J. Symbolic Computation (1990) **9**, 281–299

Functional Decomposition of Polynomials: the Tame Case

JOACHIM VON ZUR GATHEN

Department of Computer Science, University of Toronto
Toronto, Ontario M5S 1A4, Canada

(*Received* 22 *June* 1988)

If g and h are polynomials of degrees r and s over a field, their functional composition $f = g(h)$ has degree $n = rs$. The functional decomposition problem is: given f of degree $n = rs$, determine whether such g and h exist, and, in the affirmative case, compute them. We first deal with univariate polynomials, and present sequential algorithms that use $O(n \log^2 n \log\log n)$ arithmetic operations, and a parallel algorithm with optimal depth $O(\log n)$. Then we consider the case where f and h are multivariate, and g is univariate. All algorithms work only in the "tame" case, where the characteristic of the field does not divide r.

1. Introduction

If F is a field and $g, h \in F[x]$, then $f = g \circ h = g(h) \in F[x]$ is their (functional) *composition*, and (g, h) is a (functional) *decomposition* of f. Given $f \in F[x]$, there exists an essentially unique *complete* decomposition $f = f_1 \circ f_2 \circ \cdots \circ f_k$, where $f_1, \cdots, f_k \in F[x]$ are indecomposable polynomials (we give references in Section 2). This result is valid if the characteristic p of F does not divide the degree of f.

We start with the following decomposition problem: given $f \in F[x]$ of degree n, and $r, s \in \mathbb{N}$ with $n = rs$, decide whether there exist $g, h \in F[x]$ of degrees r, s, respectively, such that $f = g \circ h$. For some time, this problem was considered to be computationally hard: the security of a cryptographic protocol was based on its hardness (Cade 1985, broken by Berkovits and Lidl & Niederreiter), and exponential-time algorithms (in characteristic zero) were given by Alagar & Thanh (1985) and Barton & Zippel (1985) (a first version of which appeared in 1976). Major progress was made by Kozen & Landau (1989), who presented the first polynomial-time algorithm, in 1986. Their algorithm runs in sequential time $O(n^2)$, and $O(\log^2 n)$ in parallel. Gutiérrez *et al.* (1989) present a similar algorithm.

The present paper continues the work of Kozen & Landau in several directions, using their basic method: faster sequential and parallel algorithms, Boolean computations,

This work was supported by National Science and Engineering Council of Canada, grant A2514, by DFG grant SFB 124, and by Fundación Andes, beca C-10246. Parts of this work were done during visits to Universität des Saarlandes, Saarbrücken, Germany, and to Pontificia Universidad Católica de Chile, Santiago, Chile, and as a Visiting Fellow at the Computer Sciences Laboratory, Australian National University, Canberra, Australia.

0747–7171/90/030281 + 19 $03.00/0

complete decompositions, and multivariate polynomials. A fast decomposition method in Section 2 uses $O(n \log^2 n \log\log n)$ arithmetic operations, and $O(n \log^2 n)$ if F supports a Fast Fourier Transform. We find a complete decomposition with $O(n^{1+\epsilon})$ operations, for any $\epsilon > 0$. Over $\mathbb{Q}$, our approach yields a polynomial bound on the binary length of intermediate results, and thus shows that the decision problem is in the Boolean complexity class P. A parallel algorithm of optimal (up to constant factors) depth $O(\log n)$ is in Section 3.

As an application, Section 4 gives a (random) polynomial-time algorithm to decide whether a *separated polynomial* $f_1(x) - f_2(y) \in F[x, y]$ has a nontrivial separated factor, assuming that F supports a (random) polynomial-time root-finding procedure.

In Section 5 we consider decompositions of the form $f = g \circ h$ with $f, h \in F[x_1, \ldots, x_m]$ and $g \in F[x]$. The first polynomial-time algorithm for this problem is in Dickerson (1987). We present a conceptually simple Newton approach that yields polynomial-time algorithms for densely presented inputs, and random polynomial time for inputs given by arithmetic circuits. For the important sparse representation, we have no polynomial-time results.

All results of this paper work only in the "tame" case where $p = \mathrm{char}(F)$ does not divide r. A subsequent paper (von zur Gathen 1988) deals with the "wild" case, where p divides r; Kozen & Landau (1989) also have results for that case. (The terminology of "tame" and "wild" is borrowed from number theory, regarding r as some "ramification index"; see e.g., Hasse 1980.) Some of the present results were reported in von zur Gathen, Kozen & Landau (1987).

2. Fast univariate decomposition

The subject of this and the next section is the following decomposition problem $\mathrm{DEC}_{n,r}^F$. We have a field F, integers $n, r \in \mathbb{N}$ with r dividing n, and $f \in F[x]$ of degree n. Let $s = n/r$. The problem is to decide whether there exist $g, h \in F[x]$ of degrees r, s, respectively, such that $f = g \circ h = g(h)$ is the composition of g with h, and, in the affirmative case, to compute g and h. f is *indecomposable* if no such g and h exist, with $2 \le r < n$. The "tame" case is when the characteristic p of F does not divide r. This paper deals only with the tame case.

For the question of uniqueness, note the following three types of ambiguous decompositions. For any $u \in F[x]$, $c, d \in F$, $c \neq 0$, and $r, m \ge 2$ we have $u \circ (cx+d) \circ ((x-d)/c) = u$, $(x^m \cdot u^r) \circ x^r = x^r \circ (x^m \cdot u(x^r))$, and $T_r \circ T_m = T_m \circ T_r \, (= T_{rm})$, where T_i is the ith Chebyshev polynomial. "Ritt's First Theorem" states that a *complete decomposition* $f = f_1 \circ \cdots \circ f_k$ with $f_1, \ldots, f_k$ indecomposable is unique up to these ambiguities, i.e., that any two complete decompositions can be obtained from each other using these equalities (Ritt (1922) for $F = \mathbb{C}$, Engstrom (1941) for $p = 0$, Fried & MacRae (1969a) for $p = 0$ or $p > n$).

Decompositions are intimately related to the intermediate fields between $F(f)$ and $F(x)$ (Ritt 1922, Dorey & Whaples 1974) and between F and a splitting field of f over F (Kozen & Landau 1989).

If $f = g \circ h$ and a and c are the leading coefficients of f and h, respectively, then an affine linear transformation yields

$$\frac{f}{a} = \left(\frac{1}{a} g(cx + h(0))\right) \circ \frac{h - h(0)}{c},$$

a *normal* decomposition of a monic polynomial into monic polynomials, where the second

composition factor has constant term zero. So we can assume that f, g, and h are monic, and that $h(0) = 0$. Denoting by $\mathcal{M} \subseteq F[x]$ the set of monic polynomials, we consider the relation of *normal decompositions*

$$\begin{aligned}\mathrm{DEC}^F_{n,r} \;=\; & \{(f,(g,h)) \in \mathcal{M} \times \mathcal{M}^2 : \\ & f = g \circ h,\ \deg f = n,\ \deg g = r,\ \text{and } h(0) = 0\}.\end{aligned}$$

Formally, the computational problem has $f \in \mathcal{M}$ and $r \in \mathbb{N}$ as input, and as output the set of all $(g,h) \in \mathcal{M}^2$ with $(f,(g,h)) \in \mathrm{DEC}^F_{n,r}$.

In the tame case, the algorithm of Kozen & Landau (1989) (or Corollary 2.3 below) implies that for every f there exists at most one such (g,h), so that we can view $\mathrm{DEC}^F_{n,r} : \mathcal{M} \longrightarrow \mathcal{M}^2$ as a partial function. Furthermore, the problem is rational, i.e., if $f \in F[x]$ and there exists a field extension $K \supseteq F$ and $(f,(g,h)) \in \mathrm{DEC}^K_{n,r}$, then in fact $g, h \in F[x]$ (Levi 1942). Both facts and Ritt's First Theorem may fail in the "wild" case, where p divides r. Incidentally, a variant of the new algorithms is implicit in Levi (1942), Section 2.

Brent & Kung (1978) deal with a different problem: given $g, h \in F[x]$ of degree at most n, compute the lowest n coefficients of $g \circ h$, and, assuming $g(0) = 0$ and $g'(0) = 1$, of the functional inverse of g. They give algorithms using time $O((n \log n)^{3/2})$, if F supports a Fast Fourier Transform. Ritzmann (1984) has a better bound for computing such compositions, in a different "numerical" model.

Our model of computation is the "arithmetic Boolean circuit", which uses indeterminate inputs (in our case the coefficients of f), constants from F, the arithmetic operations $+, -, *, /$, tests $a \stackrel{?}{=} 0$, binary Boolean operations on the resulting Boolean values, and selection gates that select one of two arithmetic values according to the value of a Boolean third input (see von zur Gathen 1986). In our analyses, we usually neglect the Boolean cost, since it is always dominated by the arithmetic cost.

In Theorem 2.7, we consider the ground field $\mathbb{Q}$ and inputs presented in binary and Boolean computations, say on a Turing machine or on Boolean circuits, and derive a polynomial bound on the binary length of intermediate results.

