Symbolic Polynomials with Sparse Exponents

Stephen M. Watt
Ontario Research Centre for Computer Algebra
Department of Computer Science, University of Western Ontario
London Ontario, CANADA N6A 5B7
watt@uwo.ca

Abstract

Earlier work has presented algorithms to factor and compute GCDs of symbolic Laurent polynomials, that is multivariate polynomials whose exponents are integer-valued polynomials. These earlier algorithms had the problem of high computational complexity in the number of exponent variables and their degree. The present paper solves this problem, presenting a method that preserves the structure of sparse exponent polynomials.

1 Introduction

We are interested in the algebra of polynomials whose exponents are not known in advance, but rather are given by integer-valued expressions, for example $x^{2m^2+n} + 3x^n y^{m^3+1} + 4$. In particular, we consider the case where the exponents are integer-valued polynomials with coefficients in $\mathbb{Q}$. One could imagine other models for integer-valued expressions, but this seems sufficiently general for a number of purposes. We call these "symbolic polynomials." Symbolic polynomials can be related to exponential polynomials [1] and to families of polynomials with parametric exponents [2, 3, 4].

To date, computer algebra systems have only been able to do simple ring operations on symbolic polynomials. They can add and multiply symbolic polynomials, but not much else. In earlier work, we have given a technical definition of symbolic polynomials, have shown that these symbolic polynomials over the integers form a UFD, and have given algorithms to compute GCDs and factor them [5, 6]. These algorithms fall into two families: *extension methods*, based on the algebraic independence of variables to different monomial powers (*e.g.* x, x^n, x^{n^2},...), and *homomorphism methods*, based on the evaluation and interpolation of exponent polynomials.

There is a problem with these earlier algorithms, however: they become impractical when the exponent polynomials are sparse. Extension methods introduce an exponential number of new variables and homomorphism methods require an exponential number of images. We have attempted to address this by performing sparse interpolation of exponents [7, 8], but this leads to impractical factorizations in the image polynomial domain.

This paper presents solves these problems. We show a substitution for the extension method that introduces only a linear number of new variables. The resulting polynomials are super-sparse and may be factored by taking images using Fermat's little theorem, as done by Giesbrecht and Roche [9]. (Indeed, Fermat's little theorem can be used in a second stage of projection for our homomorphism method, but there combining images is more complicated.)

The remainder of the paper is organized as follows: Section 2 recalls a few elementary facts about integer-valued polynomials and fixed divisors. Section 3 summarizes the extension algorithm that we have presented earlier for dense exponents. Section 4 explains why this algorithm is not suitable for the situation when the exponent polynomials are sparse and shows how to deal with this problem. Section 5 presents the extension algorithms adapted to sparse exponents and Section 6 concludes the paper.

2 Preliminaries

We recall the definitions of *integer-valued* polynomial and *fixed divisor*, and note some of their elementary properties.

Definition 1 (Integer-valued polynomial)**.** For an integral domain D with quotient field K, the (univariate) integer-valued polynomials over D, denoted $\mathrm{Int}(D)$, are defined as

$$\mathrm{Int}(D) = \{f \mid f \in K[X] \text{ and } f(a) \in D, \text{ for all } a \in D\}$$

For example, $\frac{1}{2}n^2 - \frac{1}{2}n \in \mathrm{Int}(\mathbb{Z})$ because if $n \in \mathbb{Z}$, either n or $n-1$ is even. Integer-valued polynomials have been studied for many years, with classic papers dating back 90 years [10, 11]. We make the obvious generalization to multivariate polynomials.

Definition 2 (Multivariate integer-valued polynomial)**.** For an integral domain D with quotient field K, the (multivariate) integer-valued polynomials over D in variables $X_1, \ldots, X_n$, denoted $\mathrm{Int}_{[X_1,\ldots,X_n]}(D)$, are defined as

$$\mathrm{Int}_{[X_1,\ldots,X_n]}(D) = \{f \mid f \in K[X_1, \ldots, X_n] \text{ and } f(a) \in D, \text{ for all } a \in D^n\}$$

For consistency we will use the notation $\mathrm{Int}_{[X]}(D)$ for univariate integer-valued polynomials.

