A Subresultant Theory for Linear Differential, Linear Difference and Ore Polynomials, with Applications

Dissertation

zur Erlangung des akademischen Grades
"Doktor der technischen Wissenschaften"

Eingereicht von

Ziming Li, B.S., M.S.

Februar 1996

Erster Begutachter:	Univ.-Doz. Dr. Franz Winkler
Zweiter Begutachter:	o.Univ.-Prof. Dr. George E. Collins

Angefertigt am Forschungsinstitut für Symbolisches Rechnen
Technisch-Naturwissenschaftliche Fakultät
Johannes Kepler Universität Linz

Eidesstattliche Erklärung

Ich versichere, daß ich die Dissertation selbständig verfaßt habe, andere als die angegebenen Quellen und Hilfsmittel nicht verwendet und mich auch sonst keiner unerlaubten Hilfe bedient habe.

Ziming Li

Hagenberg, 10, Februar 1996

Abstract

The subresultant theory for usual commutative polynomials is generalized to linear differential, linear difference and Ore polynomials. The generalization includes the subresultant theorem, the gap structure, and the subresultant algorithm. The subresultant algorithm reduces the coefficient growth in the computation of polynomial remainder sequences without computing coefficient GCDs.

Using the subresultant theorem, we present a characterization of the compatibility of two elements in an Ore polynomial module, and determinant formulas for the greatest common right divisor and least common left multiple of two elements in an Ore polynomial ring. Furthermore, we present a modular algorithm for computing the greatest common right divisor of two Ore polynomials whose coefficients belong to the ring of univariate commutative polynomials over the integers. Experimental results illustrate that this modular algorithm is markedly superior to non-modular ones.

Zusammenfassung

Die Subresultanten-Theorie der gewöhnlichen kommutativen Polynome wird auf die linearen Differentialpolynome, Differenzpolynome, und Oreschen Polynome verallgemeinert. Diese Verallgemeinerung enthält den Subresultantensatz, die Spaltstruktur und den Subresultantenalgorithmus. Der Subresultantenalgorithmus reduziert das Wachstum der Koeffizienten in der Berechnung der Polynomrestfolgen, ohne den größten gemeinsamen Teiler der Koeffizienten zu berechnen.

Mit Hilfe des Subresultantensatzes präsentieren wir eine Charakterisierung der Berechenbarkeit von zwei Elementen in einem Oreschen Polynommodul, und entsprechende Determinanten-Formel für die größten gemeinsamen rechten Teiler und für die kleinsten gemeinsamen linken Vielfachen von zwei Elementen in einem Oreschen Ring. Außerdem präsentieren wir einen modularen Algorithmus für die Berechnung des größten gemeinsamen rechten Teilers von zwei Oreschen Polynomen, dessen Koeffizientenbereich der Ring der Polynome in einer Variablen über den ganzen Zahlen ist. Die experimentellen Ergebnisse zeigen, daß dieser modulare Algorithmus deutlich besser ist als nicht-modulare Algorithmen.

Contents

Chapter 0

Introduction

0.1 Survey of the Thesis

The purpose of this survey is to provide the reader with an outline of this thesis and a summary of the main results. Precise definitions of the terms that we use can be found in the relevant chapters.

The work in this thesis is motivated by applications of various generalizations of the Euclidean algorithm for linear operational polynomials, for example, the characteristic set method for linear differential (difference) polynomials, and algorithms for computing the greatest common right divisor and least common left multiple of two elements in an Ore polynomial ring. We present a subresultant theory for two elements in an Ore polynomial module to avoid the inefficiency of the generalizations of the Euclidean algorithm that use pseudo-division. With the help of this subresultant theory we extend the modular techniques used in the manipulation of algebraic polynomials to linear operational polynomials.

In Chapter 1, we extend Ore polynomial rings to Ore polynomial modules so that both linear homogeneous and inhomogeneous differential (difference) polynomials can be placed in one framework. We then define subresultants and establish a subresultant theory in an Ore polynomial module.

The main results of this chapter are the subresultant theorem (Theorem 1.4.2) and subresultant algorithm (Theorem 1.4.7). The subresultant theorem describes the gap structure of the subresultant sequence of two Ore polynomials. The subresultant algorithm computes the subresultant sequence of the first kind of two Ore polynomials without any GCD-calculation in the coefficient domain.

In Chapter 2, we apply this subresultant theory to three basic problems, namely, deciding the compatibility of two elements in an Ore polynomial module, computing the greatest common right divisor, and computing the least common left multiple of two elements in an Ore polynomial ring. We show that these three problems are closely related to subresultants.

The main results of Chapter 2 include two algorithms (COMP_t and COMP_b) for deciding the compatibility of two elements in an Ore polynomial module, and determinant formulas for the greatest common right divisor and least common left multiple of two elements in an Ore polynomial ring (Propositions 2.2.3 and 2.3.3).

In Chapter 3, we present a modular algorithm for computing the greatest common right divisor of two Ore polynomials over $\mathbf{Z}[t]$, where $\mathbf{Z}$ is the set of integers and t is an indeterminate. Experimental results illustrate that the modular algorithm is markedly superior to non-modular ones.

There are three algorithms, namely, GCRD_e, GCRD_p, and GCRD_m in Chapter 3. GCRD_e computes the evaluation homomorphic images of the monic associate of the greatest common right divisor of two Ore polynomials over $\mathbf{Z}_p[t]$, where p is a prime and $\mathbf{Z}_p$ is the Galois field of p elements. This algorithm hinges on the notion of subresultants. GCRD_p and GCRD_m compute the greatest common right divisor of two Ore polynomials over $\mathbf{Z}_p[t]$ and $\mathbf{Z}[t]$, respectively.

0.2 Notation and Abbreviations

Throughout the thesis, the sets of positive integers, non-negative integers, integers, and rational numbers are denoted by $\mathbf{N}^+$, $\mathbf{N}$, $\mathbf{Z}$, and $\mathbf{Q}$, respectively. We abbreviate *polynomial remainder sequence* as *PRS*, *greatest common right divisor* as *GCRD,* and *least common left multiple* as *LCLM.* All the modules in the thesis are *left* modules.

0.3 Acknowledgments

I am grateful to my thesis advisor, Franz Winkler, for his constant support, kind advice, and excellent lectures.

I thank George Collins for teaching me so much and serving on my thesis committee.

I thank Bruno Buchberger for his selfless dedication to RISC and his lectures on Thinking, Speaking, and Writing.

Much of my interest in this thesis was greatly stimulated by discussions with Hoon Hong, Peter Paule, and Jochen Pfalzgraf.

Many members of RISC helped me to complete my graduate education in one way or another. I appreciate the participants of the computer algebra and combinatorics seminars for their comments on the work in this thesis. Special thanks are due to Mark Encarnación and Josef Schicho for their friendship and help, and for what I learned from them. I also enjoyed working with István Nemes and Kazuhiro Yokoyama.

The agreeable atmosphere in Schloß Hagenberg made my four-year stay possible and fruitful. Not mentioning many other "castle-mates", I would like to thank Christopher Brown, Olga Caprotti, Roberto Pirastu, Karel Stokkermans, Volker Stahl and Emil Volcheck for their valuable help.

I thank Wentsün Wu for initiating me into the subject of computer algebra and encouraging me to choose a thesis topic connected with differential equations.

I participated in the Special Year in Computational Differential Algebra and Algebraic Geometry at the City College of New York in the spring semester, 1995. My thanks go to Francqis Boulier, Phyllis Cassidy, Raymond Hoobler, William Keigher, Sally Morrison, Michael Singer, William Sit, and Dongming Wang for their interesting lectures and insightful comments.

When writing the thesis, I received helpful references and information on particular points from Manuel Bronstein, Giuseppa Carra' Ferro, Marc Chardin, Xiaoshan Gao, and Dongming Wang.

Austrian Academic Exchange Service (ÖAD) provided me a scholarship from February, 1992 to December, 1995. *EC project PoSSo (ESPRIT III Basic Research Action, project no. 6846 and Fonds zur Förderung der wissenschaftlichen Forschung, project no. P9181-TEC)* financially supported my trips for conferences.

Chapter 1

A Subresultant Theory for Ore Polynomials

The objective of this chapter is to generalize the subresultant theory for univariate algebraic polynomials to univariate Ore polynomials. The subresultant theory for univariate algebraic polynomials was developed by Collins [8, 9] in order to avoid the high inefficiency of the Euclidean algorithm for computing PRS's. Brown and Traub [3, 4] subsequently improved Collins' results. As the calculation of PRS's is ubiquitous in solving polynomial systems, the algebraic subresultant theory is applicable to many areas such as: real root isolation [13], the computation of Sylvester's resultants [10], computation in algebraic extensions [37], cylindrical algebraic decomposition [11], computer-aided geometric design [30, 31], geometric coding theory [39], and the characteristic set method [23]. Loos [26] used Habicht's approach to present a fresh look at the subresultant theory and introduced the famous picture of the gap structure of a subresultant chain. We refer the reader to [32] for a detailed account of the subresultant theory based on Habicht's approach. Attempts to extend the subresultant theory to multivariate polynomials were also made for different purposes by Gonzàlez-Vega [17] and Mandache [27, 28].

Since various generalizations of the Euclidean algorithm are widely used in linear differential and difference algebra (see, respectively, Poole [35, § 9] and Milne-Thomson [29, § 12.2]), we naturally want to extend the algebraic subresultant theory to linear differential (difference) polynomials. Subresultants of differential operators were first defined and investigated by Chardin [5]. Chardin claimed that there existed a differential subresultant algorithm for differential operators. Proofs of

Habicht's theorem, the subresultant theorem, and the correctness of subresultant algorithm for linear differential polynomials are given by the author [24]. When proving the differential subresultant theorem, I observed that the proof had little to do with differentiation. This observation motived me to develop a general subresultant theory for both linear differential and difference polynomials. For this purpose we extend the notion of Ore polynomial rings to Ore polynomial modules so that both homogeneous and inhomogeneous linear differential (difference) polynomials can be placed in one framework. We then define subresultants and establish a subresultant theory in an Ore polynomial module. This subresultant theory will focus on describing the relations among the subresultants of two Ore polynomials and devising efficient algorithms for computing PRS's.

This chapter is organized as follows. In Section 1.1, we present some background materials and discuss our motivation in greater detail. The notion of Ore polynomial modules is defined in Section 1.2. In Section 1.3, we define the subresultants of two Ore polynomials. Section 1.4 is devoted to proving the subresultant theorem and presenting the subresultant algorithm for Ore polynomials.

1.1 Background and Motivation

Linear ordinary differential equations are equations of the form

$$a_n(t)\frac{d^n y(t)}{dt^n} + \cdots + a_1(t)\frac{dy(t)}{dt} + a_0(t)y(t) = a(t)$$

and linear ordinary difference equations are equations of the form

$$a_n(t)y(t+n) + \cdots + a_1(t)y(t+1) + a_0(t)y(t) = a(t)$$

where $y(t)$, $a(t)$, and each of the $a_i(t)$'s are functions of the variable t. If $a(t)$ is identically zero, then these two equations are said to be homogeneous.

We use algebraic language to describe the sets of linear differential equations. Let $\mathcal{R}$ be a commutative domain and D a derivation operator on $\mathcal{R}$. Then there always exists a differential polynomial ring $(\mathcal{R}\{y\}, D)$ over $\mathcal{R}$, where y is a differential indeterminate with respect to D (see Kolchin [21, p. 70]). The set of linear ordinary differential polynomials is

$$\mathcal{R}\{y\}_l = \{a_n D^n(y) + \cdots + a_1 D(y) + a_0 D^0(y) - a \mid a_n, \ldots, a_1, a_0, a \in \mathcal{R},\, n \in \mathbf{N}\}.$$

It is easy to see that $\mathcal{R}\{y\}_l$ is a D-module. The set of linear homogeneous ordinary differential polynomials is

$$\mathcal{R}\{y\}_1 = \{a_n D^n(y) + \cdots + a_1 D(y) + a_0 D^0(y) \mid a_n, \ldots, a_1, a_0 \in \mathcal{R},\, n \in \mathbf{N}\}.$$

If $A = a_n D^n(y) + \cdots + a_1 D(y) + a_0 D^0(y)$ and $B \in \mathcal{R}\{y\}_1$, then we define the product of A and B to be

$$(a_n D^n + \cdots + a_1 D + a_0 D^0)(B),$$

that is, the image of B under the linear operator $(a_n D^n + \cdots + a_1 D + a_0 D^0)$. Hence, $\mathcal{R}\{y\}_1$ can be regarded as a (non-commutative) ring. Notice that the multiplication just defined on $\mathcal{R}\{y\}_1$ is different from the multiplication on the ring $\mathcal{R}\{y\}$. Briefly, we have the following inclusions:

$$\mathcal{R}\{y\}_1 \subset \mathcal{R}\{y\}_l \subset \mathcal{R}\{y\}.$$

Both $\mathcal{R}\{y\}_1$ and $\mathcal{R}\{y\}_l$ are D-modules. In particular, $\mathcal{R}\{y\}_1$ can be viewed as a ring.