We will use the following well-known facts about some computational problems. Let $M = M_F : \mathbb{N} \longrightarrow \mathbb{R}$ be such that the product of two polynomials in $F[x]$ of degree at most n can be computed with $O(M(n))$ arithmetic operations. We can choose $M(n) = n \log n \log\log n$ (Schönhage 1977, Cantor & Kaltofen 1987), and $M(n) = n \log n$ if F supports a Fast Fourier Transform.

FACT 2.1. *(i)* [Inversion] *Given $f \in F[x]$ with $f(0) = 1$, one can compute $f^{-1} \bmod x^{n+1}$ with $O(M(n))$ operations.*

(ii) [Division with remainder] *Given $f, g \in F[x]$ with degree at most n and $g \neq 0$, one can compute $q, r \in F[x]$ such that $f = qg + r$ and $\deg r < \deg g$ with $O(M(n))$ operations.*

(iii) [Roots] *Given $f \in F[x]$ of degree n with $f(0) = 1$, and $r \in \mathbb{N}$ not divisible by $\mathrm{char}(F)$, one can compute the unique $h \in F[x]$ of degree at most n such that $h^r \equiv f \bmod x^{n+1}$ and $h(0) = 1$ with $O(M(n) \log r)$ operations.*

(iv) [Taylor expansion] *Given $f, h \in F[x]$ of degrees n, s respectively, let $r = \lceil n/s \rceil$. One can compute with $O(M(n) \log n)$ operations the unique $b_0, \ldots, b_r$ in $F[x]$ such that*

$$f = \sum_{0 \le i \le r} b_i h^i \text{ and } \deg b_i < \deg h \text{ for all } i.$$

(v) [Composition] *Given $g, h \in F[x]$ of degrees r, s, repectively, one can compute $g \circ h$ in $O(M(n)\log n)$ operations, where $n = rs$.*

PROOF. (i) and (ii) are in Borodin & Munro (1975), Section 4.4. (iii) follows with Newton iteration, as in Brent & Kung (1978). The generalized "Taylor expansion" (iv) of f around h is most familiar when $h = x - h_0$ is linear, and we have the usual Taylor expansion of f around h_0. A divide-and-conquer approach gives the following algorithm. Let $t = \lceil r/2 \rceil$. In the allowed cost $O(M(n)\log n)$, we can assume that $v = h^t$ has been computed. Then we compute $q, w \in F[x]$ with

$$f = qv + w \text{ and } \deg w < \deg v,$$

and then recursively solve "Taylor expansion" for (w, h) (yielding $b_0, \ldots, b_{t-1}$), and for (q, h) (yielding $b_t, \ldots, b_r$). (v) is in Brent & Kung (1978), Lemma 2.1. □

Using fast polynomial arithmetic as quoted above, we can implement the algorithm of Kozen & Landau (1989) very efficiently. For a polynomial $f = x^n + a_{n-1}x^{n-1} + \cdots + a_0 \in F[x]$, we denote by $\tilde{f} = a_0 x^n + \cdots + a_{n-1}x + 1 = x^n \cdot f(1/x)$ the *reversal* of f.

Algorithm **Univariate decomposition.**

Input: $f \in F[x]$ monic of degree $n = rs$, and $r \in \mathbf{N}$, not divisible by $\mathrm{char}(F)$.

Output: The unique normal decomposition (g, h) of f with $\deg g = r$, if such a decomposition exists, and "no decomposition" otherwise.

1. Let $\tilde{f}$ be the reversal of f, and compute $\tilde{h} \in F[x]$ of degree less than s with $\tilde{h}^r \equiv \tilde{f} \bmod x^s$ and $h(0) = 1$. Let $h = x^s \tilde{h}(1/x) \in F[x]$.
2. Compute $b_0, \ldots, b_r \in F[x]$ as in "Taylor expansion".
3. If $b_0, \ldots, b_r \in F$, set $g = \sum_{0 \le i \le r} b_i x^i \in F[x]$ and return (g, h). Otherwise return "no decomposition".

THEOREM 2.2. *Over any field F, algorithm* **Univariate decomposition** *correctly solves $\mathrm{DEC}^F_{n,r}$, if $\mathrm{char}(F)$ does not divide r. It can be implemented with $O(M(n)\log n)$ arithmetic operations.*

PROOF. Let $f \in F[x]$ be monic of degree $n = rs$. Clearly (g, h) returned by the algorithm is correct, because $f = g \circ h$, h is monic since $\tilde{h}(0) = 1$, and $h(0) = 0$ since $\deg \tilde{h} < s$. So assume that there is a decomposition $f = \hat{g} \circ \hat{h}$, with $\hat{g}, \hat{h}$ monic, and $\hat{h}(0) = 0$. Then f and $\hat{h}^r$ agree on the highest s terms, i.e., $\deg(f - \hat{h}^r) \le n - s$. Let $h_1 = x^s \cdot \hat{h}(1/x)$ be the reversal of $\hat{h}$. Then

$$x^n \hat{h}(1/x)^r = (x^s \hat{h}(1/x))^r = h_1^r,$$

$$\begin{aligned} \deg(f - \hat{h}^r) \le n - s &\iff x^n \cdot ((f - \hat{h}^r)(1/x)) \equiv 0 \bmod x^s \\ &\iff \tilde{f} - h_1^r \equiv 0 \bmod x^s. \end{aligned}$$

Since the solution to "Roots" is unique, we have $\tilde{h} = h_1$ and $h = \hat{h}$. Then also $g = \hat{g}$ is computed in step 3.

With the algorithms for "Roots" and "Taylor expansion" from Fact 2.1, the algorithm can be performed with $O(M(n)\log n)$ operations. □

The algorithm shows rationality (as quoted above) and uniqueness of normal univariate decomposition into two polynomials, which will be used in Section 4. The following corollary describes the structure of the set of all decompositions of a fixed polynomial into two decomposition factors, without restriction to normality. Let us call two decompositions $f = g_1 \circ h_1 = g_2 \circ h_2$ *similar* if they differ by an affine linear transformation, i.e., if there exist $c, d \in F$ such that $c \neq 0$, $g_1 = g_2(cx + d)$, and $h_1 = (h_2 - d)/c$.

COROLLARY 2.3. *Let F be a field, $f \in F[x]$ of degree n, and $r \in \mathbb{N}$ with $\mathrm{char}(F)$ not dividing r.*

(i) Any two decompositions $f = g_1 \circ h_1 = g_2 \circ h_2$ with $\deg g_1 = \deg g_2 = r$ are similar.

(ii) There exists at most one normal decomposition $f = g \circ h$ with $(f, (g, h)) \in \mathrm{DEC}^F_{n,r}$.

(iii) If $f = g \circ h$ is a decomposition over an extension field K of F, and $h = cx^s + \cdots + d \in K[x]$, where $s = n/r = \deg h$, then $g_1 = g(cx + d)$ and $h_1 = (h - d)/c$ are in $F[x]$, and $f = g_1 \circ h_1$.

PROOF. Let $f = g \circ h$ be any decomposition over K, $a \in F$ the leading coefficient of f, and s, c, d, g_1, h_1 as in (iii). Then

$$a^{-1} f = a^{-1} g_1 \circ h_1$$

is a decomposition, with $a^{-1}f$ and h_1 monic, so that also $a^{-1}g_1$ is monic. Thus this is the unique normal decomposition computed by the algorithm, all polynomials are over F, and also g_1 is over F. This proves all claims. □

Algorithm **Complete decomposition.**

Input: Monic $f \in \mathcal{M} \subset F[x]$ of degree n, not divisible by $\mathrm{char}(F)$.

Output: A complete decomposition of f into indecomposable polynomials.

1. Compute the prime factorization $n = p_1^{e_1} \cdots p_k^{e_k}$ of n. Let $d(n) = (e_1 + 1) \cdots (e_k + 1)$ be the number of divisors of n, and $r_1 = 1 < r_2 < \ldots < r_{d(n)} = n$ the divisors.

2. For $j = 2, \ldots, d(n) - 1$ solve the problem DEC^F_{n,r_j}, with input f. If the first decomposition $(f, (g, h)) \in \mathrm{DEC}^F_{n,r_j}$ is found, apply the algorithm recursively to h, with output $f_2, \cdots, f_k$ such that $h = f_2 \circ f_3 \circ \cdots \circ f_k$. [$f_1 = g$ then is indecomposable.]

3. Return $(f_1, f_2, \ldots, f_k)$.

THEOREM 2.4. *Let $\epsilon > 0$. If $\mathrm{char}(F)$ does not divide the degree n of f, algorithm* **Complete decomposition** *computes a complete decomposition of f into indecomposable polynomials with $O(n^{1+\epsilon})$ arithmetic operations.*

PROOF. Clearly the algorithm works correctly. Its cost is $O(d(n) \cdot M(n) \log n)$, since $\deg h$ in the recursive call is a proper divisor of n. We now use that $d(n) = O(n^\epsilon)$ (Hardy & Wright 1962, Theorem 317). □

The algorithm finds the *lexicographically first complete decomposition*, for which the degree sequence $(\deg f_1, \ldots, \deg f_k)$ is lexicographically smallest. Furthermore, each f_i is

monic and, except possibly f_1, has constant term zero. The uniqueness in Corollary 2.3 shows that this decomposition exists and is unique.

