

   $$\{(G_{i,0}(y), q_i), (G_{i,1}(y^p), q_i/p), \ldots, (G_{i,\log_p(q_i)}(y^{q_i}), 1)\} := \operatorname{Sqr}(G_i(y^{q_i})).$$

   Here $\log_p$ represents the logarithm in base p.
3. Return $\left\{\left(\prod_{q_i m_i = np^j} G_{i,j}(y^{p^j}), n\right) \mid n \in \{1, \ldots, d\}\right\}$.

Proposition 10. *Algorithm 5 works correctly as specified. If $\mathbb{A}$ is a field then it costs $\mathcal{O}(\mathsf{M}(d)\log(d))$ operations in $\mathbb{A}$, $\mathcal{O}(\mathsf{S}(d))$ for "basic split", and $\mathcal{O}(d)$ extractions of pth roots.*

Proof. Step 1 takes $\mathcal{O}(\mathsf{M}(d)\log(d))$ operations in $\mathbb{A}$ by Proposition 5. The cost of step 2 follows the previous lemma and $\sum_{i=1}^{s} \log_p(q_i)\deg(G_i) \leq d$. By means of the subproduct tree technique, the last step amounts to $\mathcal{O}(\mathsf{M}(d)\log(d))$ more operations in $\mathbb{A}$. □

The following corollary is an other solution of [GG03, Exercise 14.30], yet with a slightly slower algorithm if one considers the constants hidden behind the $\mathcal{O}$:

Corollary 2. *If $\mathbb{A}$ is perfect field then the squarefree factorization of F can be computed with $\mathcal{O}(\mathsf{M}(d)\log(d))$ arithmetic operations in $\mathbb{A}$ and $\mathcal{O}(d)$ extractions of pth roots in $\mathbb{A}$. If $\mathbb{A} = \mathbb{F}_q$ (the finite field with q elements) then the latter cost amounts to $\mathcal{O}(\mathsf{M}(d)\log(d) + d\log(q/p))$ operations in $\mathbb{F}_q$.*

Proof. Since $\mathbb{A}$ is perfect, the "basic split" of a polynomial F returns $F_i = 1$ and $F_s = F$, hence S is negligible. On the other hand, one pth root extraction in $\mathbb{F}_q$ takes $\mathcal{O}(\log(q/p))$ operations in $\mathbb{F}_q$ via binary powering. □

With the notation being as in Section 3.3, let us now examine the case of bivariate polynomials over perfect fields. We are to focus on fast probabilistic multi-modular algorithms. We extend the computation tree model with a function that takes an integer n and a finite subset $\mathcal{N}$ of $\mathbb{K}$ as input, and that returns a random subset of $\mathcal{N}$ of cardinality n, assuming that the cardinality of $\mathcal{N}$ is at least n. The cost of this operation is assumed to be bounded by a super-additive function written $\mathsf{R}(n)$, that only depends on n. The probability distribution is supposed to be uniform in the space of subsets of $\mathcal{N}$ of cardinality n. *Until then end of this paper every tree that is executable on a given input computes the expected result.*

Over a perfect field, the "basic split" of the bivariate polynomial $F(x, y)$ is simply given by

$$F_s = \gcd\left(F, \frac{\partial F}{\partial x}\right), \qquad F_i = \frac{F}{F_s}.$$

The cost of this operation is a consequence of the following classical proposition:

Proposition 11. *For each bidegree (d_x, d_y), there exists a computation tree that takes two polynomials F and G in $\mathbb{K}[x, y]$ of bidegree (d_x, d_y) as input, and returns $D := \gcd(F, G)$, together with F/D and G/D. The tree takes $\mathcal{O}(d_x\mathsf{M}(d_y)\log(d_y) + d_y\mathsf{M}(d_x)\log(d_x))$ operations in $\mathbb{K}$, plus $\mathsf{R}(2(d_x + d_y + 1))$ for one random set generation. If this set is taken at random in a set of cardinality N, then the probability that the tree is executable on a given input is at least $\mathcal{E}((2d_y + 1)d_x, N)$.*

Proof. The case $d_x = 0$ or $d_y = 0$ is trivial. Then the general result is derived from [GG03, Algorithm 6.36] *mutatis mutandis*, with the help of Lemma 9. Note that F/D and G/D are actually computed in [GG03, Algorithm 6.36]. □

Based on this proposition we can adapt Lemma 10 to our bivariate case.