If E is an injective endomorphism of $\mathcal{R}$, then $\mathcal{R}$ and E form a difference domain (see Cohn [7, p. 57]). In the same vein, we can define a difference polynomial ring in a difference indeterminate (with respect to E), the E-module of linear difference polynomials, and the E-module of linear homogeneous difference polynomials. Similarly, the E-module of linear homogeneous difference polynomials can be viewed as a (non-commutative) ring.

A fundamental operation on differential (difference) polynomials is pseudo-division (see, respectively, Ritt [36, p. 6] and Cohn [7, p. 90]). The D-module (E-module) of linear differential (difference) polynomials is closed under differential (difference) pseudo-division. Hence, we may define pseudo-polynomial remainder sequences and design the differential (difference) Euclidean algorithm in the two modules. The Euclidean algorithm in the D-module (E-module) of linear differential (difference) polynomials is used to determine the compatibility of two elements of the D-module (E-module). The Euclidean algorithm in the ring of linear homogeneous differential (difference) polynomials is used to compute greatest common right divisors.

The differential (difference) Euclidean algorithm, which uses pseudo-division, is highly inefficient because the coefficients grow exponentially as the algorithm proceeds. If $\mathcal{R}$ is a unique factorization domain, then one may easily design the primitive differential (difference) PRS algorithm to minimize coefficient growth. Unfortunately this method requires many coefficient GCD-calculations, which may be very time-consuming.

The purpose of the subresultant theory in this chapter is to reduce coefficient growth in the Euclidean algorithm for Ore polynomials without any coefficient GCD-calculation. Note that linear differential and difference polynomials are just two special instances of Ore polynomials.

1.2 Ore Polynomial Modules

Bronstein and Petkovšek [2] observe that Ore polynomial rings (see Ore [33]) may be taken as an appropriate model for studying computational problems for linear homogeneous differential and difference polynomials. Inspired by their observation, we extend Ore polynomial rings to Ore polynomial modules so as to set up a subresultant theory for both linear homogeneous and inhomogeneous differential (difference) polynomials in one fell swoop. We will define Ore polynomial rings and Ore polynomial modules in terms of operators, because we want to introduce pseudo-division without requiring multiplication.

In the rest of this thesis, $\mathcal{R}$ is a commutative domain and X is an indeterminate over $\mathcal{R}$. The algebraic polynomial ring $\mathcal{R}[X]$ is regarded as the $\mathcal{R}$-module $\bigoplus_{n=0}^{\infty} \mathcal{R}_n$, where $\bigoplus$ stands for the direct sum of $\mathcal{R}$-modules and $\mathcal{R}_n = \mathcal{R}$, for $n \in \mathbf{N}$. The power X^n is understood as the element $(0 \ldots, 0, 1, 0, \ldots)$, whose $(n+1)$th component is 1 and other components are 0. In particular, we do *not* identify X^0 with the multiplicative identity of the domain $\mathcal{R}$. The additive identity in $\mathcal{R}[X]$ is denoted by 0. The degree of a polynomial A in $\mathcal{R}[X]$ is denoted by $\deg A$. The degree of 0 is set to be $-\infty$.

This section is organized as follows. In Section 1.2.1, we define Ore operators and Ore polynomial rings. The notion of Ore modules is defined in Section 1.2.2. Pseudo-division for two Ore polynomials is defined in Section 1.2.3.

1.2.1 Ore Operators and Ore Polynomial Rings

In this section, we present an equivalent definition of Ore polynomial rings using operators. Most of the results in this section can be found in [33, 2].

Definition 1.2.1 The mapping Θ from $\mathcal{R}[X]$ to itself is called an *Ore operator* if the following conditions are fulfilled:

1. Θ is an endomorphism of the additive group $\mathcal{R}[X]$.

2. $\Theta(X^n) = X^{n+1}$, for $n \in \mathbf{N}$.

3. $\deg \Theta(A) = \deg A + 1$, for $A \in \mathcal{R}[X]$.

4. (Multiplicative rule) There exist two mappings σ and δ from $\mathcal{R}$ to itself such that

$$\Theta(rA) = \sigma(r)\Theta(A) + \delta(r)A, \quad \text{for } r \in \mathcal{R} \text{ and } A \in \mathcal{R}[X]. \tag{1.1}$$

The next proposition describes the relation between an Ore operator Θ and the two mappings σ and δ appearing in the multiplicative rule (1.1).

Proposition 1.2.1 If Θ is an Ore operator on $\mathcal{R}[X]$ with the multiplicative rule (1.1), then

1. σ is an injective endomorphism of the ring $\mathcal{R}$;

2. δ is an endomorphism of the additive group $\mathcal{R}$;

3. for all $r, s \in \mathcal{R}$,
$$\delta(rs) = \sigma(r)\delta(s) + \delta(r)s. \tag{1.2}$$

Conversely, if σ and δ satisfy the three properties just listed, then there exists a unique Ore operator Θ with the multiplicative rule (1.1).

Proof If r and s are in $\mathcal{R}$, then (1.1) implies that

$$\Theta((r+s)X) = \sigma(r+s)X^2 + \delta(r+s)X$$

and that

$$\Theta(rX + sX) = (\sigma(r) + \sigma(s))X^2 + (\delta(r) + \delta(s))X.$$

Thus, both σ and δ are distributive with respect to addition. Setting $s = 0$ in either of the above equalities yields $\Theta(rX) = \sigma(r)X^2 + \delta(r)X$. Then σ is injective by the degree constraint on Θ, moreover, $\sigma(1) = 1$ by letting $r = 1$. It remains to show that $\sigma(rs) = \sigma(r)\sigma(s)$ and (1.2). Again, (1.1) implies that

$$\Theta((rs)X) = \sigma(rs)X^2 + \delta(rs)X \quad \text{and} \quad \Theta(r(sX)) = (\sigma(r)\sigma(s))X^2 + (\sigma(r)\delta(s) + \delta(r)s)X.$$

Comparing the respective coefficients of X^2 and X yields the desired results.

Conversely, assume that σ and δ satisfy the three conditions listed in the statement of the proposition. Then $\delta(1) = 0$ by (1.2). Define Θ to be the endomorphism of the additive group $\mathcal{R}[X]$ that sends sX^n to $\sigma(s)X^{n+1} + \delta(s)X^n$, for $s \in \mathcal{R}$ and $n \in \mathbf{N}$. Clearly, $\Theta(X^n) = X^{n+1}$, for $n \in \mathbf{N}$, and $\deg \Theta(A) = 1 + \deg A$, for $A \in \mathcal{R}[X]$. For r and s in $\mathcal{R}$, the following calculation verifies (1.1).

$$\begin{aligned}
\Theta\left(r\left(sX^n\right)\right) &= \Theta((rs)X^n) \;=\; \sigma(rs)X^{n+1} + \delta(rs)X^n \\
&= \sigma(r)\sigma(s)X^{n+1} + (\sigma(r)\delta(s) + \delta(r)s)X^n \qquad \text{(by (1.2))} \\
&= \sigma(r)\left(\sigma(s)X^{n+1} + \delta(s)X^n\right) + \delta(r)sX^n \;=\; \sigma(r)\Theta(sX^n) + \delta(r)sX^n.
\end{aligned}$$

The uniqueness of Θ is evident. $\square$

Remark 1.2.2 If σ and δ satisfy (1.2), then $\delta(1) = 0$.

If Θ is an Ore operator on $\mathcal{R}[X]$ with the multiplicative rule (1.1), then we call σ the *conjugate operator* and δ the *pseudo-derivation* (with respect to σ) associated with Θ. For $n \in \mathbf{N}$, by Θ^n we mean the n-fold composition of Θ. In particular, Θ^0 is defined to be the identity mapping. The same convention also applies to σ^n and δ^n. If

$$A = a_n X^n + a_{n-1} X^{n-1} + \cdots + a_0 X^0$$

is an Ore polynomial in $\mathcal{R}[X]$, then $A(\Theta)$ is understood as the mapping

$$a_n \Theta^n + a_{n-1} \Theta^{n-1} + \cdots + a_0 \Theta^0.$$

The next theorem enables us to introduce multiplication on the free $\mathcal{R}$-module $\mathcal{R}[X]$ via an Ore operator. The following proof is due to Bronstein and Petkovšek [2].

Theorem 1.2.2 *Let Θ be an Ore operator on $\mathcal{R}[X]$ with the conjugate operator σ and pseudo-derivation δ. For A and B in $\mathcal{R}[X]$, define the product AB of A and B to be $A(\Theta)(B)$. Then $\mathcal{R}[X]$ becomes a domain with the multiplicative identity X^0.*

Proof As Θ is distributive with respect to addition, we see that

$$A(B + C) = AB + BC \quad \text{and} \quad (B + C)A = BA + CA,$$

for $A, B, C \in \mathcal{R}[X]$. Clearly, $X^0 A = A$. Since $\Theta(X^0) = X$, $AX^0 = A$. To verify the associativity of multiplication, we claim that

$$(X^n(rX^m))\,A = X^n\,(rX^m A)\,, \qquad \text{for } n, m \in \mathbf{N},\ r \in \mathcal{R}, \text{ and } A \in \mathcal{R}[X]. \tag{1.3}$$

Proof of the Claim. The proof is done by induction on n. Equation (1.3) trivially holds for $n = 0$. Assume that it holds for $n - 1$. We compute

$$\begin{aligned}
(X^n(rX^m))\,A &= \left(X^{n-1}(\sigma(r)X^{m+1} + \delta(r)X^m)\right) A \\
&= \left(X^{n-1}\left(\sigma(r)X^{m+1}\right)\right) A + \left(X^{n-1}\left(\delta(r)X^m\right)\right) A \\
&= X^{n-1}(\sigma(r)X^{m+1}A) + X^{n-1}(\delta(r)X^m A) \quad \text{(by the induction hypothesis)} \\
&= X^{n-1}(\sigma(r)X^{m+1}A + \delta(r)X^m A) \\
&= X^{n-1}(\sigma(r)\Theta(X^m A) + \delta(r)(X^m A)) \\
&= X^{n-1}\Theta\left(r(X^m A)\right) \quad \text{(by the multiplicative rule (1.1))} \\
&= X^{n-1}(X(rX^m A)) \\
&= X^n((rX^m)A) \quad \text{(by the induction hypothesis)}.
\end{aligned}$$

This proves our claim.

Write $A = \sum_k a_k X^k$, $B = \sum_i b_i X^i$, and $C = \sum_j c_j X^j$, where A, B, and C belong to $\mathcal{R}[X]$. The following calculation verifies the associative law.

$$\begin{aligned} (BC)A &= \sum_i \sum_j \left((b_i X^i (c_j X^j))A\right) \quad \text{(by definition)} \\ &= \sum_i \sum_j b_i X^i (c_j X^j A) \quad \text{(by the claim)} \\ &= \sum_i b_i X^i \left(\sum_j c_j X^j A \right) \;=\; \sum_i (b_i X^i)(CA) \;=\; B(CA) \end{aligned}$$

□

With the multiplication defined in this theorem, we call the triple $(\mathcal{R}[X], \sigma, \delta)$ an *Ore polynomial ring*. The Ore operator Θ is omitted in this notation because Θ is uniquely determined by σ and δ. A fundamental property of the Ore polynomial ring $\mathcal{R}[X]$ is that

$$\deg AB = \deg A + \deg B, \quad \text{for all } A, B \in \mathcal{R}[X].$$

The following examples illustrate that Ore polynomial rings establish a general mathematical setting for linear (homogeneous) operational polynomials. As a matter of notation, we denote by $\mathbf{1}$ and $\mathbf{0}$ the identity and null mappings of $\mathcal{R}$, respectively.

Example 1.2.3 The Ore polynomial ring $(\mathcal{R}[X], \mathbf{1}, \mathbf{0})$ is the ring of usual commutative polynomials in X over $\mathcal{R}$.

Example 1.2.4 (Differential Operator) If D is a derivation operator on $\mathcal{R}$, then D is a pseudo-derivation with respect to $\mathbf{1}$ because $D(rs) = rD(s) + D(r)s$, for $r, s \in \mathcal{R}$. Hence, $(\mathcal{R}[X], \mathbf{1}, D)$ is the ring with the multiplication given by $X(rX^0) = rX + D(r)X^0$, for $r \in \mathcal{R}$. This ring is isomorphic to the ring of linear homogeneous differential polynomials in one differential indeterminate over $\mathcal{R}$.