The number of distinct complete decompositions (with h monic and $h(0) = 0$) is not too large: any polynomial of degree n has at most n distinct decompositions, and if n is the product of the first k prime numbers, then the number $k!$ of decompositions of $f = x^n$ is $\omega(n^{1-\epsilon})$ for every $\epsilon > 0$.

OPEN QUESTION 2.5. Can one reduce polynomial multiplication to decomposition? Is it possible to improve the running time for decomposition further, say to $O(M(n))$? Given f, g, h, can one compute $g \circ h$, or at least test $f = g \circ h$, deterministically in time $O(M(n))$?

Note that polynomial multiplication is reducible to squaring (if $\mathrm{char}(F) \neq 2$), which is a special case of composition.

REMARK 2.6. We have stated Theorem 2.2 only for the case of a field F. The algorithm actually works for an arbitrary commutative ring F with 1, provided that r is a unit in F. Of course, the uniqueness of Corollary 2.3 (ii) may get lost, as in $x^2 \circ (x^2 + ex) = x^2 \circ x^2$, if $\mathrm{char}(F) = 2$ and $e^2 = 0$.

Over the fields of greatest importance in computer algebra, $\mathbb{Q}$ and finite fields, the algorithm can be executed in polynomial time also by Boolean computations, say on a Turing machine or on Booolean circuits. This is trivial over a finite field, where an arithmetic operation can be performed in polynomial time, namely with $O(k \log k \log\log k)$ Boolean operations if $\#F \leq 2^k$. Over $\mathbb{Q}$, we have to show that the binary length of intermediate results is polynomially bounded. Applying naive estimates to iterative methods—such as Kozen & Landau's for the decomposition problem, the Newton iteration used in Fact 2.1 (iii), or the divide-and-conquer for 2.1 (iv)—yields at best "quasi-polynomial" bounds like $n^{\log n}$ times input length. We now prove a polynomial bound.

We represent a rational number a as the quotient of two relatively prime integers $a = b/c$, and call $\max\{\log_2 |b|, \log_2 |c|\}$ the length $l(a)$ of a (assuming $b \neq 0$). Then a can be represented by a string of $O(l(a))$ bits. For a polynomial $f \in \mathbb{Q}[x]$, $l(f)$ is the maximum length of its coefficients. If $a \in \mathbb{Z}$ is the sum of m integers, each of length at most k, then $l(a) \leq \log m + k$.

THEOREM 2.7. *Suppose that $f \in \mathbb{Q}[x]$ is monic of degree n and has length $l(f) \leq k$ with $k \geq 2$. Then all rational numbers computed in the algorithm* **Univariate decomposition** *with input f and $r \geq 2$ have length at most $2n^3k$.*

PROOF. In the notation of the algorithm, write $\tilde{f} = 1 + f_1$, so that x divides f_1, and $f_1 = a^{-1}f_2 = a^{-1}bf_3$ with $a, b \in \mathbb{N}$, $f_2, f_3 \in \mathbb{Z}[x]$, f_3 primitive, and $\gcd(a, b) = 1$. Then $l(a) \leq nk$ and $l(f_2) \leq nk$. We consider the binomial expansion

$$\tilde{h} \equiv (1 + a^{-1}f_2)^{1/r} \equiv \sum_{0 \leq i} \binom{1/r}{i} a^{-i} f_2^i$$

$$\equiv \frac{1}{r^{s-1}(s-1)!a^{s-1}} \sum_{0 \leq i < s} r^{s-1}(s-1)! \binom{1/r}{i} a^{s-1-i} f_2^i \bmod x^s,$$

where the last sum is in $\mathbf{Z}[x]$. The denominator $u = r^{s-1}\,(s-1)!a^{s-1}$ has length $l(u) \leq (s-1)(nk+\log n)$. For the integer coefficients occurring in the sum, we have

$$\begin{aligned} & l\left((r^{s-1}(s-1)!\binom{1/r}{i}\right) \\ &= l\left(r^{s-1-i}\frac{(s-1)!}{i!}\cdot 1\cdot(1-r)\cdot(1-2r)\cdots(1-(i-1)r)\right) \\ &\leq (s-1-i)\log r + (s-1-i)\log s + i\log sr = (s-1)\log n. \end{aligned}$$

Recall that $j \in \mathbf{N}$ has $\binom{j-1}{i-1}$ ordered partitions into i positive integers. For any $i < s$, in the binomial expansion of f_2^i each coefficient of an x^j with $j < s$ is the sum of $\binom{j-1}{i-1} < 2^{s-1}$ terms, each of which is the product of at most i coefficients of f_2. This shows that

$$l(a^{s-1-i}f_2^i) \leq (s-1-i)nk + (s-1) + ink = (s-1)(nk+1).$$

Putting this together, we find $u\tilde{h}, uh \in \mathbf{Z}[x]$, and

$$\begin{aligned} l(uh) &= l(u\tilde{h}) \leq (s-1)\log n + (s-1)(nk+1) + \log s \\ &\leq (s-1)(nk+\log n+2), \\ l(h) &\leq \max\{(s-1)(nk+\log n), (s-1)(nk+\log n+2)\} \\ &= (s-1)(nk+\log n+2). \end{aligned}$$

The Taylor expansion problem

$$f = \sum_{0\leq i<r} b_i h^i + h^r,\ b_i \in \mathbf{Q}[x],\ \deg b_i < s$$

is—by equating coefficients of powers of x—equivalent to an $n \times n$-system of linear equations for the rational coefficients of $b_0, \cdots, b_{r-1}$. (We have normalized $b_r = 1$.) Each entry of this system is a coefficient of $f - h^r, h^0, \cdots, h^{r-1}$. Since $uh, uf \in \mathbf{Z}[x]$, multiplication by u^r of each equation yields an equivalent system S with only integral entries. For any $i \leq r$ we have

$$\begin{aligned} l(u^r h^i) &= l(u^{r-i}) + l((uh)^i) \\ &\leq (r-i)(s-1)(nk+\log n) + (n-1) + i(s-1)(nk+\log n+2) \\ &\leq n(nk+\log n) - 1 \end{aligned}$$

Thus each entry of S has length at most $n(nk+\log n)$. (The -1 covers the coefficients of $f - h^r$.) Cramer's rule and Hadamard's inequality imply that the (unique) solution to S has each component bounded in length by

$$\frac{n}{2}\log n + n\cdot n(nk+\log n) \leq 2n^3k,$$

so that $l(b_i) \leq 2n^3k$ for any i. □

This theorem easily generalizes to algebraic number fields instead of $\mathbf{Q}$.

3. Very fast parallel decomposition

Kozen & Landau (1986) observe that the general parallelization technique of Valiant *et al.* (1983) applies to their construction, and obtain an arithmetic algorithm of depth $O(\log^2 n)$, in the tame case. This section provides an algorithm of optimal (up to constant factors) depth $O(\log n)$.

For this result, we implement the algorithm of Section 2 fast in parallel. Eberly (1989) shows that division with remainder and "iterated product" of n polynomials of degree at most n can be computed in depth $O(\log n)$ on P-uniform arithmetic Boolean circuits over F. This will be our model for this section; we could also use Eberly's log-space uniform circuits of depth $O(\log n \log\log n)$. The problem is also in Boolean NC over $\mathbb{Q}$ (using bounds like Theorem 2.7) and over finite fields. (Due to the lack of fast parallel Boolean computations for the gcd of integers and of inversion in finite fields, we have to allow the "redundant representation" a/b with $a, b \in \mathbb{Z}$, $b \neq 0$, of field elements, without insisting on $\gcd(a,b) = 1$ over $\mathbb{Q}$ or $b = 1$ in $\mathbb{Z}_p$ (but: $0 \leq a, b < p$).) We start with the problem "Roots" of Fact 2.1 (iii). Note that Newton iteration would only yield depth $O(\log^2 n)$.