When written in the binomial basis, integer-valued polynomials have the following useful property:

Property 1. *If f is a polynomial in $\mathrm{Int}_{[n_1,\ldots,n_p]}(\mathbb{Z}) \subset \mathbb{Q}[n_1, \ldots n_p]$, then when f is written in the basis $\binom{n_1}{i_1} \cdots \binom{n_p}{i_p}$, its coefficients are integers.*

If a polynomial is integer-valued, then there may be a non-trivial common divisor of all its integer evaluations.

Definition 3 (Fixed divisor)**.** A fixed divisor of an integer-valued polynomial $f \in \mathrm{Int}(D)$ is a value $q \in D$ such that $q | f(a)$ for all $a \in D$.

Given the following result, it is easy to compute the largest fixed divisor of a multivariate integer-valued polynomial.

Property 2. *If f is a polynomial in $Z[n_1, \ldots, n_p]$, then the largest fixed divisor of f may be computed as the* gcd *of the coefficients of f when written in the binomial basis.*

3 Algorithms for Dense Exponents

Following earlier work [5, 6] we define the ring of symbolic polynomials as follows:

Definition 4 (Ring of symbolic polynomials)**.** The ring of symbolic polynomials in $x_1, \ldots, x_v$ with exponents in $n_1, \ldots, n_p$ over the coefficient ring R is the ring consisting of finite sums of the form

$$\sum_i c_i x_1^{e_{i1}} x_2^{e_{i2}} \cdots x_n^{e_{in}}$$

where $c_i \in R$ and $e_{ij} \in \mathrm{Int}_{[n_1,n_2,\ldots,n_p]}(\mathbb{Z})$. Multiplication is defined by

$$c_1 x_1^{e_{11}} \cdots x_n^{e_{1n}} \quad \times \quad c_2 x_1^{e_{21}} \cdots x_n^{e_{2n}} = c_1 c_2 x_1^{e_{11}+e_{21}} \cdots x_n^{e_{1n}+e_{2n}}$$

We denote this ring $R[n_1, \ldots, n_p; x_1, \ldots, x_v]$.

A more elaborate definition is available that allows symbolic exponents on constants from the coefficient ring and everything we say here can be carried over.

We have already shown [5, 6] that symbolic polynomials with integer coefficients form a UFD. The first ingredient of the proof is that x^n and x^{n^2} are algebraically independent. The second ingredient is that fixed divisors become explicit when integer-valued polynomials are written in a binomial basis. The conversion to the binomial basis detects fixed divisors. For example, $n^2 + n$ is even for any integer n so we must detect that $x^{n^2+n} - 1$ is a difference of squares:

$$x^{n^2+n} - 1 = (x^{\frac{1}{2}n^2+\frac{1}{2}n} + 1)(x^{\frac{1}{2}n^2+\frac{1}{2}n} - 1)$$

This leads to the extension algorithms. For example, for factorization we have:

Dense Extension Algorithm for Symbolic Polynomial Factorization

INPUT: A symbolic polynomial $f \in \mathbb{Z}[n_1, ...n_p; x_1, ..., x_v]$.

OUTPUT: The factors $g_1, ..., g_n$ such that $\prod_i g_i = f$, unique up to units.

1. Put the exponent polynomials of f in the basis $\binom{n_i}{j}$.
2. Construct polynomial $F \in \mathbb{Z}[X_{10...0}, ..., X_{vd_1...d_p}]$, where d_i is the maximum degree of n_i in any exponent of f, using the correspondence
$$\gamma : x_k^{\binom{n_1}{i_1}\cdots\binom{n_p}{i_p}} \mapsto X_{ki_1...i_p}.$$
3. Compute the factors G_i of F.
4. Compute $g_i = \gamma^{-1}(G_i)$.

Under any evaluation map on the exponents, $\phi : \mathrm{Int}_{[n_1,...,n_p]}(\mathbb{Z}) \to \mathbb{Z}$, if $\phi(f)$ factors into $f_{\phi 1}, ..., f_{\phi r}$ these factors may be grouped to give the factors $\phi(g_i)$. That is, there is a partition of $\{1, ..., r\}$ into subsets I_i such that $\phi(g_i) = \prod_{j \in I_i} f_{\phi j}$. This factorization into g_i is the maximal uniform factorization in the sense that any other factorization g_i' has $\forall_i \exists_j g_i \mid g_j'$.