Example 1.2.5 (Hilbert's Twist [22]) If E is an injective endomorphism of the domain $\mathcal{R}$ and δ is $\mathbf{0}$, then $\mathbf{0}$ is a pseudo-derivation. Hence, $(\mathcal{R}[X], E, \mathbf{0})$ is the ring with the multiplication given by $X(rX^0) = E(r)X$, for $r \in \mathcal{R}$. This ring is isomorphic to the ring of linear homogeneous difference polynomials in one difference indeterminate (with respect to E) over $\mathcal{R}$.

Example 1.2.6 Let K be a field and $\mathcal{R}$ the usual commutative polynomial ring $K[t]$. For a non-zero $h \in K$, we define E_h and Δ_h by

$$E_h(f(t)) = f(t+h) \quad \text{and} \quad \Delta_h(f(t)) = \frac{f(t+h) - f(t)}{h}, \quad \text{for all } f \in \mathcal{R}.$$

An easy calculation shows that $\Delta_h(fg) = E_h(f)\Delta_h(g) + \Delta_h(f)g$, for $f, g \in \mathcal{R}$. Thus, $(\mathcal{R}[X], E_h, \Delta_h)$ is the ring with the multiplication given by $X(rX^0) = E_h(r)X + \Delta_h(r)X^0$, for all $r \in \mathcal{R}$.

Example 1.2.7 (q-Differential Operator [34]) Let K be a field and $\mathcal{R}$ the formal power series ring $K[[t]]$. For $q \in K$ with $q \neq 0, 1$, we define two operators E_q and Δ_q by

$$E_q(f(t)) = f(qt) \quad \text{and} \quad \Delta_q(f(t)) = \frac{f(qt) - f(t)}{qt - t}, \quad \text{for all } f(t) \in \mathcal{R}.$$

It is easy to verify that $\Delta_q(fg) = E_q(f)\Delta_q(g) + \Delta_q(f)g$, for all $f, g \in \mathcal{R}$. Hence, $(\mathcal{R}[X], E_q, \Delta_q)$ is the ring with the multiplication given by $X(fX^0) = E_q(f)X + \Delta_q(f)X^0$, for all $f \in \mathcal{R}$.

We refer the interested reader to Chyzak [6] for more examples of Ore polynomial rings.

1.2.2 Ore Polynomial Modules

To establish a single subresultant theory for both homogeneous and inhomogeneous linear differential (difference) polynomials, we use the $\mathcal{R}$-module $\mathcal{R}[X] \oplus \mathcal{R}$. We define the *degree* of an element $A \oplus a$ of $\mathcal{R}[X] \oplus \mathcal{R}$ to be the degree of A if A is nonzero, the degree of $0 \oplus a$ to be -1 if a is nonzero, and the degree of $0 \oplus 0$ to be $-\infty$. The degree of $A \oplus a$ is denoted by $\deg(A \oplus a)$. The additive identity $0 \oplus 0$ of the module $\mathcal{R}[X] \oplus \mathcal{R}$ is denoted by 0.

Definition 1.2.8 An endomorphism Θ of the additive group $\mathcal{R}[X] \oplus \mathcal{R}$ is said to be an *Ore operator* on $\mathcal{R}[X] \oplus \mathcal{R}$ if the following hold:

1. Θ restricted to $\mathcal{R}[X]$ is an Ore operator on $\mathcal{R}[X]$, with the conjugate operator σ and pseudo-derivation δ.
2. For every $r \in \mathcal{R}$, $\Theta(0 \oplus r) \in 0 \oplus \mathcal{R}$.
3. (Multiplicative rule) For every $r \in \mathcal{R}$ and $A \oplus a \in \mathcal{R}[X] \oplus \mathcal{R}$,

$$\Theta(r(A \oplus a)) = \sigma(r)\Theta(A \oplus a) + \delta(r)(A \oplus a). \tag{1.4}$$

The quadruple $(\mathcal{R}[X] \oplus \mathcal{R}, \Theta, \sigma, \delta)$ is called an *Ore polynomial module* whose elements are called *Ore polynomials*.

For an Ore polynomial ring $(\mathcal{R}[X], \sigma, \delta)$, there is a unique Ore operator Θ on $\mathcal{R}[X]$ such that equation (1.1) holds. We can extend Θ to $\mathcal{R}[X] \oplus \mathcal{R}$ by the next proposition.

Proposition 1.2.3 If Θ is an Ore operator on $\mathcal{R}[X]$, with the conjugate operator σ and pseudo-derivation δ, then the mapping

$$\begin{array}{rccc} \Theta_1: & \mathcal{R}[X]\oplus\mathcal{R} & \longrightarrow & \mathcal{R}[X]\oplus\mathcal{R} \\ & A\oplus a & \mapsto & \Theta(A)\oplus\delta(a) \end{array}$$

is an Ore operator on $\mathcal{R}[X]\oplus\mathcal{R}$. If, moreover, δ is $\mathbf{0}$, then the mapping

$$\begin{array}{rccc} \Theta_2: & \mathcal{R}[X]\oplus\mathcal{R} & \longrightarrow & \mathcal{R}[X]\oplus\mathcal{R} \\ & A\oplus a & \mapsto & \Theta(A)\oplus\sigma(a), \end{array}$$

is an Ore operator on $\mathcal{R}[X]\oplus\mathcal{R}$.

Proof It suffices to verify that both Θ_1 and Θ_2 are subject to the respective multiplicative rules. If $A\oplus a$ is in $\mathcal{R}[X]\oplus\mathcal{R}$ and r in $\mathcal{R}$, then

$$\begin{aligned} \Theta_1(r(A\oplus a)) &= \Theta_1((rA)\oplus(ra)) = \Theta(rA)\oplus\delta(ra) = \\ &= (\sigma(r)\Theta(A)+\delta(a)A)\oplus\delta(ra) \quad \text{(by (1.1))} \\ &= (\sigma(r)\Theta(A)+\delta(r)A)\oplus(\sigma(r)\delta(a)+\delta(r)a) \quad \text{(by (1.2))} \\ &= \sigma(r)\Theta(A)\oplus\sigma(r)\delta(a)+\delta(r)A\oplus\delta(r)a \\ &= \sigma(r)\left(\Theta(A)\oplus\delta(a)\right)+\delta(r)(A\oplus a) \\ &= \sigma(r)\Theta_1(A\oplus a)+\delta(r)(A\oplus a). \end{aligned}$$

This proves the first assertion. If δ is the null mapping, then

$$\Theta_2(r(A\oplus a)) = \Theta_2((rA)\oplus(ra)) = \Theta(rA)\oplus\sigma(ra) = \sigma(r)\Theta(A)\oplus\sigma(r)\sigma(a) = \sigma(r)\Theta_2(A\oplus a). \quad \square$$

Example 1.2.9 Let D be a differential operator on $\mathcal{R}$. Define the Ore operator Θ on $\mathcal{R}[X]\oplus\mathcal{R}$ to be such that $\Theta(rX^n) = rX^{n+1}+D(r)X^n$ and $\Theta(0\oplus r) = 0\oplus D(r)$, for all $r\in\mathcal{R}$ and $n\in\mathbf{N}$. Let Y be a differential indeterminate (with respect to D) over $\mathcal{R}$. Define the mapping

$$\begin{array}{rccc} \phi: & \mathcal{R}\{Y\}_l & \longrightarrow & \mathcal{R}[X]\oplus\mathcal{R} \\ & \left(\sum_{i=0}^n a_i(D^iY)\right)+a & \mapsto & \left(\sum_{i=0}^n a_iX^i\right)\oplus a. \end{array}$$

Then ϕ is an $\mathcal{R}$-module isomorphism such that the diagram below commutes.

$$\begin{array}{ccc} \mathcal{R}\{Y\}_l & \xrightarrow{\phi} & \mathcal{R}[X]\oplus\mathcal{R} \\ \downarrow D & & \downarrow \Theta \\ \mathcal{R}\{Y\}_l & \xrightarrow{\phi} & \mathcal{R}[X]\oplus\mathcal{R} \end{array}$$

Example 1.2.10 Let E be a shift operator on $\mathcal{R}$. Define the Ore operator Θ on $\mathcal{R}[X] \oplus \mathcal{R}$ to be such that $\Theta(rX^n) = E(r)X^{n+1}$ and $\Theta(0 \oplus r) = 0 \oplus E(r)$, for all $r \in \mathcal{R}$ and $n \in \mathbf{N}$. Let Y be a difference indeterminate (with respect to E) over $\mathcal{R}$. Denote by $\mathcal{R}\{Y\}_l$ the E-module of linear difference polynomials in Y over $\mathcal{R}$. Define the mapping

$$\begin{array}{cccc} \phi: & \mathcal{R}\{Y\}_l & \longrightarrow & \mathcal{R}[X] \oplus \mathcal{R} \\ & \left(\sum_{i=0}^{n} a_i(E^iY)\right) + a & \mapsto & \left(\sum_{i=0}^{n} a_iX^i\right) \oplus a. \end{array}$$

Then ϕ is an $\mathcal{R}$-module isomorphism such that the following diagram commutes.

$$\begin{array}{ccc} \mathcal{R}\{Y\}_l & \xrightarrow{\phi} & \mathcal{R}[X] \oplus \mathcal{R} \\ \downarrow E & & \downarrow \Theta \\ \mathcal{R}\{Y\}_l & \xrightarrow{\phi} & \mathcal{R}[X] \oplus \mathcal{R} \end{array}$$

Example 1.2.11 Let $\mathcal{R}$, E_q, and Δ_q be the same as in Example 1.2.7. Define the Ore operator Θ on $\mathcal{R}[X] \oplus \mathcal{R}$ to be such that $\Theta(rX^n) = E_q(r)X^{n+1} + \Delta_q(r)X^n$ and $\Theta(0 \oplus r) = 0 \oplus \Delta_q(r)$, for all $r \in \mathcal{R}$ and $n \in \mathbf{N}$. Then the Ore polynomial module $(\mathcal{R}[X] \oplus \mathcal{R}, \Theta, E_q, \Delta_q)$ can be regarded as the $\mathcal{R}$-module of linear q-differential polynomials in a Δ_q-indeterminate.

Notation *In the remainder of this chapter,* $(\mathcal{R}[X] \oplus \mathcal{R}, \Theta, \sigma, \delta)$ *is assumed to be an Ore polynomial module and simply denoted by* $\mathcal{R}[X] \oplus \mathcal{R}$*. When there is no ambiguity, we denote* $\Theta(A)$*,* $\sigma(r)$*, and* $\delta(r)$*, respectively, by* ΘA*,* σr*, and* δr*. By* $\delta\sigma$ *it is understood as the composition of* σ *and* δ*. The same also applies to* $\sigma\delta$*.*

The next lemma can be regarded as an extension of the Leibniz rule in calculus.

Lemma 1.2.4 For r in $\mathcal{R}$, A in $\mathcal{R}[X] \oplus \mathcal{R}$, and n in $\mathbf{N}^+$, $(\Theta^n(rA) - (\sigma^n r)\Theta^n A)$ is an $\mathcal{R}$-linear combination of $\Theta^{n-1}A, \ldots, \Theta A, A$.

Proof If $n = 1$, then $\Theta(rA) - (\sigma r)\Theta A = (\delta r)A$ by the multiplicative rule (1.4). Suppose that the lemma holds for $n - 1$. Then

$$\Theta^{n-1}(rA) - (\sigma^{n-1}r)\Theta^{n-1}A = \sum_{i=0}^{n-2} r_i\Theta^i A,$$

where r_i belongs to $\mathcal{R}$, for $i = 0, 1, \ldots, n-2$. Applying Θ to both sides of the above equality yields the lemma. □

Lemma 1.2.4 is referred as the *extended Leibniz rule* and will be frequently used in the sequel.

If a submodule $\mathcal{M}$ of $\mathcal{R}[X] \oplus \mathcal{R}$ has the property that $\Theta(\mathcal{M}) \subset \mathcal{M}$, then $\mathcal{M}$ is called a Θ-*submodule*. If $\mathcal{N}$ is a subset of $\mathcal{R}[X]\oplus\mathcal{R}$, then the multiplicative rule (1.4) implies that the smallest Θ-submodule containing $\mathcal{N}$ is the submodule $[\mathcal{N}]$ generated by all the elements of $\Theta(\mathcal{N})$, which we call the Θ-submodule generated by $\mathcal{N}$. Two elements A and B of $\mathcal{R}[X] \oplus \mathcal{R}$ are said to be Θ-*compatible* if the Θ-submodule $[A, B]$ generated by A and B does not contain any element of degree -1. Clearly, Θ on $\mathcal{R}[X] \oplus \mathcal{R}$ can be regarded as an Ore operator on $\mathcal{R}[X]$ via the canonical projection from $\mathcal{R}[X]\oplus\mathcal{R}$ to $\mathcal{R}[X]$. Thus, we also call Θ the Ore operator on $\mathcal{R}[X]$. The Θ-submodule $\mathcal{R}[X]\oplus 0$ is simply denoted by $\mathcal{R}[X]$. The notion of Θ-submodules is a general setting for linear differential and difference submodules, and left ideals of an Ore polynomial ring.