LEMMA 3.1. *"Roots" can be solved in depth $O(\log n)$.*

PROOF. By assumption, x divides $f_1 = f - 1$. We have

$$h \equiv (1+f_1)^{1/r} \equiv \sum_{0 \leq i} \binom{1/r}{i} f_1^i \equiv \sum_{0 \leq i < s} \binom{1/r}{i} f_1^i \bmod x^s.$$

The powers f_1^i can be computed in depth $O(\log n)$. In the model of "non-uniform" arithmetic circuits, the binomial coefficients can be considered as constants in F, and hence given for free. However, they can also be computed in F log-space uniformly, just using the constants 0 and 1 and field operations. This is trivial if $p = \mathrm{char}(F)$ is zero or at least s, by computing the numerator and denominator products separately, and then dividing. If $p < s$, Lemma 3.2 below says that we can replace $1/r$ by $u \in \mathbb{Z}$ in the binomial coefficient, if $ru \equiv 1 \bmod p^{l+1}$, where $l = \lfloor \log_p i \rfloor$. Then we can apply Lucas' (1877) formula:

$$\binom{u}{i} = \binom{u_0}{i_0} \cdots \binom{u_l}{i_l} \bmod p,$$

where $u = u_0 + u_1 p + \cdots + u_l p^l$ with $0 \leq u_j < p$ is the p-ary representation of u, and similarly for i. (The computation of l, u, u_j, i_j takes place in the "Boolean part" of the arithmetic Boolean circuit.) □

LEMMA 3.2. *Let $p \in \mathbb{N}$ be prime, $r, i, m, u, v \in \mathbb{Z}$ with p not dividing r, $i \geq 1$, $l = \lfloor \log_p i \rfloor$, $m > l$, $v \equiv ur \bmod p^m$, and $b = \binom{v/r}{i} \in \mathbb{Q}$. Then b is a p-adic integer (i.e., p does not divide the reduced denominator of b), and $b \equiv \binom{u}{i} \bmod p^{m-l}$ (i.e., the two sides differ by a multiple wp^{m-l} of p^{m-l}, where the reduced denominator of $w \in \mathbb{Q}$ is not divisible by p.)*

PROOF. It is convenient to use the ring

$$\mathbb{Z}_{(p)} = \{s/t \in \mathbb{Q} : s, t \in \mathbb{Z} \text{ and } p \nmid t\}$$

of p-adic integers, with $\mathbb{Z} \subseteq \mathbb{Z}_{(p)} \subseteq \mathbb{Q}$. For this proof, "divisibility" always refers to $\mathbb{Z}_{(p)}$. Choose a bijection

$$\phi : \{1, \ldots, i\} \longrightarrow \{u - i + 1, \ldots, u\}$$

such that

$$\forall j,\ 1 \le j \le i,\ p^l \mid j \Longrightarrow p^l \mid \phi(j).$$

Such a bijection exists. [Choosing some w with $1 \le w \le i$ and $p^l \mid u - i + w$, we can let ϕ map p^l to $u - i + w$, and then the other values preserving the order, except that a wrap-around occurs at u.]

For $1 \le j \le i$, let $\nu(j) = p^{-k}$ if $p^k \mid j$ and $p^{k+1} \nmid j$. Then $k \le l$, $j\nu(j) \in \mathbf{Z}_{(p)}$ is a unit, $(u - v/r)\nu(j) \equiv 0 \bmod p^{m-1}$, and

$$\begin{aligned}
\phi(j)\nu(j) &\equiv (\phi(j) - u + \frac{v}{r})\nu(j) \bmod p^{m-l}, \\
\binom{u}{i} &= \frac{\prod_{1\le j\le i}\phi(j)}{\prod_{1\le j\le i} j} = \frac{\prod_j \phi(j)\nu(j)}{\prod_j j\nu(j)} \\
&\equiv \frac{\prod_j(\phi(j) - u + \frac{v}{r})\nu(j)}{\prod_j j\nu(j)} = \binom{v/r}{i} = b \bmod p^{m-l}. \ \square
\end{aligned}$$

THEOREM 3.3. *Over any field F, the decomposition problem $\mathrm{DEC}^F_{n,r}$, with $\mathrm{char}(F)$ not dividing r, can be computed on an arithmetic Boolean circuit over F of depth $O(\log n)$.*

PROOF. Using algorithm **Univariate decomposition**, it is now sufficient to solve the "Taylor expansion" problem of Fact 2.1 (iv). If $0 \le i < s$, and

$$f = q_i h^i + r_i \text{ and } \deg r_i < i \deg h$$

is a division with remainder, then $b_i = q_i - hq_{i+1}$. All these computations can be done in depth $O(\log n)$, by Eberly (1989). $\square$

PROPOSITION 3.4. *If $\mathrm{char}(F)$ does not divide the degree n of f, a complete decomposition of f into indecomposable polynomials can be computed in depth $O(\log n)$.*

PROOF. In the notation of algorithm **Complete decomposition**, solve all problems DEC^F_{n,r_j} in parallel. Suppose that k decompositions $(g_1, h_1), \cdots, (g_k, h_k)$ are found, with degrees $(t_1, s_1), \cdots, (t_k, s_k)$. Order these decompositions so that

$$s_0 = 1 < s_1 < s_2 < \cdots < s_k < s_{k+1} = n.$$

Determine $m \le k$ so that $s_m \le n^{1/2} < s_{m+1}$. If $s_m \ge n^{1/3}$, decompose g_m and h_m recursively. If $s_m < n^{1/3}$ and $s_{m+1} \le n^{2/3}$, decompose g_{m+1} and h_{m+1} recursively. If $s_m < n^{1/3}$ and $s_{m+1} > n^{2/3}$, then compute the decomposition $g_m = e_1 \circ e_2$ with e_2 of minimal degree (at least 2), and decompose e_1 and h_m recursively (using $e_1 = x$ if g_m is indecomposable.) Since $f = e_1 \circ e_2 \circ h_m$, we have $\deg e_1 < n^{1/3}$ in this case; e_2 is indecomposable.

If $D(n)$ denotes the maximal depth for degrees up to n, we have

$$D(n) = O(\log n) + D(n^{2/3}),$$

from which $D(n) = O(\log n)$ follows. $\square$

We show by example that this procedure does not necessarily find the lexicographically first complete decomposition. If $i \ge 1$, $p \in \mathbf{N}$ a prime with $2^i < p \le 2^{2i}$, $n = 2^i p$, and $f = x^n$, then at the top level of the algorithm the first of the three cases occurs with

$$n^{1/3} \le s_m = 2^i \le n^{1/2} < s_{m+1} = p,$$

and the complete decomposition

$$f = g_m \circ h_m = x^p \circ x^{2^i} = x^p \circ x^2 \circ \cdots \circ x^2$$

is computed.

4. Separated polynomials

Let $f_1, f_2 \in F[x]$. Then $f_1(x) - f_2(y) \in F[x,y]$ is called a *separated polynomial.* Finding separated factors of separated polynomials is equivalent to simultaneous decomposition of two polynomials.

FACT 4.1. *(Fried & MacRae 1969b) Let $f_1, f_2, h_1, h_2 \in F[x] \setminus F$ have degrees n_1, n_2, s_1, s_2, respectively. Then $h_1(x) - h_2(y)$ divides $f_1(x) - f_2(y)$ in $F[x,y]$ if and only if there exists $g \in F[x]$ such that $f_i = g \circ h_i$ for $i = 1,2$. If this is the case, then* $\deg g = n_1/s_1 = n_2/s_2 \in \mathbb{N}$.

A separated factor $h_1(x) - h_2(y)$ is called *normal* if h_1 is monic and $h_1(0) = 0$; any separated factor can be made normal by the affine linear transformation

$$u(x,y)(h_1(x) - h_2(y)) = au(x,y)\left(a^{-1}(h_1(x) - h_1(0)) - a^{-1}(h_2(y) - h_1(0))\right),$$

where a is the leading coefficient of h_1. Let us call $\mathrm{SEP}_{n,r}^F$ the problem of determining, on input $r \in \mathbb{N}$ and two polynomials $f_1, f_2 \in F[x]$ of degree at most n, whether there exists an normal separated factor $h_1(x) - h_2(y)$ of $f_1(x) - f_2(y)$ in $F[x,y]$, with $\deg h_1 = \deg f_1/r$. The literature contains no polynomial-time algorithm to solve this problem, say over $\mathbb{Q}$ or finite fields. One method, requiring exponential time in the worst case, is to compute all irreducible factors of $f_1(x) - f_2(y)$ in $F[x,y]$, and test each product of these factors for being separated. Alagar & Thanh (1985) and Barton & Zippel (1985) based their (exponential-time) decomposition algorithms on this approach, with $f_1 = f_2$. We now turn the fact around and have the following fast algorithm for $\mathrm{SEP}_{n,r}^F$ in the tame case, using univariate decomposition.

Algorithm **Separated factors.**

Input: $r \in \mathbb{N}$ with $\mathrm{char}(F)$ not dividing r, and $f_1, f_2 \in F[x]$ of degrees n_1, n_2, respectively.

Output: All (h_1, h_2) with $h_1(x) - h_2(y)$ a proper normal separated factor of $f_1(x) - f_2(y)$ in $F[x,y]$, and h_1 and $h_2 \in F[x]$ of degrees n_1/r and n_2/r, respectively. If no such polynomials exist: "no separated factors".

1. For $i = 1,2$ do the following. Let a_i be the leading coefficient of f_i. Compute the normal decomposition $a_i^{-1} f_i = g_i \circ \hat{h}_i$ with $\deg g_i = r$, h_i monic and $\hat{h}_i(0) = 0$. Set $h_1 = \hat{h}_1$. If one of the two polynomials has no such decomposition, return "no separated factors" and stop.

