RADICALS OF ORE POLYNOMIALS

MAXIMILIAN JAROSCHEK

Abstract. We give a comprehensible algorithm to compute the radical of an Ore operator. Given an operator P, we find another operator L and a positive integer k such that $P = L^k$ and k is maximal among all integers for which such an operator L exists.

1. Introduction

The problem of factoring commutative univariate polynomials with the help of a computer is a classical and still active field of research. Frequently utilized algorithms are based on the work of Cantor-Zassenhaus [4], Berlekamp [2], Kaltofen [5] and van Hoeij [10]. An important preprocessing step for many of these algorithms is to compute the squarefree decomposition of the input polynomial, i.e. given a polynomial $p \in \mathbb{K}[x]$, find $g_1, \ldots, g_m \in \mathbb{K}[x]$ with $\gcd(g_i, g_j) = \gcd(g_i, g_i') = 1$ such that

$$p = g_1 g_2^2 \ldots g_m^m.$$

In the noncommutative setting, simplifying the process of factoring Ore polynomials by identifying repeated factors is not yet available for the existing factoring algorithms like [3, 8]. As a first step towards such a preprocessing method we present an algorithmic solution to the following problem:

Given an Ore operator P, i.e. an element of an Ore algebra $\mathbb{K}[y][X; \sigma, \delta]$, find another operator $L \in \mathbb{K}[y][X; \sigma, \delta]$ and a positive integer k such that

$$P = L^k, \tag{1}$$

and k is maximal among all integers for which such an operator L exists. In the outline of the algorithm presented here, we only consider the shift case for sake of simplicity, i.e. we take Ore operators in the algebra $\mathbb{K}[y][X; \sigma, 0]$ with trivial pseudo-derivation $\delta = 0$. Remarks on the differential case with $P \in \mathbb{K}[y][X; 1, \delta]$ and the case of general Ore algebras are given in Section 4. For details on Ore algebras see [7].

Our algorithm is based on the solution of the corresponding problem in the commutative case and on (linear) algebra methods for determining suitable values for symbolic coefficients.

2. Preliminaries

Let $\mathbb{K}$ be a computable field. We fix an operator P in the Ore algebra $\mathbb{K}[y][X; \sigma, 0]$. Operators are denoted by capital letters and the ith coefficient of an operator by the corresponding lower case letter with the index i. The order of an operator L is its degree with respect to X and is denoted by $\operatorname{ord}(L)$. By $\deg(L)$ we refer to the degree of L in y.

Author supported by Technologietransfer 2013/14 des Landes Oberösterreich.

3. The algorithm

3.1. **Degenerate Cases.** In this section we cover the cases where either $\operatorname{ord}(P) = 0$ or $\deg(P) = 0$. In both cases, we can assume $P \in \mathbb{K}[x]$, i.e. we work in a commutative domain. We compute the squarefree decomposition $g_1, \ldots, g_m$ of p (e.g. by Yun's algorithm [11]) and the greatest common divisor k of all the elements in the set $\{i \in \mathbb{N}^* \mid \deg(g_i) > 0\}$. Then it is easy to see that k is as required and L is given by

$$L = \operatorname{lc}(P)^{1/k} g_1^{1/k} g_2^{2/k} \ldots g_m^{m/k}.$$

Now we establish how to solve this problem for shift operators by solving a linear and an algebraic system of equations.

3.2. **Main Idea.** For the noncommutative case, we don't have a squarefree decomposition to simplify the task. We exploit noncommutativity to solve the problem:

Observe that for P and L as above we have that

$$PL = L^k L = L^{k+1} = LL^k = LP.$$

Therefore, L is a solution to the equation

$$PL - LP = 0. \tag{2}$$

This means that L is an element of the centralizer of P.

Definition 3.1. Let $r, d \in \mathbb{N}$. We call the set

$$\mathcal{C}_{r,d}(P) = \{L \in \mathbb{K}[y][X;\sigma,0] \mid PL = LP, \operatorname{ord}(L) \le r, \deg(L) \le d\},$$

the centralizer of P (with order r and degree d).