Lemma 11. *For each bidegree (d_x, d_y) and each integer $k \geq 1$, there exists a computation tree that takes a separable primitive polynomial $F \in \mathbb{K}[x][y]$ as input and returns $\mathrm{Sqr}(F(y^{p^k}))$. The tree takes $\mathcal{O}(d_x\mathsf{M}(d_y)\log(d_y) + d_y\mathsf{M}(d_x)\log(d_x))$ operations in $\mathbb{K}$, plus $\mathcal{O}(d_xd_y)$ root extractions in $\mathbb{K}$, and $\mathsf{R}(2k(d_x + d_y + 1))$ for generating random sets. If all these sets are taken uniformly at random in a subset of $\mathbb{K}$ of cardinality N, then the probability that the tree is executable on a given input F is at least $\mathcal{E}\left(\frac{p^k-1}{(p-1)p^{k-1}}(2d_y + 1)d_x, N\right)$.*

Proof. We follow the notation and the induction of Lemma 10. For when $k = 1$ one needs to perform $\mathcal{O}(d_xd_y)$ root extractions in $\mathbb{K}$. Then the degree in x of $\tilde{F}_s$ is at most d_x/p, which concludes the cost analysis since $\sum_{i=0}^{k-1} 1/p^i = \frac{p^k-1}{(p-1)p^{k-1}} \leq 2$. The bound of the probability of success is a consequence of Lemma 12 below. □

Finally Algorithm 5 adapted to the bivariate case provides us with:

Proposition 12. *For each bidegree (d_x, d_y), there exists a computation tree that takes a primitive polynomial $F \in \mathbb{K}[x][y]$ as input, and returns $\operatorname{Sqr}(F)$. The cost of the tree amounts to $\mathcal{O}(d_x \mathsf{M}(d_y)\log(d_y) + d_y \mathsf{M}(d_x)\log(d_x))$ or $\tilde{\mathcal{O}}(d_x d_y)$ operations in $\mathbb{K}$, plus $\mathcal{O}(d_x d_y)$ root extractions in $\mathbb{K}$, and at most $\mathsf{R}(4d_x + 4d_y + 2)$ for generating random sets. If all these sets are taken uniformly at random in a subset of $\mathbb{K}$ of cardinality N, then the probability that the tree is executable on a given input is at least $\mathcal{E}(2(3d_y + 1)d_x, N)$.*

Proof. For step 1 of Algorithm 5 we use Proposition 9. For step 2 we appeal to Lemma 11. The final step amounts to $\mathcal{O}(d_x \mathsf{M}(d_y)\log(d_y) + d_y \mathsf{M}(d_x)\log(d_x))$ operations in $\mathbb{K}$ by the multi-modular subproduct tree technique. The total numbers of operations in $\mathbb{K}$ and of pth root extractions easily follow. The cost for generating random sets is

$$\mathsf{R}(2(d_x + 1)) + \sum_{i=1}^{s} \mathsf{R}(2\log_p(q_i)(\deg_x(G_i) + \deg_y(G_i) + 1)) \leq \mathsf{R}(4d_x + 4d_y + 2).$$

The probability that the tree is executable on a given input is at least

$$\mathcal{E}(2d_x d_y, N)\prod_{i=1}^{s}\mathcal{E}(2(2\deg_y(G_i) + 1)\deg_x(G_i), N) \geq \mathcal{E}(2(3d_y + 1)d_x, N),$$

by Lemma 12 below. □

Lemma 12. *For all non-negative integers M_1, M_2, and N, we have that*

$$\mathcal{E}(M_1, N)\mathcal{E}(M_2, N) \geq \mathcal{E}(M_1 + M_2, N).$$

Proof. The case when $N < 2(M_1 + M_2)$ is trivial. The other case follows from a straightforward calculation that we omit here for the sake of conciseness. □

4.2. Irreducible Factorization.

In this subsection we revisit how the irreducible factorization reduces to the separable case, with a special interest in efficiency. For this purpose we enlarge our computational model with the irreducible factorization of separable polynomials in $\mathbb{A}[y]$.