2. Compute the roots $c_1, \cdots, c_t \in F$ of $x^r - a_2/a_1 \in F[x]$ $(0 \le t \le r)$. If $t = 0$, return "no separated factors" and stop.

3. Let $b_1, b_2 \in F$ be the coefficients of x^{r-1} in g_1, g_2, respectively. For $1 \le j \le t$ do steps 4 and 5.

4. Compute $d_j = (b_2 c_j - b_1)/r$.

5. If $a_2 g_2 = a_1 g_1(c_j x + d_j)$, then return h_1 and $h_2 = c_j \hat{h}_2 + d_j$.

6. If for no value of j step 5 was successful, return "no separated factors".

THEOREM 4.2. *Let $n = \max\{n_1, n_2\}$. The algorithm works correctly as described in "Output", and can be performed with $O(r^2 \log^4 r + n \log^2 n \log\log n)$ arithmetic operations in F, plus the computation of all rth roots in F of some $a \in F$.*

PROOF. If h_1 and h_2 are returned in step 5, one checks that $f_i = a_1 g_1 \circ h_i$ for $i = 1, 2$. Thus in view of Fact 4.1, it is sufficient to show:

$$\forall \tilde{g}, \tilde{h}_1, \tilde{h}_2 \in F[x] \ (\deg \tilde{g} = r, f_i = \tilde{g} \circ \tilde{h}_i \text{ for } i = 1, 2, \ \tilde{h}_1 \text{ monic with } \tilde{h}_1(0) = 0$$
$$\Longrightarrow \text{ for some value of } j, \ h_1 = \tilde{h}_1 \text{ and } h_2 = \tilde{h}_2 \text{ are returned in step 5.)}$$

So assume that $\tilde{g}, \tilde{h}_1, \tilde{h}_2$ satisfy the hypothesis, let $s = \deg \tilde{h}_2 = n_2/r$, and write $\tilde{h}_2 = \tilde{c}x^s + \cdots + \tilde{d}$. First note that Corollary 2.3(i) and the existence of $\tilde{g}, \tilde{h}_1, \tilde{h}_2$ imply that step 1 successfully computes $g_1, g_2, h_1, \hat{h}_2$. Furthermore, g_1 and g_2 are monic. From

$$g_1 \circ h_1 = a_1^{-1} f_1 = a_1^{-1} \tilde{g} \circ \tilde{h}_1,$$

$$g_2 \circ \hat{h}_2 = a_2^{-1} f_2 = a_2^{-1} \tilde{g} \circ \tilde{h}_2 = a_2^{-1} \tilde{g}(\tilde{c}x + \tilde{d}) \circ (\tilde{h}_2 - \tilde{d})/\tilde{c},$$

and uniqueness of normal decompositions (Corollary 2.3 (ii)) we find

$$a_1^{-1} \tilde{g} = g_1, \quad \tilde{h}_1 = h_1, \quad a_2^{-1} \tilde{g}(\tilde{c}x + \tilde{d}) = g_2, \quad (\tilde{h}_2 - \tilde{d})/\tilde{c} = \hat{h}_2.$$

Comparing coefficients of x^r and x^{r-1} in the third equation, and using the first equation, we obtain

$$a_2^{-1} a_1 \tilde{c}^r = 1, \quad a_2^{-1} \cdot a_1 \tilde{c}^{r-1}(b_1 + r\tilde{d}) = b_2.$$

This shows that $(\tilde{c}, \tilde{d})$ equals (c_j, d_j) for some $j \leq t$, and for this value of j, $h_1 = \tilde{h}_1$ and $h_2 = \tilde{c}\hat{h}_2 + \tilde{d} = \tilde{h}_2$ are returned.

The dominating computing costs occur in step 1, with $O(n \log^2 n \log\log n)$ operations in F, and step 5. The equality can be tested by substituting $r+1$ different values from F for x, at a cost of $O(r \log^2 r)$ arithmetic operations (Borodin & Munro 1975, Corollary 4.5.4). If F has at most r elements, we may have to perform this test over a field extension of degree at most $\log(r+1)$ over F, where one arithmetic operation can be done with $O(\log^2 r)$ operations in F. Thus for one value of j, the test has cost $O(r \log^4 r)$, for a total cost of $O(tr \log^4 r) = O(r^2 \log^4 r)$. □.

While a separated polynomial of degree n may have $2^n - 2$ proper factors, only few of these are (normal) separated:

COROLLARY 4.3. *Let F be a field, $r \in \mathbf{N}$, $f_1, f_2 \in F[x]$ of degrees n_1, n_2, respectively, $n = \max\{n_1, n_2\}$, $m = \gcd(n_1, n_2)$, $f = f_1(x) - f_2(y) \in F[x, y]$, and $\epsilon > 0$.*

(i) If $\mathrm{char}(F)$ does not divide r, then f has at most r normal separated factors $h_1(x) - h_2(y)$ with h_1 and $h_2 \in F[x]$ of degrees n_1/r and n_2/r, respectively.

(ii) If $F = \mathbb{Q}$, there are at most two normal separated factors of degrees n_1/r and n_2/r, and at most one if r is odd. They can be computed with $O(n \log^2 n \log\log n)$ arithmetic operations, plus one root extraction in $\mathbb{Q}$.

(iii) If $\mathrm{char}(F)$ does not divide m, then f has $O(m \log\log m)$ normal separated factors over F (of arbitrary degrees). They can be computed with $O(n^2 \log^4 n)$ operations in F, plus the extraction of $O(m \log\log m)$ roots in F.

(iv) If $F = \mathbb{Q}$, f has at most $2d(m) = O(m^\epsilon)$ normal separated factors, where $d(m)$ is the number of factors of m. They can be computed with $O(n^{1+\epsilon})$ operations in $\mathbb{Q}$, plus the extraction of at most $d(m)$ roots in $\mathbb{Q}$.

PROOF. (ii) a_2/a_1 has at most two rth roots in $\mathbb{Q}$. It is sufficient to compute one root c_1; $c_2 = -c_1$ is the other one if r is even.

(iii) We apply **Separated factors** with r running through all divisors of m, and use $\sum_{r|m} r = O(m \log\log m)$ (Hardy & Wright 1962, Theorem 323) and $\sum_{r|m} r^2 = O(m^2)$.

(iv) $d(m) = O(m^\epsilon)$ (Hardy & Wright 1962, Theorem 317). □

The algorithm **Separated factors** requires all rth roots in F of some $a \in F$. Is this really necessary? Let us call ROOT_r^F the problem of finding all rth roots of an input $a \in F$, where we consider r to be the input size.

THEOREM 4.4. *In the tame case, where $\mathrm{char}(F)$ does not divide r, ROOT_r^F is linear-time reducible to $\mathrm{SEP}_{r,r}^F$.*

PROOF. Let $x^r - a = f_1 \cdots f_m$ be a factorization into monic irreducible factors in $F[x]$, with $f_i = x - c_i$ linear for $1 \le i \le t$, and $f_{t+1}, \cdots, f_m$ nonlinear. For any $f \in F[x]$ of degree n, we denote by

$$\tilde{f} = y^n f(x/y) \in F[x, y]$$

the homogeneous version of f. Then $x^r - ay^r = \tilde{f}_1 \cdots \tilde{f}_m$, and for $c \in F$ we have

$$\begin{aligned} c \text{ is an } r\text{th root of } a &\iff x - c \mid x^r - a \text{ in } F[x] \\ &\iff x - cy \mid x^r - ay^r \text{ in } F[x, y] \\ &\iff x - cy \text{ is a separated factor of } x^r - ay^r. \end{aligned}$$ □

Since root-extraction is not a rational process, we now consider Boolean computations over a "computable field"; the most important cases are $\mathbb{Q}$, where roots are easy to compute in polynomial time, or a finite field, where univariate polynomials can be factored in random polynomial time (Berlekamp 1970). On the other hand, Fröhlich & Shepherdson (1955) exhibit computable fields of characteristic zero over which the existence of (square) roots of a given field element is undecidable.

COROLLARY 4.5. *(i) There exist fields F over which $\mathrm{SEP}_{n,r}^F$ is uncomputable.*

(ii) $\mathrm{SEP}_{n,r}^{\mathbb{Q}}$ can be computed in polynomial time.

(iii) If F is a finite field with q elements, then $\mathrm{SEP}_{n,r}^F$ can be computed by a probabilistic algorithm using time polynomial in $\log q$ and n.

PROOF. For (ii), use Theorem 2.7. □

EXAMPLE 4.6. It is a bit surprising that although decomposition is rational (Corollary 2.3) and separated factorization is equivalent to simultaneous decomposition of two univariate polynomials (Fact 4.1), separated factorization is **not** rational. The algorithm indicates how to generate examples such as the following. The separated polynomial $f = x^2 + y^2 + 10y + 25 \in \mathbf{Q}[x, y]$ is irreducible over $\mathbf{Q}$, but has the separated factorization $f = (x - iy - 5i) \cdot (x + iy + 5i)$ over $\mathbf{Q}[i] \subset \mathbf{C}$, where $i = \sqrt{-1}$. The corresponding simultaneous decomposition is $f_1 = x^2 = x^2 \circ x$, $f_2 = -x^2 - 10x - 25 = x^2 \circ (ix + 5i)$. In step 2 of the algorithm, we would obtain the normal decomposition $a_2^{-1} f_2 = x^2 + 10x + 25 = (x^2 + 10x + 25) \circ x$.