4 Sparse Exponents

The problem with the previous algorithm is that the change to the binomial basis makes the exponent polynomials dense. If all exponent variables are of degree d or less, the new factorization involves $v \times (d+1)^p$ indeterminates. If the number of exponent variables or their degree is large, then the problem becomes difficult. We solve this by introducing a different substitution.

In factorization and related algorithms, the reason we have transformed the exponents of the input symbolic polynomial to a binomial basis is so that all factors will have exponent polynomials with integer coefficients. Then, because $x^{cb_i} = \left(x^{b_i}\right)^c$, we can treat the algebraically independent x^{b_i} as new polynomial variables. The only thing that matters, really, is that the coefficients of the factored symbolic polynomials' exponents be integers.

We can achieve the same effect by scaling the original variables and using a Pochhammer basis for the exponent polynomials. Any polynomial in variables z_i may be written in terms of the basis $(z_i)_{(j)}$ and *vice versa*, in the same coefficient ring. The binomial coefficients and the Pochhammer symbols are related by $\binom{x}{j} = (x)_{(j)}/j!$ so multiplying the exponent polynomials by a suitable constant will make them integer-valued. To see this, we make use of the following result.

Lemma 1. *If $h \in \mathrm{Int}_{[n_1,\ldots,n_p]}(\mathbb{Z})$ is of degree at most d in each of the n_i, then $d!^p \times h \in \mathbb{Z}[n_1, ..., n_p]$.*

Proof. Because h is integer-valued, we can write

$$d!^p \times h = \sum_{0 \le i_1,\ldots,i_p \le d} h_{i_1,\ldots,i_p} d!^p \binom{n_1}{i_1} \cdots \binom{n_p}{i_p} \qquad h_{i_1,\ldots,i_p} \in \mathbb{Z}.$$

If $0 \le i \le d$, then $d!\binom{w}{i} = (d \times \cdots \times (d-i+1)) \times (w \times \cdots \times (w-i+1)) \in \mathbb{Z}[w]$ and the result is immediate. □

We now use this to avoid having to make a change of basis.

Theorem 1. *If $f \in \mathcal{P} = R[n_1, ..., n_p; x_1, ..., x_v]$ has factors $g_i \in \mathcal{P}$ with exponents in $\mathrm{Int}_{[n_1,\ldots,n_p]}(\mathbb{Z})$ with each exponent of degree at most d in any n_i, then making the substitution $x_i \mapsto X_i^{d!^p}$ gives factors in $R[n_1, ..., n_p, X_1, ..., X_v]$ with exponents in $\mathbb{Z}[n_1, ..., n_p]$.*

Proof. Let $\mathbf{exdeg}_n f$ denote the maximum degree in n of any exponent polynomial in f. By hypothesis, we have $\max_i \mathbf{exdeg}_{n_i} f = d$. Then for all g_i and n_j we have $\mathbf{exdeg}_{n_j} g_i \le \mathbf{exdeg}_{n_j} f$. Therefore all exponent polynomials occurring in any g_i are elements of $\mathrm{Int}_{[n_1,\ldots,n_p]}(\mathbb{Z})$ of degree at most d in any n_i. By Lemma 1, multiplying all the exponent polynomials by $d!^p$ will give exponent polynomials in $\mathbb{Z}[n_1, ..., n_p]$. Making the substitution $x_i \mapsto X_i^{d!^p}$ multiplies the exponent polynomials in exactly this way. □

The exponent multiplier given by the change of variables $x_i \mapsto X_i^{d!^p}$ may be larger than required to give integer coefficients in the exponents of the factors. This may lead to factors whose exponents are not integer-valued polynomials when the change of variables is inverted. It is easy to give an example of such an "over factorization" resulting from too large a multiplier. Suppose we wish to factor

$$f = x^{n^3+n^2} - x^{n^3} + x^{n^2} - 1.$$

The substitution from the theorem is $x \mapsto X^{3!}$ and this gives

$$\begin{aligned} f &= X^{6n^3+6n^2} - X^{6n^3} + X^{6n^2} - 1 \\ &= (X^{n^3})^6 (X^{n^2})^6 - (X^{n^3})^6 + (X^{n^2})^6 - 1. \end{aligned}$$