We are to turn into an algorithm the relationship between the separable and the irreducible factorization seen in the proof of Proposition 4. We start with the computation of Φ^{-1}, as defined in Section 1.2.

Algorithm 6. *Computation of Φ^{-1}.*

```
Input: $(G, q) \in \mathcal{S} \times \mathcal{B}$.
Output: $(H, h) \in \mathbb{A}[y] \times \mathcal{B}$ with $H$ irreducible and such that $H^h = \Phi^{-1}(G, q)$.
1. Set $\tilde{G} := G$ and $h := 1$.
2. While $\tilde{G}(y^p)$ is a $p$th power and while $h < q$ do
       Replace $\tilde{G}$ by the $p$th root of $\tilde{G}(y^p)$ and multiply $h$ by $p$.
3. Return $(\tilde{G}(y^{q/h}), h)$.
```

Lemma 13. *Algorithm 6 works correctly as specified and costs $\mathcal{O}(\deg(G)\log_p(q))$ extractions of pth roots in $\mathbb{A}$.*

Proof. $\tilde{G}$ and h computed within the algorithm coincide to the one constructed in the proof of Proposition 3. The cost analysis is immediate. □

Algorithm 7. *Reduction of the irreducible factorization to the separable case.*

```
Input: a primitive polynomial $F \in \mathbb{A}[y]$ of degree $d$.
Output: $\operatorname{Irr}(F)$.
1. Compute the separable decomposition $\operatorname{Sep}(F)$ of $F$.
2. For all $(G, q, m) \in \operatorname{Sep}(F)$ compute the irreducible factorization of $G$.
```

3. Return

$$\bigcup_{(G,q,m)\in \mathrm{Sep}(F)} \Big\{(H,h,m)\,|H^h := \Phi^{-1}(\bar{G},q),$$

$$\text{for all irreducible factor } \bar{G} \text{ of } G\Big\}.$$

Proposition 13. *Algorithm 7 works correctly as specified. If* $\mathbb{A}$ *is a field, then it performs irreducible factorizations of polynomials in* $\mathbb{A}[y]$ *whose degree sum is at most* d*, plus* $\mathcal{O}(\mathsf{M}(d)\log(d))$ *arithmetic operations in* $\mathbb{A}$ *and* $\mathcal{O}(d)$ *extractions of* p*th roots in* $\mathbb{A}$*.*

Proof. The cost of the first step comes from Proposition 5 and the cost of the last step follows from the previous lemma. □

Finally, let us mention that this reduction to the separable case is a central ingredient of the irreducible bivariate polynomial factorization algorithm presented in [Lec07b].

References

[BCS97] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic complexity theory.* Springer-Verlag, 1997.

[Ber97] L. Bernardin. On square-free factorization of multivariate polynomials over a finite field. *Theoret. Comput. Sci.*, 187(1-2):105–116, 1997.

[BHKS04] K. Belabas, M. van Hoeij, J. Klüners, and A. Steel. Factoring polynomials over global fields. Manuscript available at http://arxiv.org/abs/math.NT/0409510, September 2004.

[BM97] L. Bernardin and M. B. Monagan. Efficient multivariate factorization over finite fields. In *Applied algebra, algebraic algorithms and error-correcting codes (Toulouse, 1997)*, volume 1255 of *Lecture Notes in Comput. Sci.*, pages 15–28. Springer, Berlin, 1997.

[DT81] J. H. Davenport and B. M. Trager. Factorization over finitely generated fields. In *SYMSAC '81: Proceedings of the fourth ACM symposium on Symbolic and algebraic computation*, pages 200–205. ACM Press, 1981.

[FG99] E. Fortuna and P. Gianni. Square-free decomposition in finite characteristic: an application to Jordan form computation. *SIGSAM Bull.*, 33(4):14–32, 1999.

[FS55] A. Fröhlich and J. C. Shepherdson. On the factorisation of polynomials in a finite number of steps. *Math. Z.*, 62:331–334, 1955.

[FS56] A. Fröhlich and J. C. Shepherdson. Effective procedures in field theory. *Philos. Trans. Roy. Soc. London. Ser. A.*, 248:407–432, 1956.

[Gao01] S. Gao. Absolute irreducibility of polynomials via Newton polytopes. *J. Algebra*, 237(2):501–520, 2001.

