Polynomial Decomposition Algorithms For Multivariate Polynomials†

Matthew Dickerson
87-826

April 1987

Department of Computer Science
Cornell University
Ithaca, New York 14853-7501

†Supported by NSF grant DCR-8602663.

Polynomial Decomposition Algorithms
For
Multivariate Polynomials

Matthew Dickerson

Department of Computer Science
Cornell University
Ithaca, NY 14853

Supported by NSF grant DCR-8602663

1. Introduction:

In a recent paper [KL], Kozen and Landau give two algorithms for the nontrivial decomposition $f(x) = g(h(x))$ of a polynomial $f(x)$ over a commutative ring K. Their first algorithm requires $O(r^3s^2)$ algebraic operations where $r = \deg g$ and $s = \deg h$ and will work in any commutative ring containing a multiplicative inverse of r. Their second algorithm uses Fourier transforms to improve the bound to $O(s^2 r\log r)$ algebraic operations and will work in any commutative ring containing a multiplicative inverse of r and appropriate roots of unity.

Previous algorithms [BZ,AT], are exponential-time in the worst case, require polynomial factorization, and only work over fields of characteristic 0.

In this paper we present an algorithm for the nontrivial decomposition $f(x_1,x_2,\ldots,x_d) = g(h(x_1,x_2,\ldots x_d))$ of a given polynomial f over a commutative ring K. The algorithm requires $O(N^3)$ algebraic operations where $N = r^2s^2$ is the number of coefficients of f; $r = \deg g$ and $s = \deg h$. The algorithms are based on the method developed in [KL] and will work in any commutative ring containing a multiplicative inverse of r.

We give the complete algorithm for the case d = 2 in §2, and indicate how to extend the algorithm to general d in §3.

2. Decomposition Algorithms for the Case d = 2

In this section we give an algorithm for the decomposition of $f(x,y)$ requiring at most $O(N^{2.5}) = O(r^5s^5)$ algebraic operations, where $f(x,y) \in K(x,y)$ is monic and K is a commutative ring containing a multiplicative inverse of r.

We begin with the following lemma.

<u>Lemma 1:</u>

Let K be a commutative ring, and let

$$F_1(x,y) = \Sigma_{0\le i\le n}\Sigma_{0\le j\le n}(a_{i,j}x^iy^j) \in K(x,y)$$
$$F_2(x,y) = \Sigma_{0\le i\le n}\Sigma_{0\le j\le n}(b_{i,j}x^iy^j) \in K(x,y)$$

be monic. i.e. $a_{n,n} = b_{n,n} = 1$.

Let $G(y) \in K(y)$ be of degree m.

If $a_{c,i} = b_{c,i}$ for all $i: n-k \le i \le n$ and for some constant c, then $d_{c,i} = e_{c,i}$ for all $i: mn-k \le i \le mn$, where $d_{i,j}$ and $e_{i,j}$ are coefficients of $G(y)F_1(x,y)$ and $G(y)F_2(x,y)$ respectively. That is, $G(y)F_1(x,y) = \Sigma_{0\le i\le n}\Sigma_{0\le j\le mn}(d_{i,j}x^iy^j)$ and $G(y)F_2(x,y) = \Sigma_{0\le i\le n}\Sigma_{0\le j\le mn}(e_{i,j}x^iy^j)$.

Proof: Consider $\Sigma_{0\leq i\leq n} a_{c,i}(x^c y^i)$ and $\Sigma_{0\leq i\leq n} b_{c,i}(x^c y^i)$ to be polynomials over y with x^c as part of the coefficient. The proof follows from Lemma 3.1 of [KL].

$\square$

Before considering Lemma2, we introduce the following notation: Let $F(x,y)$ be a polynomial of degree n over x and y. (That is, the largest term is $c_{n,n}x^n y^n$) We say that the first K x L coefficients of $F(x,y)$ are the coefficients of $x^i y^j$ for all i,j: $n-K+1 \leq i \leq n$, $n-L+1 \leq j \leq n$.

Lemma 2:

Let K be a commutative ring and let $F_1(x,y)$, $F_2(x,y)$, and $G(x,y) \in K(x,y)$ be monic. If $F_1(x,y)$ and $F_2(x,y)$ agree on their first I x J coefficients, then so do $F_1(x,y)G(x,y)$ and $F_2(x,y)G(x,y)$.