This then factors as

$$\begin{aligned} f = &\left((X^{n^2})^2 + 1\right) \times \left((X^{n^2})^4 - (X^{n^2})^2 + 1\right) \\ &\times \left(X^{n^3} - 1\right) \times \left((X^{n^3})^2 + X^{n^3} + 1\right) \times \left(X^{n^3} + 1\right) \times \left((X^{n^3})^2 - X^{n^3} + 1\right) \\ = &\left(x^{\frac{1}{3}n^2} + 1\right) \times \left(x^{\frac{2}{3}n^2} - x^{\frac{1}{3}n^2} + 1\right) \\ &\times \left(x^{\frac{1}{6}n^3} - 1\right) \times \left(x^{\frac{1}{3}n^3} + x^{\frac{1}{6}n^3} + 1\right) \times \left(x^{\frac{1}{6}n^3} + 1\right) \times \left(x^{\frac{1}{3}n^3} - x^{\frac{1}{6}n^3} + 1\right). \end{aligned}$$

The these factors do not have integer-valued polynomials as exponents. Combinations of these factors, however, do:

$$\begin{aligned} \left(x^{\frac{1}{3}n^2} + 1\right) \times \left(x^{\frac{2}{3}n^2} - x^{\frac{1}{3}n^2} + 1\right) &= x^{n^2} + 1 \\ \left(x^{\frac{1}{6}n^3} - 1\right) \times \left(x^{\frac{1}{3}n^3} + x^{\frac{1}{6}n^3} + 1\right) \times \left(x^{\frac{1}{6}n^3} + 1\right) \times \left(x^{\frac{1}{3}n^3} - x^{\frac{1}{6}n^3} + 1\right) &= x^{n^3} - 1 \end{aligned}$$

Because $\mathbb{Z}[n_1, ..., n_p; x_1, ..., x_v]$ is a UFD, there will be a grouping of factors that leads to a unique fullest factorization, up to units.

5 Algorithms for Sparse Exponents

The transformation given in Section 4 allows us to adapt the dense exponent algorithms for symbolic polynomial factorization, GCD, *etc* to sparse exponents. In each case we substitute the variables for a suitable power, compute the result, combine factors and substitute back. We show the algorithm for factorization of symbolic polynomials in more detail:

Sparse Extension Algorithm for Symbolic Polynomial Factorization

INPUT: A symbolic polynomial $f \in \mathcal{P} = \mathbb{Z}[n_1, ...n_p; x_1, ..., x_v]$.

OUTPUT: The factors $g_1, ..., g_n$ such that $\prod_i g_i = f$, unique up to units.

1. Construct $E = \rho f \in \mathbb{Z}[n_1, ..., n_p; X_1, ...X_v]$, using the substitution
$$\rho : x_i \mapsto X_i^{d!^p}.$$
2. Construct $F = \gamma E \in \mathbb{Z}[X_{10...0}, ..., X_{vd...d}]$, using the correspondence
$$\gamma : X_k^{{n_1}^{i_1} \cdots {n_p}^{i_p}} \mapsto X_{ki_1...i_p}.$$
3. Compute the factors G_j of F.
4. Compute $H_j = \gamma^{-1}(G_j)$.
5. Find the finest partition $\mathcal{H}_1 \cup \cdots \cup \mathcal{H}_N$ of $\{H_j\}$ such that for all $\mathcal{H}_i$ we have $g_i = \rho^{-1}\left(\prod_{G \in \mathcal{H}_i} G\right) \in \mathcal{P}$.

This gives the maximal uniform factorization of the symbolic polynomial f. We may compute the GCD and related quantities similarly.

We make a few general observations:

In Step 1, we need not necessarily substitute all variables with $x_i \mapsto X_i^{d!^p}$. The exponents of each x_i form independent spaces so we may calculate separate bounds b_i and substitute $x_i \mapsto X_i^{b_i}$. If any x_i has fewer than all p exponent variables or if some exponent variables have lower degrees, then the corresponding b_i will be lower.