In general, we have the following description of all separated factorizations, rational or not. Let F be a field, and K an algebraic closure of F. Corollary 4.3 (i) and (iii), with F replaced by K, give a bound on the number of factorizations.

COROLLARY 4.7. *Let F, K be as above, $r \in \mathbf{N}$, $f_1, f_2 \in F[x]$ of degrees n_1, n_2, respectively, $n = \max\{n_1, n_2\}$, a_i the leading coefficient of f_i, for $i = 1, 2$, and $a = a_2/a_1$.*

(i) If h_1 and $h_2 \in K[x]$ have degrees n_1/r and n_2/r, respectively, and $h_1(x) - h_2(y)$ is a separated factor of $f_1(x) - f_2(y)$, then $h_1, h_2 \in F[b][x]$ for some $b \in K$ with $b^r = a$.

(ii) A representative set of all h_1, h_2 as in (i), up to conjugation over F, can be computed with $O(M(r)(r^2 \log^4 r + n \log^2 n \log\log n))$ operations in F, plus the factorization of $x^r - a$ in $F[x]$.

PROOF. (i) follows from (the proof of) Theorem 4.2. For (ii), let $x^r - a = g_1 \cdots g_m$ be a factorization into monic irreducible polynomials in $F[x]$, and $d_i = \deg g_i$. ($g_1, \ldots, g_m$ are pairwise distinct.) If $\alpha_i = x \bmod g_i$ is an rth root of a in $F_i = F[x]/(g_i)$, for any i, then any (h_1, h_2) as in (i) is found by Algorithm **Separated factors** with some α_i substituted for c_i in steps 4 and 5 (up to conjugates over F). One arithmetic operation in F_i can be simulated by $O(M(d_i))$ operations in F. Since $\sum_{1 \le i \le m} M(d_i) \le M(r)$ (assuming that $M(d_i)/d_i \le M(r)/r$, which is the case for the choice made in Section 2), the claim for the total cost follows. □

If $\varsigma \in K$ is a primitive rth root of unity, $b \in K$ with $b^r = a$, and $L = F[\varsigma, b] \subseteq K$, then $[L : F] \le r\phi(r)$, and all h_1 and h_2 as above are in $L[x]$. One operation in L can be simulated with $O(M(r\phi(r)))$ operations in F, and the cost of performing the algorithm for Corollary 4.7 in L would be as in (ii) above, except that $M(r)$ is replaced by $M(r\phi(r))$.

5. Multivariate polynomials

There are several generalizations of the decomposition problem to multivariate polynomials. We solve the following type of problem:

$$f = g \circ h,\ f, h \in F[x_1, \ldots, x_m],\ g \in F[x]$$

by a simple linearly converging Newton method: substitute for $x_2, \cdots, x_m$, solve the univariate problem, and lift the unique solution.

We first have to make some normalizations. For $\alpha \in F$ we have

$$g \circ h = [g \circ (x - \alpha)] \circ [(x + \alpha) \circ h],$$

so that we may assume throughout this section that $h(0,\ldots,0) = 0$. We define the set $\mathcal{M}_m$ of polynomials which are *strongly monic in* x_1 as follows:

$$\mathcal{M}_m = \{f = \sum_{0\le i\le n} f_i x_1^i \in F[x_1,\ldots,x_m]: \\ n \in \mathbf{N}, f_0,\cdots,f_n \in F[x_2,\ldots,x_m],\ f_n = 1,\ \deg f = n\},$$

where $\deg f$ is the total degree of f. Thus $\mathcal{M} = \mathcal{M}_1$ (identifying x with x_1). The *normal decomposition problem* is

$$\mathrm{DEC}^F_{n,r,m} = \{(f,(g,h)) \in \mathcal{M}_m \times (\mathcal{M} \times \mathcal{M}_m): \\ f = g\circ h,\ \deg f = n,\ \deg g = r,\ \text{and } h(0,\ldots,0) = 0\}.$$

Thus $\mathrm{DEC}^F_{n,r,1} = \mathrm{DEC}^F_{n,r}$. If $(f,(g,h)) \in \mathrm{DEC}^F_{n,r,m}$, then $f = g\circ h$ (or (g,h)) is called a *normal decomposition* of f. Corollary 5.3 (iv) below states that any f has at most one normal decomposition in the tame case.

We consider the ideal $\mathbf{x} \subseteq F[x_1,\ldots,x_m]$ generated by $x_2,\cdots,x_m$. An $\mathbf{x}$-homogeneous polynomial of degree d is of the form

$$\sum_{j_2+\cdots+j_m=d} a_j x_2^{j_2}\cdots x_m^{j_m},$$

with $a_j \in F[x_1]$; 0 is homogeneous of any degree. Every nonzero $f \in F[x_1,\ldots,x_m]$ has a unique $\mathbf{x}$-homogeneous representation $f = \sum_{i\le \deg f} f_i$ with f_i $\mathbf{x}$-homogeneous of degree i. We write $f_i = H_i(f)$.

The following algorithm solves the normal decomposition problem in the tame case.

Algorithm **Multivariate normal decomposition.**
Input: $f \in \mathcal{M}_m \subset F[x_1,\ldots,x_m]$ of degree $n = rs$, and $r \in \mathbf{N}$, not divisible by $\mathrm{char}(F)$.

Output: The unique normal decomposition (g,h) of f with $\deg g = r$, if such a decomposition exists, and "no decomposition" otherwise.

1. Compute $f_0 = f(x_1,0,\cdots,0)$, and $g \in F[x]$, $h_0 \in F[x_1]$ with $(f_0,(g,h_0)) \in \mathrm{DEC}^F_{n,r,1}$ by algorithm **Univariate decomposition**. If no such g, h_0 exist, return "no decomposition" and stop.

2. Set $s = n/r \in \mathbf{N}$, $k_0 = h_0$, and $t = (\partial g/\partial x)\circ h_0 \in F[x_1]$. [We will see that $t \neq 0$.]

3. For i from 1 to s perform step 4. [This Newton step determines $h_i = H_i(h)$ and $k_i = \sum_{j\le i} h_j$.]

4. Compute $u_i = H_i(f - g\circ k_{i-1})$. If t does not divide u_i (i.e., each coefficient at a monomial in $x_2,\cdots,x_m$), then return "no decomposition" and stop. Otherwise set $h_i = u_i/t$ and $k_i = k_{i-1} + h_i$.

5. Set $h = k_s$, and return (g,h) if $f = g\circ h$, and "no decomposition" otherwise.

THEOREM 5.1. *In the tame case, every $f \in \mathcal{M}_m \subset F[x_1,\ldots,x_m]$ has at most one normal decomposition. The algorithm correctly decides existence, and computes the normal decomposition, if it exists.*

PROOF. By assumption, p does not divide r, so that $g' = \partial g/\partial x \neq 0$, and since h_0 is a nonconstant polynomial, also $t = g' \circ h_0 \neq 0$ in step 2.

Let $f = \tilde{g} \circ \tilde{h}$ be any normal decomposition, and $f = \sum f_j$ and $\tilde{h} = \sum \tilde{h}_j$ the $\mathbf{x}$-homogeneous representations. It is sufficient to show that the algorithm outputs (g, h) with $g = \tilde{g}$ and $h_i = \tilde{h}_i$, by induction on i.

For $i = 0$, we have $(f_0, (g, \tilde{h}_0)) \in \mathrm{DEC}^F_{n,r,1}$, and Corollary 2.3 (ii) implies that $g = \tilde{g}$ and $h_0 = \tilde{h}_0$ are computed in step 1. Now let $i \geq 1$. Taylor expansion of g around an indeterminate y implies that there exists $G \in F[x, y]$ such that

$$g(x) = g(y) + g'(y) \cdot (x - y) + G \cdot (x - y)^2$$

holds in $F[x, y]$. Substituting $k_{i-1} = \sum_{j<i} \tilde{h}_j$ for y (where we have used the induction hypothesis) and $y + \tilde{h}_i$ for x we find

$$g \circ \tilde{h} \equiv g \circ \sum_{j \leq i} \tilde{h}_j \equiv g \circ k_{i-1} + (g' \circ k_{i-1}) \cdot \tilde{h}_i \bmod \mathbf{x}^{i+1},$$

since $(x - y)^2 = \tilde{h}_i^2 \equiv 0 \bmod \mathbf{x}^{i+1}$. Now

$$g' \circ k_{i-1} = g' \circ \sum_{j<i} h_j \equiv g' \circ h_0 = t \bmod \mathbf{x},$$

since $h_j \equiv 0 \bmod \mathbf{x}$ for $j \geq 1$. So we have

$$0 = f - g \circ \tilde{h} \equiv \sum_{j \leq i} f_j - (g \circ k_{i-1} + t\tilde{h}_i) \bmod \mathbf{x}^{i+1},$$

$$u_i = f_i - H_i(g \circ k_{i-1}) = t\tilde{h}_i.$$

This shows that u_i is divisible by t, and indeed $h_i = \tilde{h}_i$. □

For an arbitrary polynomial $f \in F[x_1, \ldots, x_m]$ we use substitutions σ of the form $\sigma f = f(x_1, x_2 + \sigma_2 x_1, \cdots, x_m + \sigma_m x_1)$, with $\sigma = (\sigma_2, \cdots, \sigma_m) \in F^{m-1}$ to make f strongly monic in x_1. If $h(0, \ldots, 0) = 0$, then also $(\sigma h)(0, \ldots, 0) = 0$. The substitution $\sigma^{-1} = (-\sigma_2, \cdots, -\sigma_m)$ is inverse to σ, with $\sigma^{-1}\sigma f = f$ for any $f \in F[x_1, \ldots, x_m]$.