The centralizer of P with order r and degree d is a $\mathbb{K}$-vector space. In order to find L as in (1), we compute a basis of $\mathcal{C}_{r,d}(P)$ for certain $r, d \in \mathbb{N}$ and then construct L as a linear combination of the basis elements.

The outline of our algorithm is as follows.

Algorithm 3.1: OreRadical

Input: An operator $P \in \mathbb{K}[y][X;\sigma,0]$ with $\operatorname{ord}(P) \cdot \deg(P) \neq 0$.

Output: An operator L and an integer k such that $P = L^k$ and k is maximal among all integers for which such an operator L exists.

1. If $\operatorname{ord}(P) = 0$ or $\deg(P) = 0$: solve as in the commutative case.
2. Compute candidates for k.
3. For each candidate k', do:
4. Compute a basis for $\mathcal{C}_{\operatorname{ord}(P)/k', \deg(P)/k'}(P)$.
5. If it exists, compute an element $L_{k'}$ in $\mathcal{C}_{r,d}(P)$ for which $P = (L_{k'})^{k'}$ holds.
6. Return (L_k, k) such that k is maximal.

We already have seen in Section 3.1 how to carry out step **1**. We now show in detail how to carry out the steps **2** – **5**.

3.3. **Candidates for the Exponent.** If $P = L^k$ holds, then also $p_0 = l_0^k$ has to hold. Therefore it suffices to look for a candidate for k in the commutative case. If $g_1, \ldots, g_m$ are the factors in the squarefree decomposition of p_0, we see that the exponent k has to divide each index i for which g_i in the squarefree decomposition $g_1^1 g_2^2 \ldots g_m^m$ is not equal to 1 and it also has to divide $\operatorname{ord}(P)$ and $\deg(P)$ since we have that

$$\operatorname{ord}(P) = k \cdot \operatorname{ord}(L) \text{ and } \deg(P) = k \cdot \deg(L).$$

Therefore the set of candidates is given by the set of all common divisors of $\operatorname{ord}(P)$, $\deg(P)$ and of the indices i for which $g_i \neq 1$.

3.4. **Looping Through the Candidates.** It is clear that the real k has to be contained in the set of candidates computed in Section 3.3. The choice on the order in which the candidates are considered in the for loop of the algorithm will affect its running time. One way is to begin with the smallest $k' \neq 1$. This will guarantee that we find a nontrivial solution on the first try (if one exists), but it might be of the form $L^{k'}$ where $k' < k$, e.g. $P = L^4$ and we choose $k' = 2$. The algorithm then has to be applied recursively. On the other hand, we can start with the largest candidate and be guaranteed to find L without a recursive call, but it might be necessary to try several different candidates, e.g. when $P = L^2$ and $l_0 = n^2$ we would choose $k' = 4$. Depending on the input, one strategy may be better than the other, but in general it is not clear a priori which method is preferable.

3.5. **Computing a Basis for the Centralizer.** Having chosen a candidate k', we now have to look for $L_{k'}$ in $\mathcal{C} := \mathcal{C}_{\operatorname{ord}(P)/k', \deg(P)/k'}(P)$. We compute a basis for $\mathcal{C}$ by letting A be an operator in $\mathbb{K}[y][X; \sigma, 0]$ of order $\operatorname{ord}(P)/k'$ and degree $\deg(P)/k'$ with undetermined coefficients. Setting $AP - PA$ equal to zero then gives a linear system of equations by coefficient comparison. A basis for the solution space of this system is a basis for $\mathcal{C}$.

3.6. **Finding the Radical in the Centralizer.** A basis of the solution space of $\mathcal{C}$ corresponds to an ordered set of operators

$$(B_1, \ldots, B_\ell, 1),$$

with $B_i > B_{i+1} > 1$ for $1 \leq i < \ell$ with respect to the lexicographic term ordering for which $y < X$. Let $c_1, \ldots, c_{\ell+1}$ be undetermined and consider the equation

$$(\underbrace{c_1 B_1 + c_2 B_2 + \cdots + c_\ell B_\ell + c_{\ell+1}}_{=: L_{k'}})^{\operatorname{ord}(P)/k'} - P = 0. \tag{3}$$

Again, by coefficient comparison, this gives a system of algebraic equations. If there exists a solution, the special form of the system allows us to find it without the help of Gröbner bases. By construction, the operator on the left hand side of (3) contains a coefficient that only depends on c_1, a coefficient that only depends on c_1 and c_2 and so on. Thus we can determine the c_i's one after another. If we find a solution for all the c_i's, we can construct $L_{k'}$. If we don't find a solution, we have to consider the next candidate for k. If P cannot be written as the power of another operator, the algorithm will eventually return $L = P$ and $k = 1$.