In Step 2, if the original exponent polynomials are sparse, then most of the variables will not appear in F. In particular, the number of variables in F is at most linear in the size of the input polynomial.

The polynomial F will be supersparse: We have replaced the problem of having a number of new variables exponential in d with the problem of increasing the number of bits in the exponent coefficients by $p \log(d!)/\log 2 = O(pd \log d)$. In general, factoring super-sparse polynomials is intractable in a complexity theoretic sense as there may be dense factors of high degree. Likewise, the corresponding GCD problem can be reduced to an NP-complete problem [12]. In our problem, however, the symbolic polynomial factorization must be valid for all values of the exponent variables n_i. In particular, the symbolic polynomial polynomial factorization $\prod_i g_i$ evaluated with $\phi : \{n_1, ..., n_p\} \to 0$ will be a (possibly incomplete) factorization of the polynomial ϕf. The number of terms in the final symbolic polynomial factorization is therefore unaffected by the multiplication of the exponent polynomials by a large constant.

In Step 3, we may reduce the size of the exponents that occur in the factorization of F by taking several images using Fermat's little theorem for small primes. That is, if a variable x is going to be evaluated by a homomorphism to give an image problem, then reduce first using $x^p \equiv x \pmod p$. This has idea been observed by other authors (for example [9]).

In Step 5 we can limit the combinations that need be considered by examining only those for which the sum of asymptotically leading (and, separately, trailing) exponents give integer-valued polynomials.

6 Conclusions

We have shown how to preserve the sparsity of exponents in problems related to the factorization of of symbolic polynomials. We do this by making a change of variables that guarantees the exponents of the output polynomials will have integer coefficients.

We have implemented this method in Maple and have found it to allow factorizations of symbolic polynomials far larger than any we have been able to achieve using other methods. For the first time we appear to have an algorithm of reasonable practical complexity for computing the factorization of symbolic polynomials.

References

[1] C.W. Henson, L. Rubel and M. Singer, *Algebraic properties of the ring of general exponential polynomials*. Complex Variables Theory and Applications, **13** (1989) 1-20.

[2] V. Weispfenning, *Gröbner bases for binomials with parametric exponents*. Technical report, Universität Passau, Germany, 2004.

[3] K. Yokoyama, *On systems of algebraic equations with parametric exponents*. Proc. ISSAC 2004, July 4-7, 2004, Santander, Spain, ACM Press, 312-319.

[4] W. Pan and D. Wang. *Uniform Gröbner bases for ideals generated by polynomials with parametric exponents*. Proc. ISSAC 2006, ACM Press, 269–276.

[5] S.M. Watt, *Making computer algebra more symbolic*. Proc. Transgressive Computing 2006: A conference in honor of Jean Della Dora, April 24–26, 2006, Granada, Spain, 44–49.

[6] S.M. Watt, *Two families of algorithms for symbolic polynomials*. Computer Algebra 2006: Latest Advances in Symbolic Algorithms — Proceedings of the Waterloo Workshop I. Kotsireas, E. Zima (editors), World Scientific 2007, 193–210.

[7] M. Malenfant and S.M. Watt, *Sparse exponents in symbolic polynomials*. Symposium on Algebraic Geometry and Its Applications: In honor of the 60th birthday of Gilles Lachaud (SAGA 2007) (Abstracts), May 7–11 2007, Papeete, Tahiti.

[8] M. Malenfant, *A comparison of two families of algorithms for symbolic polynomials*. MSc.Thesis, Dept of Computer Science, University of Western Ontario, December 2007.

[9] M. Giesbrecht and D. Roche, *Interpolation of shifted-lacunary polynomials*. Proc. Mathematical Aspects of Computer and Information Sciences (MACIS), 2007.

[10] A. Ostrowski, *Über ganzwertige Polynome in algebraischen Zahlköpern*. J. Reine Angew. Math., **149** (1919), 117–124.

[11] G. Pólya, *Über ganzwertige Polynome in algebraischen Zahlköpern*. J. Reine Angew. Math., **149** (1919), 97–116.

[12] D. A. Plaisted, *New NP-hard and NP-complete polynomial and integer divisibility problems*. Theoret. Comput. Sci., **31** (1984), 125–138.