Proof:

$2(J+1)^2(K+1)^2$ applications of lemma 1.

$\square$

Let

$$
\begin{aligned}
f(x,y) &= x^{rs}y^{rs} + a_{rs-1,rs}x^{rs-1}y^{rs} + a_{rs,rs-1}x^{rs}y^{rs-1} + \ldots + a_{0,0} \\
g(x) &= x^r + b_{r-1}x^{r-1} + \ldots + b_0 \\
h(x,y) &= x^s y^s + c_{s-1,s}x^{s-1}y^s + c_{s,s-1}x^s y^{s-1} + \ldots + c_{0,0}
\end{aligned}
$$

such that $r,s < rs$ and $f = g \circ h$. Let g factor in an extension of K as

$$g(x) = \Pi_{1\leq i\leq r}(x-\beta_i).$$

Then

$$f(x,y) = g(h(x,y)) = \Pi_{1\leq i\leq r}(h(x,y)-\beta_i).$$

By absorbing the constant coefficient of $h(x)$ into the β_i we can assume w.l.o.g. that $c_{0,0} = 0$; that is, if $f = g\circ h$, then $f = g_1 \circ h_1$ where

$$
\begin{aligned}
g_1(x) &= \Pi_{1\leq i\leq r}(x-(\beta_i + c_0)). \\
h_1(x,y) &= h(x,y) - c_0.
\end{aligned}
$$

The coefficients of g_1 are symmetric functions of the β_i and hence are in K.

By 4(r-1) applications of Lemma 2 we have:

(3) $h(x,y)^r$ and $f(x,y) = \Pi_{1 \leq i \leq r}(h(x,y)-\beta_i)$ agree on their first $(s+1) \times s$ coefficients and on their first $s \times (s+1)$ coefficients.

That is, they agree on all of their first $s \times s$ coefficients with the possible exception of the coefficient of the $x^{rs}y^{rs}$ term.

Let

$$q_{j,k}(x,y) = x^s y^s + c_{s-1,s}x^{s-1}y^s + \ldots + c_{s-j,s-k}x^{s-j}y^{s-k}\ , \quad 0 \leqq j \leqq s \text{ and } 0 \leqq k \leqq s.$$

Then $q_{0,0}(x,y) = x^s y^s$, $q_{s,s}(x,y) = h(x,y)$, and

$$q_{j,k}(x,y) = (q_{j-1,k-1}(x,y) + \Sigma_{0 \leq l \leq k-1}(c_{s-j,s-l}x^{s-j}y^{s-l}) + \Sigma_{0 \leq m \leq j-1}(c_{s-m,s-k}x^{s-m}y^{s-k}) + c_{s-j,s-k}x^{s-j}y^{s-k})$$

Again by lemma 3,

(4) $h(x,y)^r$ and $q_{j,k}(x,y)^r$ agree on their first $(j+1) \times (k+1)$ coefficents for $0 \leqq j \leqq s$ and $0 \leqq k \leqq s$.

By (3) and (4) above, for all $j \leqq s$ and $k \leqq s$ and not $j = k = s$, $a_{rs-j,rs-k}$ is equivalent to the coefficient of $x^{rs-j}y^{rs-k}$ in $q_{j,k}(x,y)^r$. We can therefore compute $c_{s-j,s-k}$ using the following equations, if we know $c_{s-l,s-m}$ for all $l \leqq j$ and $m \leqq k$.

$$\begin{aligned}
&q_{j,k}(x,y)^r = \\
&(q_{j-1,k-1}(x,y) + \Sigma_{0 \leq l \leq k-1}(c_{s-j,s-l}x^{s-j}y^{s-l}) + \Sigma_{0 \leq m \leq j-1}(c_{s-m,s-k}x^{s-m}y^{s-k}) + c_{s-j,s-k}x^{s-j}y^{s-k})^r \\
= \ &q_{j-1,k-1}(x,y)^r + rV_1 q_{j-1,k-1}(x,y)^{r-1} + rV_2 q_{j-1,k-1}(x,y)^{r-1} + r(r-1)V_1V_2 q_{j-1,k-1}(x,y)^{r-2} + \\
&+ rc_{s-j,s-k}x^{s-j}y^{s-k}q_{j-1,k-1}(x,y)^{r-1} + p(x,y)
\end{aligned}$$

where

$p(x,y)$ is some polynomial of degree $\leqq$ rs-2j,rs-2k

$V_1 = \Sigma_{0 \leq l \leq k-1}(c_{s-j,s-l}x^{s-j}y^{s-l})$

and $V_2 = \Sigma_{0 \leq m \leq j-1}(c_{s-m,s-k}x^{s-m}y^{s-k})$.