Algorithm **Multivariate decomposition.**

Input: $f \in F[x_1, \ldots, x_m]$ of (total) degree $n = rs$, and $r \in \mathbb{N}$, not divisible by $\mathrm{char}(F)$.

Output: $g \in F[x]$ and $h \in F[x_1, \ldots, x_m]$ with $f = g \circ h$ and $\deg g = r$, if such a decomposition exists, and "no decomposition" otherwise.

1. Choose a substitution $\sigma \in F^{m-1}$ such that $\deg_{x_1} \sigma f = n$.

2. Let $a \in F$ be the leading coefficient of σf with respect to x_1, and $\tilde{f} = a^{-1}\sigma f$. [Then $\tilde{f} \in \mathcal{M}_m$.]

3. Call Algorithm **Multivariate normal decomposition** with input $\tilde{f}$. If no decomposition of $\tilde{f}$ exists, return "no decomposition". If $\tilde{f} = \tilde{g} \circ \tilde{h}$ is returned, return $g = a\tilde{g}$ and $h = \sigma^{-1}\tilde{h}$, where $\sigma^{-1} = (-\sigma_2, \cdots, -\sigma_m)$ is the substitution inverse to σ.

THEOREM 5.2. *Assume that a substitution σ is chosen as required in step 1 of the algorithm. Then the algorithm correctly determines whether $f \in F[x_1, \ldots, x_m]$ has a decomposition with the required degrees, and if so, computes a decomposition.*

PROOF. Clearly any answer (g, h) of **Multivariate decomposition** is indeed a decomposition $f = g \circ h$ with the required degrees. Now suppose that $f \in F[x_1, \ldots, x_m]$ has a decomposition $f = \hat{g} \circ \hat{h}$, with the required degrees, and $\hat{h}(0, \ldots, 0) = 0$. Let c be the leading coefficient of $\sigma\hat{h}$ with respect to x_1. Since $\sigma f = \hat{g} \circ \sigma\hat{h}$ has degree n in x_1, c is in F. $\tilde{f}$ is in $\mathcal{M}_m$, and

$$\tilde{f} = a^{-1}\sigma f = a^{-1}\hat{g} \circ \sigma\hat{h} = a^{-1}\hat{g}(cx) \circ c^{-1}\sigma\hat{h}$$

is a normal decomposition, with $\tilde{f}$ and $c^{-1}\sigma\hat{h}$ monic in x_1, and $c^{-1}\sigma\hat{h}(0, \ldots, 0) = 0$. Thus also $a^{-1}\hat{g}(cx)$ is monic, and the unique solutions $\tilde{g} = a^{-1}\hat{g}(cx)$ and $\tilde{h} = c^{-1}\sigma\hat{h}$ are returned by **Multivariate normal decomposition**. Thus the algorithm returns the correct decomposition $f = g \circ h = \hat{g}(cx) \circ c^{-1}\hat{h}$. □

A "lucky substitution" σ in step 1 is easy to find:

LEMMA 5.3. *Let $f \in F[x_1, \ldots, x_m]$ have degree n, $A \subseteq F$ finite, and a substitution σ uniformly chosen at random in A^{m-1}. Then $\deg_{x_1} \sigma f = n$ with probability at least $1 - n/\#A$.*

PROOF. For $0 \leq i \leq n$, let $u_i \in F[x_2, \cdots, x_m]$ be the homogeneous part of (highest) degree $n - i$ of the coefficient of x_1^i in f. Thus the homogeneous part of f of (total) degree n is $\sum u_i x_1^i \neq 0$, and by the homogeneity, also $u = \sum u_i \in F[x_2, \cdots, x_m]$ is nonzero, and of degree at most n. Now we have for $\sigma \in F^{m-1}$

$$\begin{aligned} &\deg_{x_1} \sigma f = n \\ &\quad \Longleftrightarrow \quad \deg f(x_1, \sigma_2 x_1, \cdots, \sigma_m x_1) = \deg[(\sigma f)(x_1, 0, \cdots, 0)] = n \\ &\quad \Longleftrightarrow \quad u(\sigma_2, \cdots, \sigma_m) \neq 0. \end{aligned}$$

The claim now follows from Schwartz (1980). □

Over small fields, no lucky substitutions may exist. For example, $f = x_1^2 x_2 + x_1 x_2^2 \in \mathbf{Z}_2[x_1, x_2]$ has no substitution $\sigma \in \mathbf{Z}_2$ with $\deg_{x_1} \sigma f = 3$.

We now describe the structure of the set of multivariate decomposition of a fixed polynomial, and show rationality as in the univariate case. Let $n = rs \in \mathbf{N}$. Consider two decompositions $f = g_1 \circ h_1 = g_2 \circ h_2$, with $g_1, g_2 \in F[x]$, $f, h_1, h_2 \in F[x_1, \ldots, x_m]$, $\deg f = n > r = \deg g_1 = \deg g_2 \geq 2$, and, as always, $h_1(0, \cdots, 0) = h_2(0, \cdots, 0) = 0$. Let us call the two decompositions *similar* if there exists $c \in F \setminus \{0\}$ such that $g_2 = g_1(cx)$ and $h_2 = c^{-1}h_1$. The constant of similarity c is uniquely determined.

COROLLARY 5.4. *In the tame case, where $\mathrm{char}(F)$ does not divide r, the following hold.*

(i) Any two multivariate decompositions are similar.

(ii) If the first decomposition factors g_1, g_2 are monic, then the constant of similarity is an rth root of unity.

(iii) If $c \in F$ is an rth root of unity and $f = g \circ h$ a decomposition with g monic, then $f = g(cx) \circ c^{-1}h$ is a decomposition of the same form.

(iv) Any $f \in \mathcal{M}_m$ has at most one normal decomposition $f = g \circ h$ with $h \in \mathcal{M}_m$.

(v) If $f = g \circ h$ is a decomposition over some extension field K of F, $h(0, \cdots, 0) = 0$, and $c \in K$ a nonzero coefficient of h, then $g(cx)$ and $c^{-1}h$ have coefficients in F.

PROOF. (i) Suppose that we have two decompositions $f = g_1 \circ h_1 = g_2 \circ h_2$, and first assume that $\#F > n = \deg f$. By Lemma 5.3 there exists a substitution $\sigma \in F^{m-1}$ such that $\deg_{x_1} \sigma f = n$. The proof of Theorem 5.2 shows that both decompositions are similar to the output of **Multivariate decomposition**, and thus similar to each other. In the general case, consider a field extension $K \supset F$ with $\#K > n$. The above implies that $g_2 = g_1(cx)$ and $h_2 = c^{-1}h_1$ for some $c \in K \setminus \{0\}$. Comparing coefficients in h_1 and h_2 of some monomial that occurs with nonzero coefficients (both in F) shows that $c \in F$.

(v) Suppose that $f = g \circ h$ is a decomposition over some $K \supset F$, with $h(0, \cdots, 0) = 0$. First assume $\#F > n$, and let $\sigma \in F^{m-1}$ be chosen in step 1 of the algorithm with $\deg_{x_1} \sigma f = n$. Let $a \in F \setminus \{0\}$ and $b \in K \setminus \{0\}$ be the leading coefficients with respect to x_1 of σf and σh, respectively. Then

$$\frac{\sigma f}{a}(x, 0, \cdots, 0) = \frac{g}{a} \circ \sigma h(x, 0, \cdots, 0) = \frac{g}{a}(bx) \circ b^{-1}\sigma h(x, 0, \cdots, 0),$$

and $g(bx)$ and $b^{-1}\sigma h(x, 0, \cdots, 0)$ have coefficients in F, by Corollary 2.3 (iii). This univariate decomposition is computed in step 2 of the algorithm. As proven in Theorem 5.2 (over K), the algorithm then computes the decomposition $f = g(bx) \circ b^{-1}h$. However, all steps of the algorithm are rational, so that indeed $g(bx)$ and $b^{-1}h$ are over F. Then also $g(cx) = g(bx) \circ \frac{c}{b}x$ and $c^{-1}h = \frac{b}{c}b^{-1}h$ are over F for any nonzero coefficient c of h.