1.2.3 Pseudo-Remainders and Polynomial Remainder Sequences

In this section, we define pseudo-division and polynomial remainder sequences. To simplify the notation that will be used later, we extend the following factorial notation [29, p. 25].

Definition 1.2.12 For n in $\mathbf{N}^+$ and r in $\mathcal{R}$, the nth *σ-factorial of* r is defined to be the product

$$\prod_{i=0}^{n-1} \sigma^i r,$$

which is denoted by $r^{[n]}$. In addition, $r^{[0]}$ is set to be 1.

Lemma 1.2.5 If $r, s \in \mathcal{R}$, and $m, n \in \mathbf{N}$, then

1. $(rs)^{[m]} = r^{[m]} s^{[m]}$,
2. $r^{[m+n]} = r^{[m]} (\sigma^m r)^{[n]}$,
3. $(r^{[m]})^{[n]} = (r^{[n]})^{[m]}$,
4. $r^{[m+1][n+1]} = r^{[m+n+1]} (\sigma r)^{[m][n]}$.

Proof The first and second assertions are immediate from Definition 1.2.12. The third assertion is proved by the following calculation:

$$(r^{[m]})^{[n]} = \prod_{j=0}^{n-1} \sigma^j \left(\prod_{i=0}^{m-1} \sigma^i r \right) = \prod_{j=0}^{n-1} \prod_{i=0}^{m-1} \sigma^{i+j} r = \prod_{i=0}^{m-1} \sigma^i \left(\prod_{j=0}^{n-1} \sigma^j r \right) = (r^{[n]})^{[m]}.$$

We calculate

$$r^{[m+1][n+1]} = \prod_{i=0}^{n} \sigma^i \left(r^{[m+1]} \right) = \prod_{i=0}^{n} \sigma^i \left(r (\sigma r)^{[m]} \right) = r^{[n+1]} (\sigma r)^{[n+1][m]} = r^{[n+1]} (\sigma r)^{[n][m]} (\sigma^{n+1} r)^{[m]}.$$

The last assertion is then proved by the equality $r^{[m+n+1]} = r^{[n+1]}(\sigma^{n+1}r)^{[m]}$. $\square$

If $P \oplus p$ belongs to $\mathcal{R}[X] \oplus \mathcal{R}$ and P is nonzero, then the leading coefficient of P is also called the leading coefficient of $P \oplus p$, and denoted by $\mathrm{lc}(P \oplus p)$.

Definition 1.2.13 Let A and B be in $\mathcal{R}[X] \oplus \mathcal{R}$, with respective degrees m and n, where $n \geq 0$. A *pseudo-remainder* of A and B is defined to be either A, if $m < n$; or $C \in \mathcal{R}[X] \oplus \mathcal{R}$ such that $\deg C < \deg B$ and

$$\left(\prod_{i=0}^{m-n} \mathrm{lc}(\Theta^i B) \right) A = \sum_{i=0}^{m-n} r_i \Theta^i B + C, \tag{1.5}$$

where r_i belongs to $\mathcal{R}$, for $i = 0, 1, \ldots, m-n$.

The pseudo-remainder, as defined in equation (1.5), can be computed by a process analogous to the algebraic pseudo-division. As $\deg(\Theta^{i+1}B) = \deg(\Theta^i B) + 1$, for all $i \in \mathbf{N}$, the pseudo-remainder of A and B is unique. We denote the pseudo-remainder of A and B by $\mathrm{prem}(A, B)$.

Lemma 1.2.6 If B is a non-zero polynomial in $\mathcal{R}[X] \oplus \mathcal{R}$, then $\mathrm{lc}(\Theta^m B) = \sigma^m \mathrm{lc}(B)$, for $m \in \mathbf{N}^+$.

Proof If $B = (b_n X^n + \cdots + b_1 X + b_0) \oplus b$, then

$$\Theta B = (\sigma b_n) X^{n+1} + \text{ terms of degree lower than } (n+1)$$

by the multiplicative rule (1.4), so $\mathrm{lc}(\Theta B) = \sigma \mathrm{lc}(B)$. The lemma then follows by induction on m. $\square$

Corollary 1.2.7 If A and B are the same as in Definition 1.2.13, then equation (1.5) can be rewritten as

$$\mathrm{lc}(B)^{[m-n+1]} A = \sum_{i=0}^{m-n} r_i \Theta^i B + \mathrm{prem}(A, B). \tag{1.6}$$

Proof It is immediate from (1.5) and Lemma 1.2.6. $\square$

We call (1.6) the *pseudo-remainder formula*. If A and B are in $\mathcal{R}[X]$, then (1.6) can be written as:

$$\mathrm{lc}(B)^{[m-n+1]} A = QB + \mathrm{prem}(A, B),$$

where Q is in $\mathcal{R}[X]$, since $\mathcal{R}[X]$ is a ring. We call Q the *left pseudo-quotient* of A and B.

Example 1.2.14 Let $(\mathcal{R}[X] \oplus \mathcal{R}, \Theta, \mathbf{1}, D)$ be the same as in Example 1.2.9. Then equation (1.6) gives us the pseudo-remainder formula for two linear differential polynomials, that is,

$$\mathrm{lc}(B)^{m-n+1} A = \sum_{i=0}^{m-n} r_i \Theta^i B + \mathrm{prem}(A, B).$$

Let $(\mathcal{R}[X] \oplus \mathcal{R}, \Theta, E, \mathbf{0})$ be the same as in Example 1.2.10. Then equation (1.6) specializes to the pseudo-remainder formula for two linear difference polynomials, that is,

$$\mathrm{lc}(B)^{[m-n+1]} A = \sum_{i=0}^{m-n} r_i \Theta^i B + \mathrm{prem}(A, B).$$

Similarly, we can obtain the pseudo-remainder formulas for linear Δ_h- and Δ_q-polynomials.

For A and B in $\mathcal{R}[X] \oplus \mathcal{R}$, A and B are *similar* over $\mathcal{R}$ $(A \sim_{\mathcal{R}} B)$ if there exist non-zero r and s in $\mathcal{R}$ such that $rA = sB$. For $A_1, A_2 \in \mathcal{R}[X] \oplus \mathcal{R}$ with $\deg(A_1) \geq \deg(A_2) \geq 0$, let

$$A_1, A_2, \ldots, A_k \tag{1.7}$$

be a sequence of non-zero elements of $\mathcal{R}[X] \oplus \mathcal{R}$ such that $A_i \sim_{\mathcal{R}} \mathrm{prem}(A_{i-2}, A_{i-1})$, for $i = 3, \ldots\ k$, and either $\deg(A_k) < 0$ or $\mathrm{prem}(A_{k-1}, A_k) = 0$. Such a sequence is called a *PRS* of A_1 and A_2. If $A_i = \mathrm{prem}(A_{i-2}, A_{i-1})$, for $i = 3, \ldots, k$, then the sequence (1.7) is said to be *Euclidean*. If $\mathcal{R}$ is a unique factorization domain and each of the A_i's $(i > 2)$ given in (1.7) is primitive, then this sequence is said to be *primitive*. From the definition, it follows that there exist non-zero r_i and s_i in $\mathcal{R}$ such that $r_i A_{i-2} - s_i A_i \in [A_{i-1}]$, for $i = 3, \ldots, k$. Just as in the algebraic case, A_1 and A_2 are Θ-compatible if and only if $\deg(A_k) \geq 0$.

1.3 Subresultants of Two Ore Polynomials

In this section, we define the subresultants of two Ore polynomials. Algebraic and differential subresultants are two special instances of our general definition. We review determinant polynomials (see Loos [26] and Mishra [32]) in Section 1.3.1. The definition of subresultants is given in Section 1.3.2. Section 1.3.3 is devoted to presenting the row-reduction formula for subresultants. This formula is used to prove the subresultant theorem in Section 1.4.

Throughout the remainder of this chapter, an Ore polynomial A with degree n is written as

$$A = a_n X^n + \cdots + a_0 X^0 + a_{-1} X^{-1}.$$

1.3.1 Determinant Polynomials

Definition 1.3.1 Let M be an $r \times c$ matrix with entries in $\mathcal{R}$. If $r \leq c$, then the *determinant polynomial* of M is defined to be

$$\mid M \mid = \sum_{i=-1}^{c-r-1} \det(M_i) X^i,$$

where M_i is the $r \times r$ matrix whose first $(r-1)$ columns are the first $(r-1)$ columns of M and whose last column is the $(c-i-1)$th column of M, for $i = -1, 0, \ldots, c-r-1$.

The polynomial $\mid M \mid$ just defined is nothing but DetPol(M) (see [32, p. 241]) divided by X.

Let

$$\mathcal{A} : A_1, A_2, \ldots, A_m \tag{1.8}$$

be a sequence in $\mathcal{R}[X] \oplus \mathcal{R}$. We denote by $\deg \mathcal{A}$ the maximum of the degrees of the members in $\mathcal{A}$. Let $\deg \mathcal{A} = n > -1$ and write A_i as

$$A_i = \sum_{j=-1}^{n} a_{ij} X^j, \quad (1 \leq i \leq m) \tag{1.9}$$

where each of the a_{ij}'s belongs to $\mathcal{R}$. The *matrix associated with* $\mathcal{A}$ is defined to be the $m \times (n+2)$ matrix whose entry in the ith row and jth column is the coefficient of X^{n+1-j} in A_i, for $i = 1, \ldots, m$, and $j = 1, \ldots, n+2$. In other words, the matrix associated with $\mathcal{A}$ is

$$\begin{pmatrix} a_{1n} & a_{1,n-1} & \cdots & a_{10} & a_{1,-1} \\ a_{2n} & a_{2,n-1} & \cdots & a_{20} & a_{2,-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{mn} & a_{m,n-1} & \cdots & a_{m0} & a_{m,-1} \end{pmatrix}.$$

This matrix is denoted by $\mathrm{mat}(A_1, A_2, \ldots A_m)$ or $\mathrm{mat}(\mathcal{A})$.

Definition 1.3.2 The sequence $\mathcal{A}$ given in (1.8) is said to be *determinantal* if $m \leq n+2$. If $\mathcal{A}$ is determinantal, then the *determinant polynomial* of $\mathcal{A}$ is defined to be $\mid \mathrm{mat}(\mathcal{A}) \mid$. The determinant polynomial of $\mathcal{A}$ is denoted by $\mid \mathcal{A} \mid$.

Convention *In the rest of this section, the sequence $\mathcal{A}$ given in (1.8) is always a determinantal sequence of degree n.*

Remark 1.3.3 By the determinant of the $(m \times m)$ matrix

$$N = \begin{pmatrix} a_{1n} & a_{1,n-1} & \cdots & a_{1,n-m+2} & A_1 \\ a_{2n} & a_{2,n-1} & \cdots & a_{2,n-m+2} & A_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m-1,n} & a_{m-1,n-1} & \cdots & a_{m-1,n-m+2} & A_{m-1} \\ a_{mn} & a_{m,n-1} & \cdots & a_{m,n-m+2} & A_m \end{pmatrix},$$

we mean the sum

$$\sum_{k=1}^{m}(-1)^{k+m}\det(N_k)A_k,$$

where N_k is the $(m-1)\times(m-1)$ submatrix obtained from deleting the kth row and the last column of N. One sees that this definition is just the expansion of $\det(M)$ by its last column. However, our remark is necessary because the a_{kj}'s are in $\mathcal{R}$ while the A_k's are in $\mathcal{R}[X]\oplus\mathcal{R}$.

The following lemmas provide some useful properties of determinant polynomials.

Lemma 1.3.1 With the notation used in Remark 1.3.3, we have $\mid \mathcal{A} \mid = \det(N)$. In particular, $\mid \mathcal{A} \mid$ is an $\mathcal{R}$-linear combination of the members of $\mathcal{A}$.

Proof It is immediate from Remark 1.3.3 and the formula for expanding a determinant by a column. □

Lemma 1.3.2 The determinant polynomial of a matrix is a multilinear alternating function of rows.