[Gat84] J. von zur Gathen. Hensel and Newton methods in valuation rings. *Math. Comp.*, 42(166):637–661, 1984.

[Ger01] J. Gerhard. Fast modular algorithms for squarefree factorization and Hermite integration. *Appl. Algebra Engrg. Comm. Comput.*, 11(3):203–226, 2001.

[GG03] J. von zur Gathen and J. Gerhard. *Modern computer algebra.* Cambridge University Press, second edition, 2003.

[GT96] P. Gianni and B. Trager. Square-free algorithms in positive characteristic. *Appl. Algebra Engrg. Comm. Comput.*, 7(1):1–14, 1996.

[Her26] G. Hermann. Die Frage der endlich vielen Schritte in der Theorie der Polynomideale. *Math. Ann.*, 95(1):736–788, 1926.

[Kal82] E. Kaltofen. Polynomial factorization. In B. Buchberger, G. Collins, and R. Loos, editors, *Computer algebra*, pages 95–113. Springer-Verlag, 1982.

[Kal90] E. Kaltofen. Polynomial factorization 1982–1986. In *Computers in mathematics (Stanford, CA, 1986)*, volume 125 of *Lecture Notes in Pure and Appl. Math.*, pages 285–309. Dekker, 1990.

[Kal92] E. Kaltofen. Polynomial factorization 1987–1991. In *LATIN '92 (São Paulo, 1992)*, volume 583 of *Lecture Notes in Comput. Sci.*, pages 294–313. Springer-Verlag, 1992.

[Kro82] L. Kronecker. Grundzüge einer arithmetischen theorie de algebraischen grössen. *J. reine angew. Math.*, 92:1–122, 1882.

[Lec06] G. Lecerf. Sharp precision in Hensel lifting for bivariate polynomial factorization. *Math. Comp.*, 75:921–933, 2006.

[Lec07a] G. Lecerf. Improved dense multivariate polynomial factorization algorithms. *J. Symbolic Comput.*, 42(4):477–494, 2007.

[Lec07b] G. Lecerf. New recombination algorithms for bivariate polynomial factorization based on Hensel lifting. Manuscript, 2007.

[Mag] The Magma computational algebra system for algebra, number theory and geometry. http://magma.maths.usyd.edu.au/magma/. Computational Algebra Group, School of Mathematics and Statistics, University of Sydney, NSW 2006 Australia.

[MR82] R. Mines and F. Richman. Separability and factoring polynomials. *Rocky Mountain J. Math.*, 12(1):43–54, 1982.

[MRR88] R. Mines, F. Richman, and W. Ruitenburg. *A course in constructive algebra*. Universitext. Springer-Verlag, 1988.

[Mus71] D. R. Musser. *Algorithms for Polynomial Factorization*. PhD thesis, C.S. Department, Univ. of Wisconsin, 1971.

[Ric81] F. Richman. Seidenberg's condition P. In *Constructive mathematics (Las Cruces, N.M., 1980)*, volume 873 of *Lecture Notes in Math.*, pages 1–11. Springer-Verlag, 1981.

[Sei70] A. Seidenberg. Construction of the integral closure of a finite integral domain. *Rend. Sem. Mat. Fis. Milano*, 40:100–120, 1970.

[Sei74] A. Seidenberg. Constructions in algebra. *Trans. Amer. Math. Soc.*, 197:273–313, 1974.

[Sei78] A. Seidenberg. Constructions in a polynomial ring over the ring of integers. *Amer. J. Math.*, 100(4):685–703, 1978.

[Ste05] A. Steel. Conquering inseparability: primary decomposition and multivariate factorization over algebraic function fields of positive characteristic. *J. Symbolic Comput.*, 40(3):1053–1075, 2005.

[Wae30] B. L. van der Waerden. Eine Bemerkung über die Unzerlegbarkeit von Polynomen. *Math. Ann.*, 102(1):738–739, 1930.

[Wae49] B. L. van der Waerden. *Modern Algebra. Vol. I.* Frederick Ungar Publishing Co., New York, N. Y., 1949.

Grégoire Lecerf, Laboratoire de Mathématiques (UMR 8100 CNRS), Université de Versailles Saint-Quentin-en-Yvelines, 45 avenue des États-Unis, 78035 Versailles, France

E-mail address: Gregoire.Lecerf@math.uvsq.fr