If $\#F \leq n$, we choose two finite algebraic extension fields $L_1, L_2 \subseteq L$ of F with $\#L_1, \#L_2 > n$ and $[L_1 : F], [L_2 : F]$ two distinct primes, where L is an algebraic closure of F. We may also assume $K \subseteq L$. For $i = 1, 2$, we apply the above argument with L_i for F and the join of K and L_i for K. It follows that $g(cx)$ and $c^{-1}h$ are over $L_1 \cap L_2 = F$. □

We now want to evaluate the cost of the algorithm in three data structures for multivariate polynomials: the dense, sparse, and circuit representations (also called the straight-line representation). These representations are discussed in von zur Gathen (1985).

THEOREM 5.5. *In the tame case, multivariate polynomials can be decomposed in time polynomial in the length of the dense representation, and randomly in time polynomial in the length of a circuit representation.*

PROOF. Let $A \subseteq F$ have $n + 1$ elements. In the dense representation, one can choose $\sigma_2, \cdots, \sigma_m$ deterministically one after the other, making sure that $u(\sigma_2, \cdots, \sigma_i, x_{i+1}, \cdots, x_m)$ is nonzero, with $u \in F[x_2, \cdots, x_m]$ as in the proof of Lemma 5.3. The other steps of the algorithm can clearly be performed in a polynomial number of arithmetic operations.

For the circuit representation, we use $A \subseteq F$ with $2n$ elements, choose $(\sigma_2, \cdots, \sigma_m) \in A^{m-1}$ randomly, compute the coefficients of $f_0 = f(x_1, \sigma_2 x_1, \cdots, \sigma_m x_1)$, and test "$\deg f_0 = n$". By Section 2 and with Kaltofen's (1986) general techniques for manipulating arithmetic circuits, this and the other steps can be performed in random polynomial time.

If F has less than $2n$ elements, one has to perform the algorithm in a field extension K of F of degree at most $\log 2n$. Then scaling any decomposition by a nonzero coefficient yields a decomposition over F (Corollary 5.4 (v)). □

Moenck (1976) shows how to multiply two polynomials of degree at most d_1 and d_2, respectively, in each of m variables with $O(m(d_1+d_2+1)^m \log d)$ operations, assuming that the field supports a Fourier Transform. With this routine, the algorithm can be performed with $O(mn(n+1)^m \log n)$ operations, which is not much more than linear in the corresponding input size $(n+1)^m$. This compares favorably to the estimate of Dickerson's (1987) algorithm, which uses less than N^3 operations if the dense representation of f has N terms. (A variable-by-variable lifting would be appropriate for this input representation, where the degree in each variable is bounded.)

Unfortunately, for the sparse representation—the most intuitive one—it is conceivable that the obvious implementation of the algorithm uses more than polynomial time. This might be the case with $f, h \in F[x_1, \ldots, x_m]$ containing a small number t of nonzero monomials, $g = \sum g_i x^i \in F[x]$ such that $f = g \circ h$, and some h^i with $g_i \neq 0$ having more than polynomial in t many nonzero monomials. Another case might be an indecomposable f with few nonzero terms for which a "dense" h is computed. However, the algorithm might work reasonably well in practice also in the sparse representation, hoping that such bad examples do not occur too often (at least for decomposable f).

OPEN QUESTION 5.6. *Do such examples exist?*

6. Conclusion

We have exhibited fast polynomial decomposition algorithms for certain univariate and multivariate problems. Besides the open questions mentioned in the text (and the more difficult "wild case"), a next goal would be to elucidate the structure of (and find algorithms for) rational decompositions $f = g \circ h$ with $f, g, h \in F(x)$, and different multivariate polyhomial decompositions, such as $f = g(h_1, h_2)$ with $f, h_1, h_2 \in F[x]$.

References

Alagar, V. S., Thanh, M. (1985). Fast polynomial decomposition algorithms. In *Proc. EUROCAL 85, Lecture Notes in Computer Science* **204**, 150-153.

Barton, D. R., Zippel, R. (1985). Polynomial decomposition algorithms. *J. Symb. Comp.* **1**, 159-168.

Berlekamp, E. R. (1970). Factoring polynomials over large finite fields. *Math. Comp.* **24**, 713-735.

Borodin, A., Munro, I. (1975). *The computational complexity of algebraic and numeric problems.* American Elsevier: New York.

Brent, R. P., Kung, H. T. (1978). Fast algorithms for manipulating formal power series. *J. Assoc. Comput. Mach.* **25**, 581-595.

Cade, J. J. (1985). A new public-key cipher which allows signatures. *Proc. 2nd SIAM Conf. on Appl. Linear Algebra*, Raleigh NC.

Cantor, D. G., Kaltofen, E. (1987). Fast multiplication of polynomials over arbitrary rings. Tech. Rep. 87-35, Dept. of Computer Science, Rensselaer Polytechnic Institute, 16 pp. *Acta Inform.*, in press.

Dickerson, M. (1987). Polynomial decomposition algorithms for multivariate polynomials. Tech. Rep. 87-826, Department of Computer Science, Cornell University, Ithaca NY.

Dorey, F., Whaples, G. (1974). Prime and composite polynomials. *J. Algebra* **28**, 88-101.

Eberly, W. (1989). Very fast parallel matrix and polynomial arithmetic. *SIAM J. Comput.* **18**, 955-976.

Engstrom, H. T. (1941). Polynomial substitutions. *Amer. J. Math* **63**, 249-255.

Fried, M. D., MacRae, R. E. (1969a). On the invariance of chains of fields. *Ill. J. Math.* **13**, 165-171.

Fried, M. D., MacRae, R. E. (1969b). On curves with separated variables. *Math. Ann.* **180**, 220-226.

Fröhlich, A., Shepherdson, J. C. (1955). Effective Procedures in Field Theory. *Phil. Trans. Royal Soc., Ser. A* **248**, 407-432.

von zur Gathen, J. (1984). Parallel algorithms for algebraic problems. *SIAM J. Comput.* **13**, 802-824.

von zur Gathen, J. (1985). Irreducibility of multivariate polynomials. *J. Computer System Sciences* **31**, 225-264.

von zur Gathen, J. (1986). Parallel arithmetic computations: a survey. *Proc. 12th Int. Symp. Math. Foundations of Computer Science*, Bratislava, *Springer Lecture Notes in Computer Science* **233**, pp. 93-112.

von zur Gathen, J. (1988). Functional decomposition of polynomials: the wild case. Manuscript.

von zur Gathen, J., Kozen, D., Landau, S. (1987). Functional decomposition of polynomials. *Proc. 28th Ann. IEEE Symp. Foundations of Computer Science*, Los Angeles CA, pp. 127-131.

Gutiérrez, J., Recio, T., Ruíz de Velasco, C. (1989). Polynomial decomposition algorithm of almost quadratic complexity. *Springer Lecture Notes in Computer Science* **357**, 471-476.

Hardy, G. H., Wright, E. M. (1962). *An introduction to the theory of numbers.* Clarendon Press: Oxford.

Hasse, H. (1980). *Number Theory.* Grundlehren der math. Wiss. **229**, Springer Verlag.

Kaltofen, E. (1988). Greatest Common Divisors of Polynomials Given by Straight-line Programs. *J. Assoc. Comput. Mach.* **35**, 231-264.

Kozen, D., Landau, S. (1989). Polynomial decomposition algorithms. Tech. Rep. 86-773, Department of Computer Science, Cornell University, Ithaca NY. *J. Symb. Comp.* **7** (1989), 445-456.

Levi, H. (1942). Composite polynomials with coefficients in an arbitrary field of characteristic zero. *Amer. J. Math.* **64**, 389-400.

Lucas, E. (1877). Sur les congruences des nombres euleriens et des coefficients différentiels des fonctions trigonométriques, suivant un module premier. *Bull. Soc. Math. France* **6**, 49-54.

Moenck, R. T. (1976). Practical fast polynomial multiplication. *Proc. ACM Symp. Symbolic and Algebraic Computation*, pp. 136-148.

Ritt, J. F. (1922). Prime and composite polynomials. *Trans. Amer. Math. Soc.* **23**, 51-66.

Ritzmann, R. (1984). Ein numerischer Algorithmus zur Komposition von Potenzreihen und Komplexitätsschranken für die Nullstellenberechnung von Polynomen. Inaugural-Dissertation, Universität Zürich.

Schönhage, A. (1977). Schnelle Multiplikation von Polynomen über Körpern der Charakteristik 2. *Acta Informatica* **7**, 395-398.

Schwartz, J. T. (1980). Fast probabilistic algorithms for verification of polynomial identities. *J. Assoc. Comput. Mach.* **27**, 701-717.

Valiant, L., Skyum, S., Berkowitz, S., Rackoff, C. (1983). Fast parallel computation of polynomials using few processors. *SIAM J. Comput.* **12**, 641-644.