4. Generalizations and Future Work

The algorithm can be easily adapted to work with other Ore algebras. The only time we make use of the fact that we are working in the shift algebra is in Section 3.3. For differential operators in $\mathbb{K}[y][X;1,\delta]$, the equation $p_0 = l_0^k$ does not necessarily hold, but in this case we can replace it with $\operatorname{lc}(P) = \operatorname{lc}(L)^k$, the analogous relation for the leading coefficients of P and L.

For an Ore algebra $\mathbb{K}[y][X;\sigma,\delta]$ with nontrivial σ and δ it is well known that there exists a computable isomorphism ϕ from $\mathbb{K}(y)[X;\sigma,\delta]$ to the algebra $\mathbb{K}(y)[X;\sigma,0]$ with trivial pseudo-derivation [1]. Starting with $P \in \mathbb{K}[y][X;\sigma,\delta]$, it might happen that $\phi(P)$ has rational function coefficients. The trailing coefficient of P then is of the form r^k where r is a rational function in $\mathbb{K}(y)$ and both, the numerator and the denominator of r^k can be used to find candidates for k as in Section 3.3. Once the candidates have been determined, the rest of the algorithm can be carried out with $P \in \mathbb{K}[y][X;\sigma,\delta]$ instead of $\phi(P) \in \mathbb{K}(y)[X;\sigma,0]$.

For the shift and the differential case, the algorithm was implemented in Sage [9] and will be included in the next release of the Ore algebra package [6].

The result presented here is only a first step in simplifying the task of factoring Ore polynomials. As a next goal, we would like to be able to detect if an operator is of the form $P = AL^k$ and compute its "right squarefree part" $\operatorname{rsqfp}(P) := AL$.

References

[1] S. A. Abramov, H. Q. Le, and Z. Li. Univariate Ore polynomial rings in computer algebra. *Journal of Mathematical Sciences*, 131(5):5885–5903, 2005.

[2] Elwyn R Berlekamp. Factoring polynomials over finite fields. *Bell System Technical Journal*, 46(8):1853–1859, 1967.

[3] M. Bronstein and M. Petkovšek. An introduction to pseudo-linear algebra. *Theoretical Computer Science*, 157:3–33, 1996.

[4] David G. Cantor and Hans Zassenhaus. A new algorithm for factoring polynomials over finite fields. *Mathematics of Computation*, 36(154):pp. 587–592, 1981.

[5] Erich Kaltofen and Victor Shoup. Subquadratic-time factoring of polynomials over finite fields. In *Math. Comp*, pages 398–406, 1998.

[6] M. Kauers, M. Jaroschek, and F. Johansson. *Ore algebra package for Sage*, 2013. `https://www.risc.jku.at`.

[7] Ø. Ore. Theory of non-commutative polynomials. *Annals of Mathematics*, 34(3):480–508, 1933.

[8] M. Petkovšek. Hypergeometric solutions of linear recurrences with polynomial coefficients. *Journal of Symbolic Computation*, 14(2–3):243 – 264, 1992.

[9] W. A. Stein et al. *Sage Mathematics Software (Version 6.1.1)*. The Sage Development Team, 2014. `http://www.sagemath.org`.

[10] Mark van Hoeij. Factoring polynomials and the knapsack problem. *J. Number Theory*, 95(2):167–189, 2002.

[11] David Y.Y. Yun. On square-free decomposition algorithms. In *Proceedings of the Third ACM Symposium on Symbolic and Algebraic Computation*, SYMSAC '76, pages 26–35, New York, NY, USA, 1976. ACM.

Research Institute for Symbolic Computation, Johannes Kepler University, Linz, Austria
E-mail address: `mjarosch@risc.jku.at`