Proof See [32, pp. 242–243]. □

Lemma 1.3.3 Let r be a non-zero element of $\mathcal{R}$, A an element of $\mathcal{R}[X]\oplus\mathcal{R}$, and k a non-negative integer. If

$$H = \mid \ldots, \Theta^k(rA),\ \Theta^{k-1}(rA), \ldots, \Theta(rA),\ rA, \ldots \mid,$$

then

$$H = r^{[k+1]} \mid \ldots, \Theta^k A, \Theta^{k-1} A, \ldots, \Theta A,\ A, \ldots \mid .$$

Proof We proceed by induction on k. The lemma is trivial when $k = 0$. Assume that $k > 0$ and that the lemma is true for $k-1$. Then

$$H = r^{[k]} \mid \ldots, \Theta^k(rA), \Theta^{k-1}A, \ldots, \Theta A, A, \ldots \mid .$$

It follows from the extended Leibniz rule (Lemma 1.2.4) that the polynomial $(\sigma^k r)\Theta^k A - \Theta^k(rA)$ is an $\mathcal{R}$-linear combination of $\Theta^{k-1}A$, $\Theta^{k-2}A$, …, ΘA, A. Thus, we may replace $\Theta^k(rA)$ in the above determinant polynomial by $(\sigma^k r)(\Theta^k A)$, according to Lemma 1.3.2. □

At last, we extend the techniques for expanding triangular determinants to determinant polynomials.

Lemma 1.3.4 Let

$$\mathrm{mat}(\mathcal{A}) = \begin{pmatrix} a_{1n} & a_{1,n-1} & \cdots & \cdots & \cdots & \cdots & a_{10} & a_{1,-1} \\ 0 & a_{2,n-1} & \cdots & \cdots & \cdots & \cdots & a_{20} & a_{2,-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & a_{k-1,n+2-k} & a_{k-1,n+1-k} & \cdots & a_{k-1,0} & a_{k-1,-1} \\ 0 & \cdots & 0 & 0 & a_{k,n+1-k} & \cdots & a_{k0} & a_{k,-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 0 & a_{m-1,n+1-k} & \cdots & a_{m-1,0} & a_{m-1,-1} \\ 0 & \cdots & 0 & 0 & a_{m,n+1-k} & \cdots & a_{m0} & a_{m,-1} \end{pmatrix}.$$

1. If $k < m$, then

$$|\mathcal{A}| = \begin{cases} \left(\prod_{i=1}^{k-1} \mathrm{lc}(A_i)\right) | A_k, \ldots, A_m | & \text{if } \deg A_i = n+1-i, \text{ for all } i \text{ with } 2 \le i \le k-1, \text{ and} \\ & \deg A_j = n+1-k, \text{ for some } j \text{ with } k \le j \le m, \\ 0 & \text{otherwise.} \end{cases}$$

2. If $k = m$, then

$$|\mathcal{A}| = \begin{cases} \left(\prod_{i=1}^{m-1} \mathrm{lc}(A_i)\right) A_m & \text{if } \deg A_i = n+1-i, \text{ for all } i \text{ with } 2 \le i \le m-1, \\ 0 & \text{otherwise.} \end{cases}$$

Proof By Definition 1.3.1 and Remark 1.3.3, we have

$$|\mathcal{A}| = \left(\prod_{i=1}^{k-1} a_{i,n+1-i}\right) \det \begin{pmatrix} a_{k,n+1-k} & \cdots & a_{k,n-m+2} & A_k \\ \cdots & \cdots & \cdots & \cdots \\ a_{m-1,n+1-k} & \cdots & a_{m-1,n-m+2} & A_{m-1} \\ a_{m,n+1-k} & \cdots & a_{m,n-m+2} & A_m \end{pmatrix}. \tag{1.10}$$

Let $k < m$. If there is an integer i such that $2 \le i \le k-1$ and $\deg A_i < n+1-i$, then $a_{i,n+1-i} = 0$, so $|\mathcal{A}| = 0$ by (1.10). If $\deg A_j < n+1-k$, for all j such that $k \le j \le m$, then the determinant in the right-hand side of (1.10) is zero, and so is $|\,\mathcal{A}\,|$. If $a_{i,n+1-i} \neq 0$, for all i such that $2 \leq i \leq k-1$, and $a_{j,n+1-j} \neq 0$, for some j such that $k \leq j \leq m$, then (1.10) becomes

$$|\,\mathcal{A}\,| = \left(\prod_{i=1}^{k-1} \mathrm{lc}(A_i)\right) |\, A_k, A_{k+1}, \ldots, A_m \,| \, .$$

If $k = m$, then equation (1.10) becomes $|\,\mathcal{A}\,| = \left(\prod_{i=1}^{m-1} a_{i,n+1-i}\right) A_m$. □

1.3.2 Definition of Subresultants

Definition 1.3.4 Let A and B be polynomials in $\mathcal{R}[X] \oplus \mathcal{R}$ with respective degrees m and n, where $m \geq n \geq 0$. For $j = n-1,\, n-2,\, \ldots,\, 0,\, -1$, we define the *jth subresultant* of A and B to be the determinant polynomial

$$\mathrm{sres}_j(A,B) = |\, \underbrace{\Theta^{n-j-1}A,\, \ldots,\, \Theta A,\, A}_{n-j},\, \underbrace{\Theta^{m-j-1}B,\, \ldots,\, \Theta B,\, B}_{m-j} \,|,$$

The nth subresultant of A and B is defined to be B. The sequence

$$\mathcal{S}(A,B):\; A,\, B,\, \mathrm{sres}_{n-1}(A,B),\, \ldots,\, \mathrm{sres}_{-1}(A,B)$$

is called the *subresultant sequence* of A and B.

Example 1.3.5 Let $A = a_2X^2 + a_1X + a_0X^0 + a_{-1}X^{-1}$ and $B = b_2X^2 + b_1X + b_0X^0 + b_{-1}X^{-1}$. Remark 1.3.3 enables us to describe $\mathcal{S}(A,B)$ by determinants as follows.

$$\mathrm{sres}_1(A,B) = |\, A, B \,| = \det \begin{pmatrix} a_2 & A \\ b_2 & B \end{pmatrix}.$$

$$\mathrm{sres}_0(A,B) = |\, \Theta A, A, \Theta B, B \,| = \det \begin{pmatrix} \sigma a_2 & \delta a_2 + \sigma a_1 & \delta a_1 + \sigma a_0 & \Theta A \\ 0 & a_2 & a_1 & A \\ \sigma b_2 & \delta b_2 + \sigma b_1 & \delta b_1 + \sigma b_0 & \Theta B \\ 0 & b_2 & b_1 & B \end{pmatrix}.$$

$$\mathrm{sres}_{-1}(A,B) = |\, \Theta^2 A, \Theta A, A, \Theta^2 B, \Theta B, B \,|,$$

that is

$$\det \begin{pmatrix} \sigma^2 a_2 & \delta\sigma a_2 + \sigma\delta a_2 + \sigma^2 a_1 & \delta^2 a_2 + \delta\sigma a_1 + \sigma\delta a_1 + \sigma^2 a_0 & \delta^2 a_1 + \delta\sigma a_0 + \sigma\delta a_0 & \delta^2 a_0 & \Theta^2 A \\ 0 & \sigma a_2 & \delta a_2 + \sigma a_1 & \delta a_1 + \sigma a_0 & \delta a_0 & \Theta A \\ 0 & 0 & a_2 & a_1 & a_0 & A \\ \sigma^2 b_2 & \delta\sigma b_2 + \sigma\delta b_2 + \sigma^2 b_1 & \delta^2 b_2 + \delta\sigma b_1 + \sigma\delta b_1 + \sigma^2 b_0 & \delta^2 b_1 + \delta\sigma b_0 + \sigma\delta b_0 & \delta^2 b_0 & \Theta^2 B \\ 0 & \sigma b_2 & \delta b_2 + \sigma b_1 & \delta b_1 + \sigma b_0 & \delta b_0 & \Theta B \\ 0 & 0 & b_2 & b_1 & b_0 & B \end{pmatrix}.$$

If A and B are in $\mathcal{R}[X]$, then the coefficients of X^{-1} in the subresultants of A and B are all equal to zero because the last column of $\mathrm{mat}(\Theta^{n-j-1}A, \ldots, A,\ \Theta^{m-j-1}B, \ldots, B)$ is composed of zero entries. If $\sigma = \mathbf{1}$, $\delta = \mathbf{0}$, and $A, B \in \mathcal{R}[X]$, then Definition 1.3.4 defines the algebraic subresultants in [8, 9, 4]. If $\sigma = \mathbf{1}$ and $\delta = D$, as given in Example 1.2.4, then Definition 1.3.4 defines the differential subresultants in [5, 24].

Some elementary properties of subresultants are given in the next lemma.

Lemma 1.3.5 If A and B are in $\mathcal{R}[X] \oplus \mathcal{R}$ with respective degrees m and n, where $m \geq n \geq 0$, then

1. $\mathrm{sres}_j(A, B) \in [A, B]$, where $n - 1 \geq j \geq -1$;
2. $\deg(\mathrm{sres}_j(A, B)) \leq j$, where $n - 1 \geq j \geq -1$;
3. $\mathrm{sres}_{n-1}(A, B) = (-1)^{m-n+1}\mathrm{prem}(A, B)$.

Proof The first assertion follows from Lemma 1.3.1. Since the matrix

$$\mathrm{mat}(\Theta^{n-j-1}A, \ldots, A,\ \Theta^{m-j-1}B, \ldots, B)$$

has $(m + n - 2j)$ rows and $(m + n - j + 1)$ columns, the second assertion holds by Definition 1.3.2. Since $\mathrm{sres}_{n-1}(A, B) = \mid A,\ \Theta^{m-n}B, \ldots, B \mid$, the pseudo-remainder formula (1.6) and lemma 1.3.2 imply that

$$\mathrm{lc}(B)^{[m-n+1]}\mathrm{sres}_{n-1}(A, B) = \mid \mathrm{prem}(A, B),\ B^{[k]}, \ldots, B \mid .$$

Moving $\mathrm{prem}(A, B)$ to the last row of the above determinant, we get

$$\mathrm{lc}(B)^{[m-n+1]}\mathrm{sres}_{n-1}(A, B) = (-1)^{m-n+1} \mid B^{[m-n]}, \ldots, B, \, \mathrm{prem}(A, B) \mid .$$

Therefore, $\mathrm{sres}_{n-1}(A, B) = (-1)^{m-n+1}\mathrm{prem}(A, B)$ by Lemma 1.3.4. □

1.3.3 Row Reduction on Subresultants

Some proofs in the algebraic subresultant theory are based on the fact that, if A and B are two univariate commutative polynomials in the indeterminate x, then

$$x \, \mathrm{prem}(A, B) = \mathrm{prem}(xA, xB).$$

However, the Ore operator Θ and pseudo-division for Ore polynomials do not commute, that is, if A and B are two Ore polynomials, then in general

$$\Theta(\mathrm{prem}(A, B)) \neq \mathrm{prem}(\Theta A, \Theta B).$$

The following lemma describes the relation between $\Theta(\mathrm{prem}(A, B))$ and $\mathrm{prem}(\Theta A, \Theta B)$.

Lemma 1.3.6 Let A and B be in $\mathcal{R}[X] \oplus \mathcal{R}$, with respective degrees m and n, where $m \geq n \geq 0$. If $C = \mathrm{prem}(A, B)$ and $C_k = \mathrm{prem}(\Theta^k A, \Theta^k B)$, for $k \in \mathbf{N}^+$, then $C_k - \Theta^k C$ is an $\mathcal{R}$-linear combination of $\Theta^{k-1}A, \ldots, A, \Theta^{m-n+k-1}B, \ldots, B$.

Proof By the pseudo-remainder formula (1.6), we write

$$\mathrm{lc}(B)^{[m-n+1]}A = \sum_{i=0}^{m-n} r_i \Theta^i B + C \tag{1.11}$$

and

$$\left(\sigma^k \mathrm{lc}(B)\right)^{[m-n+1]} \Theta^k A = \sum_{i=0}^{m-n} s_i \Theta^{k+i} B + C_k, \tag{1.12}$$

where each of the r_i's and s_i's belongs to $\mathcal{R}$. Applying Θ to equation (1.11) k times and using the extended Leibniz rule (Lemma 1.2.4), we obtain

$$\left(\sigma^k \mathrm{lc}(B)\right)^{[m-n+1]} \Theta^k A + \sum_{j=0}^{k-1} q_j \Theta^j A = \sum_{i=0}^{m-n+k} h_i \Theta^i B + \Theta^k C, \tag{1.13}$$

where each of the q_j's and h_i's belongs to $\mathcal{R}$. Equations (1.12) and (1.13) imply that

$$\sum_{j=0}^{k-1} q_j \Theta^j A = \sum_{i=0}^{m-n} (h_{k+i} - s_i) \Theta^{k+i} B + \sum_{l=0}^{k-1} h_l \Theta^l B + \Theta^k C - C_k.$$

Since $\Theta^{m-n+k}B$ is the only polynomial of degree $(k+m)$ in the above equality, $h_{m-n+k} - s_{m-n} = 0$. Hence, $\Theta^k C - C_k$ is an $\mathcal{R}$-linear combination of $\Theta^{k-1}A, \ldots\ A,\ \Theta^{k+m-n-1}B, \ldots, B$. $\square$

We are ready to present the *row-reduction formula for subresultants*, by which the techniques for proving the algebraic subresultant theorem can be extended to Ore polynomials.