Let

$d_{j,k}$ be the coefficient of $x^{rs-j}y^{rs-k}$ in $q_{j-1,k-1}(x,y)^r$
$e_{j,k}$ be the coefficient of $x^{(r-1)s-j}y^{(r-1)s-k}$ in $q_{j-1,k-1}(x,y)^{r-1}$.

Then

$\Sigma_{1 \le i \le k-1}(e_{0,i}c_{s-j,s-k+i})$ is the coeffiient of $x^{rs-j}y^{rs-k}$ in $V_1 q_{j-1,k-1}(x,y)^{r-1}$
$\Sigma_{1 \le i \le j-1}(e_{i,0}c_{s-j+i,s-k})$ is the coefficient of $x^{rs-j}y^{rs-k}$ in $V_2 q_{j-1,k-1}(x,y)^{r-1}$
$c_{s,s-k}c_{s-j,s}$ is the coefficient of $x^{rs-j}y^{rs-k}$ in $V_1 V_2 q_{j-1,k-1}(x,y)^{r-2}$

and for all $0 \leqq j \leqq s$ and $0 \leqq k \leqq s$ where neither $j = k = 0$ nor $j = k = s$, we have:

(5) $$rc_{s-j,s-k} = a_{rs-j,rs-k} - d_{j,k} - r\Sigma_{1 \le i \le j-1}(e_{i,0}c_{s-j+i,s-k}) - r\Sigma_{1 \le i \le k-1}(e_{0,i}c_{s-j,s-k+i}) - r(r-1)c_{s,s-k}c_{s-j,s}.$$

Finally, note that

$$q_{j,k}(x,y)^r = (q_{j-1,k}(x,y) + \Sigma_{0 \le i \le k}c_{s-j,s-i}x^{s-j}y^{s-i})^r.$$

Let

$$Z_{1(j,k)}(x,y) = \Sigma_{0 \le i \le k}c_{s-j,s-i}x^{s-j}y^{s-i}.$$

Then

$$Z_{1(j,k)}(x,y) = Z_{1(j,k-1)}(x,y) + c_{s-j,s-k}x^{s-j}y^{s-k}.$$

and

$$Z_{1(j,k)}(x,y)^r = (Z_{1(j,k-1)}(x,y) + c_{s-j,s-k}x^{s-j}y^{s-k})^r$$
$$= \Sigma_{0 \le i \le r}(\quad)(c_{s-j,s-k}x^{s-j}y^{s-k})^i Z_{1(j,k-1)}(x,y)^{r-i}.$$

If K has a multiplicative inverse of r, this leads us to the following algorithm for computing the coefficients of h:

Algorithm 1 (Coefficients of h)

Step 0.1 Set $q_{0,0}(x,y)^i := x^{is}y^{is}$, $0 \le i \le r$
Step 0.2 Set $Z_{1(0,0)}(x,y)^i := x^{is}y^{is}$, $0 \le i \le r$

Step j,k. $0 \leqq j \leqq s$, $0 \leqq k \leqq s$, and not $j = k = s$
(Assume $q_{j-1,k-1}(x,y)^0, \ldots, q_{j-1,k-1}(x,y)^r$ and $q_{j-1,k}(x,y)^0, \ldots, q_{j-1,k}(x,y)^r$ and $Z_{1(j,k-1)}(x,y)^0, \ldots, Z_{1(j,k-1)}(x,y)^r$ are known from previous steps.)