Theorem 1.3.7 *Let A and B be in $\mathcal{R}[X] \oplus \mathcal{R}$, with respective degrees m and n, where $m \geq n \geq 0$. If there exist non-zero $u, v, w \in \mathcal{R}$ and $F, G \in \mathcal{R}[X] \oplus \mathcal{R}$ such that $uB = vF$ and $\mathrm{sres}_{n-1}(A, B) = w\,G$, then*

$$\begin{aligned} & u^{[m-i]}\mathrm{lc}(B)^{[m-n+1][n-i]}\mathrm{sres}_i(A,B) = \\ & \quad v^{[m-i]}w^{[n-i]} \mid \Theta^{m-i-1}F, \ldots, F,\ \Theta^{n-i-1}G, \ldots, G \mid, \end{aligned} \tag{1.14}$$

for $i = n-1,\ n-2,\ \ldots,\ -1$.

Proof Let $C = \mathrm{prem}(A, B)$, $C_k = \mathrm{prem}(\Theta^k A, \Theta^k B)$, for $k \in \mathbf{N}$, and $S_i = \mathrm{sres}_i(A, B)$, for $i = n-1, n-2, \ldots, -1$. Note that $S_i = \mid \Theta^{n-i-1}A, \ldots, \Theta A,\ A,\ \Theta^{m-i-1}B, \ldots, \Theta B,\ B \mid$ by Definition 1.3.4. The pseudo-remainder formula for $\Theta^{n-i-1}A$ and $\Theta^{n-i-1}B$ implies that

$$\sigma^{n-i-1}(\mathrm{lc}(B))^{[m-n+1]}\Theta^{n-i-1}A - C_{n-i-1}$$

is an $\mathcal{R}$-linear combination of $\Theta^{m-i-1}B, \ldots, \Theta^{n-i}B, \Theta^{n-i-1}B$. Therefore,

$$\sigma^{n-i-1}(\mathrm{lc}(B))^{[m-n+1]}\Theta^{n-i-1}A - \Theta^{n-i-1}C$$

is an $\mathcal{R}$-linear combination of $\Theta^{n-i-2}A, \ldots, \Theta A, A, \Theta^{m-i-1}B, \ldots, \Theta B, B$ by Lemma 1.3.6. It then follows from Lemma 1.3.2 that

$$\sigma^{n-i-1}(\mathrm{lc}(B))^{[m-n+1]}S_i = \mid \Theta^{n-i-1}C, \Theta^{n-i-2}A, \ldots, A,\ \Theta^{m-i-1}B, \ldots, B \mid. \tag{1.15}$$

In the same way, we replace $\Theta^j A$ by $\Theta^j C$ on the right-hand side of equation (1.15), while, simultaneously, we multiply the power $\sigma^j(\mathrm{lc}(B))^{[m-n+1]}$ on the left-hand side of the same equation, for $j = n-i-2,\ n-i-3, \ldots, 0$. We eventually arrive at

$$\mathrm{lc}(B)^{[n-i][m-n+1]}S_i = \mid \Theta^{n-i-1}C, \Theta^{n-i-2}C, \ldots, C,\ \Theta^{m-i-1}B, \ldots, B \mid.$$

Then, by the third assertion of Lemma 1.3.5,

$$\mathrm{lc}(B)^{[n-i][m-n+1]}S_i = \mid \Theta^{m-i-1}B, \ldots, B,\ \Theta^{n-i-1}S_{n-1}, \ldots, S_{n-1} \mid.$$

This theorem thus follows from Lemma 1.3.3. $\square$

1.4 Subresultant Theorem and Algorithm

Notation *To avoid endlessly repeating the same assumptions, in this section we let A and B be in $\mathcal{R}[X] \oplus \mathcal{R}$, with respective degrees m and n, where $m \geq n \geq 0$. Let S_n be B and S_j be $\mathrm{sres}_j(A, B)$, for $j = n-1, n-2, \ldots, -1$. The subresultant sequence $\mathcal{S}(A, B)$ consists of A, S_n, $\ldots$, S_0, S_{-1}.*

This section has two parts. First, we prove the subresultant theorem and describe the gap structure of a subresultant sequence. Second, we present the subresultant algorithm.

1.4.1 Subresultant Theorem

Definition 1.4.1 The jth subresultant S_j is *regular* if S_j is of degree j, otherwise S_j is *defective*. In particular, the nth subresultant S_n is always regular.

First, we demonstrate the relation between the members of $\mathcal{S}(A, B)$ and subresultants of two consecutive non-zero members of $\mathcal{S}(A, B)$ in the next lemma. The subresultant theorem is one of its consequences. The proof given below is based on somewhat tedious calculations because of the presence of σ-factorial expressions.

Lemma 1.4.1 Let

$$\alpha_i = \mathrm{lc}(S_i), \quad (n \geq i \geq -1), \quad \beta_n = \sigma\mathrm{lc}(S_n)^{[m-n]}, \quad \text{and} \quad \beta_i = \sigma\mathrm{lc}(S_i), \quad (n-1 \geq i \geq -1).$$

If S_{j+1} is regular and S_j has degree r, for some j such that $n-1 \geq j \geq 0$, then the following hold:

1. If $r \leq -1$, then
$$S_i = 0 \quad (j-1 \geq i \geq -1). \tag{1.16}$$

2. If $r \geq 0$, then
$$S_i = 0 \quad (j-1 \geq i \geq r+1), \tag{1.17}$$
$$\beta_{j+1}^{[j-r]} S_r = \beta_j^{[j-r]} S_j, \tag{1.18}$$
and
$$\alpha_{j+1}^{[r-i]} \beta_{j+1}^{[j-i]} S_i = \mathrm{sres}_i(S_{j+1}, S_j) \quad (r-1 \geq i \geq -1). \tag{1.19}$$

Proof We proceed by induction on the sequence of the regular subresultants in $\mathcal{S}(A, B)$. As S_n is the first regular subresultant in $\mathcal{S}(A, B)$, we start with the case $j = n-1$. Let i be an integer such that $n-2 \geq i \geq -1$. By Definition 1.3.4 we have $S_i = \mid \Theta^{n-1-i}A, \ldots, A,\ \Theta^{m-1-i}S_n, \ldots, S_n \mid$. It follows from the row-reduction formula (1.14) that

$$\alpha_n^{[m-n+1][n-i]} S_i = R_i, \tag{1.20}$$

where $R_i = \mid \Theta^{m-1-i}S_n, \ldots, S_n,\ \Theta^{n-1-i}S_{n-1}, \ldots, \Theta S_{n-1}, S_{n-1} \mid$.

If S_{n-1} has degree less than 0, then ΘS_{n-1} and S_{n-1} are $\mathcal{R}$-linearly dependent, so $R_i = 0$, and hence $S_i = 0$ by (1.20), for $i = n-2, n-3, \ldots, -1$.

Assume that $r \geq 0$. If $n-2 \geq i \geq r+1$, then $\deg S_n > 1 + \deg \Theta^{n-1-i} S_{n-1}$. Thus, $R_i = 0$ by Lemma 1.3.4, consequently, $S_i = 0$ by (1.20).

If $i = r$, then $R_r = \alpha_n^{[m-r]} \beta_{n-1}^{[n-1-r]} S_{n-1}$ by the second assertion of Lemma 1.3.4. Hence equation (1.20) can be rewritten as

$$\alpha_n^{[m-n+1][n-r]} S_r = \alpha_n^{[m-r]} \beta_{n-1}^{[n-1-r]} S_{n-1}. \tag{1.21}$$

As

$$\begin{aligned} \alpha_n^{[m-n+1][n-r]} &= \alpha_n^{[m-r]} (\sigma\alpha_n)^{[m-n][n-1-r]} \quad \text{(by (4) in Lemma 1.2.5)} \\ &= \alpha_n^{[m-r]} \beta_n^{[n-1-r]}, \end{aligned}$$

the equation $\beta_n^{[n-1-r]} S_r = \beta_{n-1}^{[n-1-r]} S_{n-1}$ holds by (1.21).

If $r-1 \geq i \geq -1$, then $R_i = (\sigma^{r-i}\alpha_n)^{[m-r]} \mathrm{sres}_i(S_n, S_{n-1})$ by the first assertion of Lemma 1.3.4. This equation and (1.20) imply that

$$\alpha_n^{[m-n+1][n-i]} S_i = (\sigma^{r-i}\alpha_n)^{[m-r]} \mathrm{sres}_i(S_n, S_{n-1}). \tag{1.22}$$

As

$$\begin{aligned} \alpha_n^{[m-n+1][n-i]} &= \alpha_n^{[m-i]} \beta_n^{[n-1-i]} \quad \text{(by (4) in Lemma 1.2.5)} \\ &= \alpha_n^{[r-i]} (\sigma^{r-i}\alpha_n)^{[m-r]} \beta_n^{[n-1-i]} \quad \text{(by (2) in Lemma 1.2.5)}, \end{aligned}$$

the equation $\alpha_n^{[r-i]} \beta_n^{[n-1-i]} S_i = \mathrm{sres}_i(S_n, S_{n-1})$ holds by (1.22). The proof of the base case is done.

We assume that the lemma holds for the regular subresultant S_{j+1}, and that $\deg S_j = r$, i.e., equations (1.16), (1.17), (1.18), and (1.19) hold. If $r \leq -1$, there is no non-zero subresultant following S_j, so there is nothing to prove. Suppose that $r \geq 0$. Then the regular subresultant next to S_{j+1} must be S_r by the induction hypothesis. Let $\deg(S_{r-1}) = t$. We have to prove that, if $t \leq -1$, then

$$S_i = 0 \quad (r-2 \geq i \geq -1); \tag{1.23}$$

and that, if $t \geq 0$, then

$$S_i = 0 \quad (r-2 \geq i \geq t+1), \tag{1.24}$$

$$\beta_r^{[r-1-t]} S_t = \beta_{r-1}^{[r-1-t]} S_{r-1}, \tag{1.25}$$

and

$$\alpha_r^{[t-i]} \beta_r^{[r-1-i]} S_i = \mathrm{sres}_i(S_r, S_{r-1}) \quad (t-1 \geq i \geq -1). \tag{1.26}$$

Before going to induction, we point out two important relations hiding in (1.18). Equating the leading coefficients of both sides of (1.18) yields

$$\beta_{j+1}^{[j-r]} \alpha_r = \beta_j^{[j-r]} \alpha_j. \tag{1.27}$$

Applying σ to both sides of (1.27) yields

$$(\sigma\beta_{j+1})^{[j-r]} \beta_r = \beta_j^{[j-r+1]}. \tag{1.28}$$

We claim that

$$\alpha_r^{[j-i+1]} \beta_r^{[r-i-1]} S_i = T_i, \quad (r-2 \geq i \geq -1), \tag{1.29}$$

where $T_i = \mid \Theta^{j-i} S_r, \ldots, S_r,\ \Theta^{r-i-1} S_{r-1}, \ldots, \Theta S_{r-1}, S_{r-1} \mid$.

Proof of the Claim. Equations (1.18) and (1.19) (setting $i = r-1$) give us

$$\beta_j^{[j-r]} S_j = \beta_{j+1}^{[j-r]} S_r \quad \text{and} \quad \mathrm{sres}_{r-1}(S_{j+1}, S_j) = \left(\alpha_{j+1} \beta_{j+1}^{[j-r+1]}\right) S_{r-1}. \tag{1.30}$$

Using the relations given in (1.30) and row-reduction formula (1.14), we derive from (1.19) that

$$\left(\beta_j^{[j-r][j-i+1]}\right) \left(\alpha_j^{[j-r+2][r-i]}\right) \left(\alpha_{j+1}^{[r-i]} \beta_{j+1}^{[j-i]} S_i\right) = \left(\beta_{j+1}^{[j-r][j-i+1]}\right) \left(\alpha_{j+1}^{[r-i]} \beta_{j+1}^{[j-r+1][r-i]}\right) T_i.$$

Let

$$r_i = \frac{\left(\beta_j^{[j-r][j-i+1]} \alpha_j^{[j-r+2][r-i]}\right) \left(\alpha_{j+1}^{[r-i]} \beta_{j+1}^{[j-i]}\right)}{\beta_{j+1}^{[j-r][j-i+1]} \left(\alpha_{j+1}^{[r-i]} \beta_{j+1}^{[j-r+1][r-i]}\right)}.$$

Then $r_i S_i = T_i$. Our claim will be proved if we show $r_i = \alpha_r^{[j-i+1]} \beta_r^{[r-i-1]}$. Canceling $\alpha_{j+1}^{[r-i]}$ yields

$$r_i = \left(\frac{\beta_j^{[j-r][j-i+1]}}{\beta_{j+1}^{[j-r][j-i+1]}}\right) \frac{\alpha_j^{[j-r+2][r-i]} \beta_{j+1}^{[j-i]}}{\beta_{j+1}^{[j-r+1][r-i]}}.$$

The above equality can be simplified by (1.27) to

$$r_i = \left(\frac{\alpha_r^{[j-i+1]}}{\alpha_j^{[j-i+1]}}\right) \frac{\alpha_j^{[j-r+2][r-i]} \beta_{j+1}^{[j-i]}}{\beta_{j+1}^{[j-r+1][r-i]}}. \tag{1.31}$$

The fourth equality in Lemma 1.2.5 implies that

$$\alpha_j^{[j-r+2][r-i]} = \alpha_j^{[j-i+1]}\beta_j^{[j-r+1][r-i-1]} \quad \text{and} \quad \beta_{j+1}^{[j-r+1][r-i]} = \beta_{j+1}^{[j-i]}(\sigma\beta_{j+1})^{[j-r][r-i-1]}.$$

So equation (1.31) can be further simplified to

$$r_i = \alpha_r^{[j-i+1]}\left(\frac{\beta_j^{[j-r+1][r-i-1]}}{(\sigma\beta_{j+1})^{[j-r][r-i-1]}}\right).$$

It then follows from (1.28) that $r_i = \alpha_r^{[j-i+1]}\beta_r^{[r-i-1]}$. The claim is proved.

If $t \leq -1$, then $T_i = 0$, for $r-2 \geq i \geq -1$, because ΘS_{r-1} and ΘS_{r-1} are $\mathcal{R}$-linearly dependent, so $S_i = 0$ by (1.29), for $i = r-2, r-3, \ldots, -1$.

Assume that $t \geq 0$. If $r-2 \geq i \geq t+1$, then $T_i = 0$ since $\deg(S_r) > 1 + \deg(\Theta^{r-i-1}S_{r-1})$. Hence $S_i = 0$ by (1.29).

If $i = t$, then $T_i = \alpha_r^{[j-t+1]}\beta_{r-1}^{[r-t-1]}S_{r-1}$ by Lemma 1.3.4. Equation (1.25) holds by (1.29).

If $t-1 \geq i \geq -1$, then $T_i = (\sigma^{t-i}\alpha_r)^{[j-t+1]}\mathrm{sres}_i(S_r, S_{r-1})$ by Lemma 1.3.4, and hence (1.29) implies that

$$\alpha_r^{[j-i+1]}\beta_r^{[r-i-1]}S_i = (\sigma^{t-i}\alpha_r)^{[j-t+1]}\mathrm{sres}_i(S_r, S_{r-1}). \tag{1.32}$$

It follows from the second assertion of Lemma 1.2.5 that

$$\alpha_r^{[j-i+1]} = \alpha_r^{[(t-i)+(j-t+1)]} = \alpha_r^{[t-i]}(\sigma^{t-i}\alpha_r)^{[j-t+1]}.$$

Using this relation to remove the like σ-factorials from both sides of (1.32), we get (1.26). □

Theorem 1.4.2 (Subresultant Theorem) *Let*

$$\alpha_i = \mathrm{lc}(S_i) \quad (n \geq i \geq -1), \quad \beta_n = \sigma\mathrm{lc}(S_n)^{[m-n]}, \quad \text{and} \quad \beta_i = \sigma\mathrm{lc}(S_i) \quad (n-1 \geq i \geq -1).$$

If S_{j+1} is regular and S_j has degree r, for some j with $n-1 \geq j \geq 0$, then the following hold:

1. *If $r \leq -1$, then*

$$S_i = 0, \quad (j-1 \geq i \geq -1). \tag{1.33}$$

2. *If $r \geq 0$, then*

$$S_i = 0, \quad (j-1 \geq i \geq r+1), \tag{1.34}$$

$$\beta_{j+1}^{[j-r]}S_r = \beta_j^{[j-r]}S_j, \tag{1.35}$$

and

$$\alpha_{j+1}\beta_{j+1}^{[j-r+1]}S_{r-1} = (-1)^{j-r}\mathrm{prem}(S_{j+1}, S_j). \tag{1.36}$$

Proof Equations (1.33), (1.34), and (1.35) hold by Lemma 1.4.1. Set $i = r-1$. Then equation (1.19) in Lemma 1.4.1 becomes $\alpha_{j+1}\beta_{j+1}^{[j-r+1]}S_{r-1} = \mathrm{sres}_{r-1}(S_n, S_{n-1})$. Hence, equation (1.36) holds by the third assertion of Lemma 1.3.5. $\square$

If $\sigma = \mathbf{1}$ and $\delta = \mathbf{0}$, and $A, B \in \mathcal{R}[X]$, then Theorem 1.4.2 becomes the algebraic subresultant theorem in [26]. If $\sigma = \mathbf{1}$ and $\delta = D$, as given in Example 1.2.4, then Theorem 1.4.2 becomes the differential subresultant theorem in [24].

The next corollary is a formula-free version of the subresultant theorem.

Corollary 1.4.3 Let $\deg S_{j+1} = j+1$ and $\deg S_j = r$, for some j such that $n-1 \geq j \geq 0$. If $r \leq -1$, then $S_i = 0$, for $i = j-1,\ j-2,\ \ldots,\ -1$. If $r > -1$, then $S_i = 0$, for $i = j-1,\ j-2,\ \ldots,\ r+1$, $S_j \sim_{\mathcal{R}} S_r$, and $S_{r-1} \sim_{\mathcal{R}} \mathrm{prem}(S_{j+1}, S_j)$. $\square$

Definition 1.4.2 A defective subresultant is said to be *isolated* if it is of degree -1.

Remark 1.4.3 $\mathcal{S}(A, B)$ does not contain any isolated subresultant if A and B belong to $\mathcal{R}[X]$.

Now, we extend subresultant sequences of the first and second kinds (see Rubald [37]). We prove that subresultant sequences of the first kind are PRS's in the next section.

Definition 1.4.4 *The subresultant sequence of A and B of the first kind* is the subsequence of $\mathcal{S}(A, B)$ that consists of the following polynomials:

1. A, B, and

2. S_j, if S_{j+1} is regular and S_j is nonzero.

The subresultant sequence of A and B of the second kind is the subsequence of $\mathcal{S}(A, B)$ that consists of A, B and other regular subresultants of $\mathcal{S}(A, B)$. The subresultant sequences of A and B of the first and second kinds are denoted by $\mathcal{S}_1(A, B)$ and $\mathcal{S}_2(A, B)$, respectively.

The next corollary describes the relation between $\mathcal{S}_1(A, B)$ and $\mathcal{S}_2(A, B)$.

Corollary 1.4.4 Let $\mathcal{S}_2(A, B)$ consist of A, S_n, S_{j_3}, $S_{j_4}, \cdots, S_{j_{l-1}}$, S_{j_l}. If $\mathcal{S}(A, B)$ does not contain any isolated subresultant, then $\mathcal{S}_1(A, B)$ consists of A, S_n, S_{n-1}, S_{j_3-1}, $S_{j_4-1}, \ldots, S_{j_{l-1}-1}$. Otherwise, $\mathcal{S}_1(A, B)$ consists of A, S_n, S_{n-1}, S_{j_3-1}, $S_{j_4-1}, \ldots, S_{j_{l-1}-1}$, S_{j_l-1}. In any case we have $S_{n-1} \sim_{\mathcal{R}} S_{j_3}$ and $S_{j_i-1} \sim_{\mathcal{R}} S_{j_{i+1}}$, for $i = 3,\ 4,\ \ldots,\ l-1$.

Proof The sequence A, S_n, S_{n-1}, S_{j_3-1}, S_{j_4-1}, $\ldots$, $S_{j_{l-1}-1}$ is a subsequence of $\mathcal{S}_1(A,B)$ by Definition 1.4.4. If S_{j_l-1} is zero, then all the subresultants following S_{j_l} are zero by Corollary 1.4.3. If S_{j_l-1} is nonzero, then it must be isolated, otherwise there would be a regular subresultant following S_{j_l} by Corollary 1.4.3, which is a contradiction. Since S_n is regular, $S_{n-1} \sim_{\mathcal{R}} S_{j_3}$ by Corollary 1.4.3. In the same way we deduce that $S_{j_i-1} \sim_{\mathcal{R}} S_{j_{i+1}}$, for $i = 3, 4, \ldots, l-1$. $\square$

If $S_{j_{i+1}}$ and S_{j_i} are consecutive members in $\mathcal{S}_2(A,B)$, then the S_i's between $S_{j_{i+1}-1}$ and S_{j_i} are all zero by Corollary 1.4.3. Hence, all the non-zero subresultants are contained in either $\mathcal{S}_2(A,B)$ or $\mathcal{S}_1(A,B)$. Accordingly, all the defective subresultants are contained in $\mathcal{S}_1(A,B)$. If there is no defective subresultant in $\mathcal{S}(A,B)$, then both $\mathcal{S}_1(A,B)$ and $\mathcal{S}_2(A,B)$ coincide with $\mathcal{S}(A,B)$.

Corollary 1.4.5 If there exists an isolated subresultant in $\mathcal{S}(A,B)$, then it is the last non-zero member in $\mathcal{S}(A,B)$.

Proof If S_j is isolated, then S_j is contained in $\mathcal{S}_1(A,B)$, and S_{j+1} is regular. Hence, all the subresultants following S_j are equal to zero by Corollary 1.4.3. $\square$

The gap structure of $\mathcal{S}(A,B)$ is given in Figure 1.1.

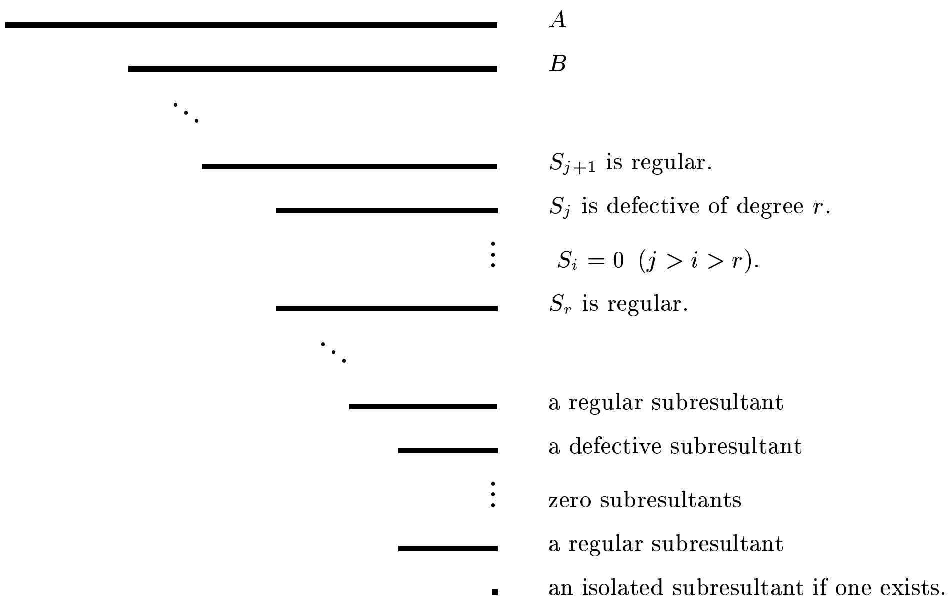

Figure 1.1: The gap structure of $\mathcal{S}(A,B)$

Note that the gap-structure of the subresultant sequence of two Ore polynomials is slightly more complicated than that of an algebraic subresultant sequence due to the possible presence of isolated subresultants.

In summary, the subresultant theorem and its corollaries reveal the following:

- If S_i and S_j $(i > j)$ are both nonzero and of the same degree, then S_i and S_j are two consecutive non-zero subresultants in $\mathcal{S}(A, B)$, S_i is defective, S_j is regular, and $S_i \sim_{\mathcal{R}} S_j$.
- If S_i and S_j $(i > j)$ are two consecutive non-zero subresultants with distinct degrees, then $i = j + 1$, S_i is regular, and $\mathrm{prem}(S_i, S_j) \sim_{\mathcal{R}} S_{r-1}$, where $r = \deg S_j$.
- The coefficients of similarity mentioned above are given explicitly by (1.35) and (1.36) in the subresultant theorem.

1.4.2 Subresultant Algorithm

Throughout this section $\mathcal{S}_1(A, B)$ and $\mathcal{S}_2(A, B)$ are

$$A_1,\, A_2,\, A_3, \cdots, A_{k_1}, \quad \text{and,} \quad B_1,\, B_2,\, B_3, \cdots, B_{k_2},$$

respectively, where $A_1 = B_1 = A$ and $A_2 = B_2 = B$. By Corollary 1.4.4, $k_1 = k_2$ if there is no isolated subresultant in $\mathcal{S}(A, B)$, otherwise $k_1 = k_2 + 1$.