(a) $c_{s-j,s-k} :=$

$(a_{rs-j,rs-k} - d_{j,k} - r\Sigma_{1\leq i\leq j-1}(e_{i,0}c_{s-j+i,s-k}) - r\Sigma_{1\leq i\leq k-1}(e_{0,i}c_{s-j,s-k+i}) - r(r-1)c_{s,s-k}c_{s-j,s})/r$

where $d_{j,k}$ is the coefficient of $x^{rs-j}y^{rs-k}$ in $q_{j-1,k-1}(x,y)^r$
and $e_{j,k}$ is the coefficient of $x^{(r-1)s-j}y^{(r-1)s-k}$ in $q_{j-1,k-1}(x,y)^{r-1}$

(b) calculate $c^i_{s-j,s-k}$, $0 \leqq i \leqq r$

(c) Set $Z_{1(j,k)}(x,y)^i := \Sigma_{0\leq m\leq i}(\ )(c_{s-j,s-k}x^{s-j}y^{s-k})^m Z_{1(j,k-1)}(x,y)^{i-m}$, $0 \leqq i \leqq r$

(d) Set $q_{j,k}(x,y)^i := (q_{j-1,k}(x,y) + Z_{1(j,k)}(x,y))^i$
$= \Sigma_{0\leq m\leq i}(\ )Z_{1(j,k)}(x,y)^m q_{j-1,k}(x,y)^{i-m}$, $0 \leqq i \leqq r$

This procedure takes $O(r^5s^5)$ arithmetic operations. The number of coefficients in $f(x,y)$ is $N = r^2s^2$, so we have an $O(N^{2.5})$ time algorithm to compute the coefficients of h. Once the coefficients of h are known, the coefficients of g may be obtained by solving the triangular system

(6) $$Ab = a$$

where A_{ij} is the coefficient of $x^{rs-is}y^{rs-is}$ in $h(x,y)^{r-j}$ for $0 \leqq i,j \leqq r$ and a is the vector $[a_{rs,rs}, a_{rs-s,rs-s}, \ldots, a_{0,0}]^T$. The entries of A are available from Algorithm 1. The matrix A is invertible over K, since it is triangular with all diagonal elements 1.

The coefficients of g and h are determined uniquely by the above procedure (up to the constant coefficient of h). However the system is overconstrained, and the g,h produced by the above procedure may not compose to give f. This can be checked in O(Nr) time using the powers of h computed in Algorithm 1. The running time of the algorithm is thus dominated by Algorithm 1.

3. Generalizing the Algorithm for any d.

In this section I consider the general multivariate case. Let f, g, and h be as follows:

$$\begin{aligned}
f(x_1,x_2,\ldots,x_d) &= x_1^{rs}x_2^{rs}\ldots x_d^{rs} + a_{rs-1,rs,\ldots,rs}x_1^{rs-1}x_2^{rs}\ldots x_d^{rs} + \ldots + a_{0,0,\ldots,0} \\
&= \Sigma_{0\leq z1,z2,\ldots,zd\leq rs}(a_{z1,z2,\ldots,zd}x_1^{z1}x_2^{z2}\ldots x_d^{zd}) \\
g(x) &= x^r + b_{r-1}x^{r-1} + \ldots + b_0 \\
h(x_1,x_2,\ldots,x_d) &= x_1^s x_2^s\ldots x_d^s + c_{s-1,s,\ldots,s}x_1^{s-1}x_2^s\ldots x_d^s + \ldots + c_{0,0,\ldots,0}
\end{aligned}$$

$$= \Sigma_{0 \le z1,z2,\dots,zd \le s}(c_{z1,z2,\dots,zd} x_1^{z1} x_2^{z2} \dots x_d^{zd})$$

where $f,h \in K(x_1,x_2,\dots,x_d)$ and $g \in K(x)$ and we want to decompose f into $f(x_1,x_2,\dots,x_d) = g(h(x_1,x_2,\dots,x_d))$.

I start by defining the following:

$$Z_{0(j1,j2,\dots,jd)}(x_1,x_2,\dots,x_d) = c_{s-j1,s-j2,\dots,s-jd}\, x_1^{s-j1} x_2^{s-j2} \dots x_d^{s-jd}$$
$$Z_{1(j1,j2,\dots,jd)}(x_1,x_2,\dots,x_d) = \Sigma_{0 \le z1 \le j1} (c_{s-z1,s-j2,\dots,s-jd}\, x_1^{s-z1} x_2^{s-j2} \dots x_d^{s-jd})$$
$$\dots$$
$$Z_{i(j1,j2,\dots,jd)}(x_1,x_2,\dots,x_d) = \Sigma_{0 \le z1,z2,\dots,zi \le j1}(c_{s-z1,s-z2,\dots s-zi,\dots,s-jd}\, x_1^{s-z1} x_2^{s-z2} \dots x_i^{s-zi} \dots x_d^{s-jd})$$
$$\dots$$
$$Z_{d(j1,j2,\dots,jd)}(x_1,x_2,\dots,x_d) = \Sigma_{0 \le z1,z2,\dots,zd \le j1}(c_{s-z1,s-z2,\dots s-zd}\, x_1^{s-z1} x_2^{s-z2} \dots x_d^{s-zd})$$

in other words, $Z_{0(j1,j2,\dots,jd)}$ is single term from h; $Z_{i(j1,j2,\dots,jd)}$ is an i-dimensional array of terms from h; and $Z_{d(j1,j2,\dots,jd)}$ is equivalent to the first $j_1 \times j_2 \times \dots \times j_d$ terms of h. Z_d thus replaces q in the Algorithm 1.

Now note the following equivalence:

$$(7) \qquad Z_{i(j1,j2,\dots,jd)}(x_1,x_2,\dots,x_d) = Z_{i(j1,j2,\dots ji-1,\dots,jd)}(x_1,x_2,\dots,x_d) + Z_{i-1(j1,j2,\dots,jd)}(x_1,x_2,\dots,x_d)$$

And thus at step $j_1,j_2,\dots,j_d$ of the algorithm we can compute $c_{s-j1,s-j2,\dots,s-jd}$ from $Z_{d(j1-1,j2-1,\dots,jd-1)}$ and the previously computed coefficients and we can compute all of the $Z_{i(j1,j2,\dots,jd)}{}^r$ from the $Z_{i-1}{}^r$ computed during the same step and the $Z_i{}^r$ computed during the previous steps using equation (7) above and the method in parts (c) and (d) of Algorithm 1. This gives a generalized algorithm which requires $O(r^{2d+1}s^{3d-1})$ algebraic operations. Since the number of coefficients in f is $N = r^d s^d$ this gives us an algorithm which is subcubic on the size of the input polynomial f.

4. Conclusion

We have given an algorithm for decomposing a polynomial $f(x,y) \in K(x,y)$ into $f(x,y) = g(h(x,y))$ which takes $O(N^{2.5})$ algebraic operations. (Alternatively, the algorithm takes $O(r^5s^5)$ algebraic operations where $r = \deg g$ and $s = \deg h$.) The algorithm can be generalized and a sketch of the method of generalization has been included. The generalized algorithm requires at most $O(r^{2d+1}s^{3d-1})$ algebraic operations. Since the polynomial f has $N = r^d s^d$ coefficients, this is subcubic in complexity. I have assumed that degrees of f (and therefore also of h) are the same. This assumption is not actually necessary but was made for notational convenience. All of the lemmas still hold and the algorithm still works for polynomials where the degrees are different. The main loop of the algorithm, instead of having step $z_1, z_2, \ldots, z_d$: $0 \leqq z_1, z_2, \ldots, z_d \leqq s$ would have step $z_1, z_2, \ldots, z_d$: $0 \leqq z_1 \leqq s_1$, $0 \leqq z_2 \leqq s_2$, ..., $0 \leqq z_d \leqq s_d$ where s_i is the degree of variable x_i in h and $rs_i = n_i$ where n_i is the degree of variable x_i in f.

References

[AT] Alagar, V.S. and M. Thanh. Fast Polynomial Decomposition Algorithms. *Proc. EUROCAL85*. Springer-Verlag Lecture Notes in Computer Science 204 (1985), 150-153

[BZ] Barton, D.R. and R.E. Zippel. Polynomial Decomposition Algorithms. *J. Sym. Comp.* 1 (1985), 159-168

[FM] Fried, M.D. and R.E. MacRae. On the Invariance of Chains of Fields. *Ill. J. Math.* 13 (1969)

[KL] Kozen, D. and S. Landau. Polynomial Decomposition Algorithms. *Cornell Univ. Tech. Report TR 86-773*. August 1986