Lemma 1.4.6 Let $b_2 = \mathrm{lc}(B)^{[m-n]}$, $a_i = \mathrm{lc}(A_i)$, $b_i = \mathrm{lc}(B_i)$, and $l_i = \deg A_{i-1} - \deg A_i + 1$, for $i = 3, \ldots, k_2$. Then

$$(\sigma a_i)^{[l_i - 2]} A_i = (\sigma b_{i-1})^{[l_i - 2]} B_i. \tag{1.37}$$

In particular, $a_i^{[l_i - 1]} = (\sigma b_{i-1})^{[l_i - 2]} b_i$.

Proof Let $B_{i-1} = S_{j+1}$. Then $A_i = S_j$ by Corollary 1.4.4. Let $\deg A_i = r$. Then $B_i = S_r$ by Corollary 1.4.3. Since $\deg A_{i-1} = \deg B_{i-1} = j + 1$ by Corollary 1.4.4, we get $l_i = j - r + 2$. Hence equation (1.37) holds by (1.35) in the subresultant theorem. Equating the leading coefficients of both sides of (1.37) yields $a_i^{[l_i - 1]} = (\sigma b_{i-1})^{[l_i - 2]} b_i$. □

The subresultant algorithm given in the next theorem generalizes the algebraic subresultant algorithm by Brown and Traub [4]. This algorithm computes $\mathcal{S}_1(A, B)$ without expanding determinants directly, and proceeds as the Euclidean algorithm but removes a factor from the coefficients in the pseudo-remainder after each pseudo-division. A byproduct of this algorithm is the sequence of the leading coefficients of the members of $\mathcal{S}_2(A, B)$. Consequently, we can use Lemma 1.4.6 to construct $\mathcal{S}(A, B)$ from the output of the subresultant algorithm.

Theorem 1.4.7 (Subresultant Algorithm) *Let*

$$a_1 = b_1 = 1, \quad a_2 = \mathrm{lc}(A_2), \quad \textit{and} \quad b_2 = \mathrm{lc}(B_2)^{[m-n]},$$

and let

$$a_i = \mathrm{lc}(A_i), \quad b_i = \mathrm{lc}(B_i), \quad \textit{and} \quad l_i = \deg A_{i-1} - \deg A_i + 1,$$

for $i = 3, \ldots, \min(k_1, k_2)$. *Then*

$$A_i = \mathrm{prem}(A_{i-2}, A_{i-1})/e_i, \tag{1.38}$$

where

$$e_i = (-1)^{l_{i-1}}(\sigma b_{i-2})^{[l_{i-1}-1]}a_{i-2}, \tag{1.39}$$

for $i = 3,\ 4,\ \ldots,\ k_1$. *In particular,* $\mathcal{S}_1(A, B)$ *is a PRS of* A *and* B.

Proof We handle the cases in which $i = 3$ or 4, and then consider the general case.

If $i = 3$, then $e_3 = (-1)^{m-n+1}$, so $\mathrm{prem}(A_1, A_2) = (-1)^{m-n+1}S_{n-1} = e_3A_3$ by the third assertion of Lemma 1.3.5. Note that $k_1 \leq 3$ if $\deg S_{n-1} \leq -1$ by Corollary 1.4.3. To proceed, we assume that $\deg A_3 = r \geq 0$. If $i = 4$, then

$$e_4 = (-1)^{n-r+1}a_2(\sigma b_2)^{[l_3-1]} = \mathrm{lc}(B)\sigma(\mathrm{lc}(B))^{[m-n][n-r]}.$$

Equation (1.36) in the subresultant theorem (setting $j = n-1$) implies that $e_4S_{r-1} = \mathrm{prem}(A_2, A_3)$. If S_{r-1} is nonzero, then $A_4 = S_{r-1}$ since S_r is regular.

Let $5 \leq i \leq k_1$. By (1.37) in Lemma 1.4.6, we have $(\sigma a_{i-2})^{[l_{i-2}-2]}A_{i-2} = (\sigma b_{i-3})^{[l_{i-2}-2]}B_{i-2}$. Therefore, $\mathrm{prem}((\sigma a_{i-2})^{[l_{i-2}-2]}A_{i-2}, A_{i-1}) = \mathrm{prem}((\sigma b_{i-3})^{[l_{i-2}-2]}B_{i-2}, A_{i-1})$. From this equation we derive

$$(\sigma a_{i-2})^{[l_{i-2}-2]}\mathrm{prem}(A_{i-2}, A_{i-1}) = (\sigma b_{i-3})^{[l_{i-2}-2]}\mathrm{prem}(B_{i-2}, A_{i-1}). \tag{1.40}$$

Let $B_{i-2} = S_{j+1}$. Then $A_{i-1} = S_j$ by Corollary 1.4.4. Assume that $\deg A_{i-1} = r$. Then $B_{i-1} = S_r$ and $A_i = S_{r-1}$ by the same corollary. We deduce that

$$\mathrm{prem}(B_{i-2}, A_{i-1}) = \mathrm{prem}(S_{j+1}, S_j) = (-b_{i-2})^{[l_{i-1}]}A_i, \tag{1.41}$$

where the last equality follows from (1.36) in the subresultant theorem, since $\mathrm{lc}(S_{j+1}) = b_{i-2}$ and $l_{i-1} = j - r + 2$. Equations (1.40) and (1.41) imply that

$$(\sigma a_{i-2})^{[l_{i-2}-2]}\mathrm{prem}(A_{i-2}, A_{i-1}) = (-b_{i-2})^{[l_{i-1}]}(\sigma b_{i-3})^{[l_{i-2}-2]}A_i.$$

Multiplying a_{i-2} to both sides of the above equation yields

$$a_{i-2}^{[l_{i-2}-1]}\mathrm{prem}(A_{i-2}, A_{i-1}) = a_{i-2}(-b_{i-2})^{[l_{i-1}]}(\sigma b_{i-3})^{[l_{i-2}-2]}A_i.$$

Simplifying the σ-factorials of the above equality by Lemma 1.4.6, we see that

$$b_{i-2}\mathrm{prem}(A_{i-2}, A_{i-1}) = (-1)^{l_{i-1}}a_{i-2}b_{i-2}^{[l_{i-1}]}A_i.$$

Equation (1.38) follows.

It remains to prove that $\deg(A_{k_1}) = -1$ or $\mathrm{prem}(A_{k_1-1}, A_{k_1}) = 0$. Assume that $\deg A_{k_1} = r \geq 0$ and that $A_{k_1} = S_j$. Then $B_{k_1-1} = S_{j+1}$ by Corollary 1.4.4. It follows that $\mathrm{prem}(A_{k_1-1}, A_{k_1})$ and $\mathrm{prem}(S_{j+1}, S_j)$ are similar over $\mathcal{R}$. Note that $\mathrm{prem}(S_{j+1}, S_j) = 0$, otherwise S_{r-1} would be nonzero, so S_{r-1} is in $\mathcal{S}_1(A, B)$, contradicting the fact that S_j is the last member of $\mathcal{S}_1(A, B)$. Consequently, $\mathrm{prem}(A_{k_1-1}, A_{k_1}) = 0$. $\square$

Remark 1.4.5 Using (1.38) in Theorem 1.4.7, we may get A_i by computing $\mathrm{prem}(A_{i-2}, A_{i-1})$ and removing the extraneous factor e_i from the pseudo-remainder, where e_i is computed by equation (1.39). At first glance, one might think that one needs both the a_i's and the b_i's to compute the e_i's. However, the recursive formula $b_i = a_i^{[l_i-1]}/(\sigma b_{i-1})^{[l_i-2]}$ in Lemma 1.4.6 enables us to compute the b_i's by the a_i's.

Chapter 2

Applications of the Subresultant Theory

We will apply the subresultant theory developed in Chapter 1 to three fundamental problems, namely, deciding Θ-compatibility, computing GCRDs, and computing LCLMs. Using the subresultant theorem we present a characterization of the Θ-compatibility of two elements in an Ore polynomial module, define the Sylvester resultant, derive determinant formulas for GCRDs and LCLMs, and estimate multiplicative bounds for the denominators of the monic GCRD and LCLM of two elements in an Ore polynomial ring. Propositions 2.2.3 and 2.2.4 in this chapter establish the basis for the modular algorithm for computing GCRDs over $\mathbf{Z}[t]$ in Chapter 3.

The subresultant algorithm described in Chapter 1 may also be applied to various back-and-forth division processes in linear differential and difference algebra, for example, computing the characteristic sets for a linear differential ideal and reducing a system of linear homogeneous equations to a diagonal form (see, respectively, Kolchin [21, pp. 150–155] and Poole [35, pp. 39–41]). But we will not study Ore polynomials of special kinds in this thesis.

Throughout this chapter $(\mathcal{R}[X] \oplus \mathcal{R}, \Theta, \sigma, \delta)$ is an Ore polynomial module. For brevity we denote this module by $\mathcal{R}[X] \oplus \mathcal{R}$. We fix A and B in $\mathcal{R}[X] \oplus \mathcal{R}$ with respective degrees m and n, where $m \geq n \geq 0$.

The organization of this chapter is as follows. In Section 2.1, we present two methods for deciding the Θ-compatibility of two elements in $\mathcal{R}[X] \oplus \mathcal{R}$ and define the Sylvester resultant of two elements in $\mathcal{R}[X]$. Section 2.2 is devoted to studying the relation between GCRDs and subresultants. We apply the subresultant theory to the computation of LCLMs in Section 2.3.

2.1 Deciding Θ-Compatibility by Subresultants

Two methods are presented for deciding Θ-compatibility by subresultants. If $\mathcal{R}[X] \oplus \mathcal{R}$ is the module of linear differential polynomials, the two methods may be seen as the improvements of the differential Euclidean algorithm [35] and differential resultants [1], respectively.

Theorem 2.1.1 The following statements are equivalent:

1. A and B are Θ-compatible.

2. $\mathrm{sres}_{-1}(A, B)$ is equal to zero and the last non-zero member of $\mathcal{S}(A, B)$ is regular.

3. The last member in $\mathcal{S}_1(A, B)$ is of degree greater than -1.

Proof $(1 \Longrightarrow 2)$ Since $\mathrm{sres}_{-1}(A, B)$ is in $[A, B]$, $\mathrm{sres}_{-1}(A, B)$ is equal to zero. If the last non-zero subresultant were defective, then it would be isolated by Corollary 1.4.4, which is a contradiction to the assumption that A and B are Θ-compatible.

$(2 \Longrightarrow 3)$ This is immediate from Corollary 1.4.4.

$(3 \Longrightarrow 1)$ This follows from the fact that $\mathcal{S}_1(A, B)$ is a PRS by Theorem 1.4.7. □

Observe that if $\mathrm{sres}_k(A, B)$ is a member in $\mathcal{S}_1(A, B)$, with degree r, then the only candidate of the member next to $\mathrm{sres}_k(A, B)$ in $\mathcal{S}_1(A, B)$ is $\mathrm{sres}_{r-1}(A, B)$ by Corollary 1.4.3. Using this observation and the third equivalent condition of Theorem 2.1.1, we present the algorithm COMP_t for deciding the Θ-compatibility of A and B. COMP_t proceeds by computing the degrees of the members in $\mathcal{S}_1(A, B)$ in a top-down fashion.

algorithm COMP_t

Input: $A, B \in \mathcal{R}[X] \oplus \mathcal{R}$ with $\deg A \geq \deg B \geq 0$.
Output: TRUE if A and B are Θ-compatible. Otherwise, FALSE.

```
r ← deg B;
while true do {
        r ← deg sres_{r−1}(A, B);
        if r = −∞ then return(TRUE);
        if r = −1 then return(FALSE); }
```

The second algorithm, named COMP_b, for deciding the Θ-compatibility is based on the second assertion of Theorem 2.1.1 and the fact that the last non-zero member in $\mathcal{S}(A, B)$ is either regular or isolated. COMP_b proceeds by computing the degrees of the members in $\mathcal{S}(A, B)$ in a bottom-up fashion.

algorithm COMP_b

Input: $A, B \in \mathcal{R}[X] \oplus \mathcal{R}$ with $\deg A \geq \deg B \geq 0$.
Output: TRUE if A and B are Θ-compatible. Otherwise, FALSE.

1. **If** $\mathrm{sres}_{-1}(A, B) \neq 0$ **then return**(FALSE);
2. $r \leftarrow \deg B$;
3. **for** $i = 0$ **to** r **do {**
4. **if** $\mathrm{coeff}(\mathrm{sres}_i(A, B), X^i) \neq 0$ **then return**(TRUE);
5. **if** $\mathrm{coeff}(\mathrm{sres}_i(A, B), X^{-1}) \neq 0$ **then return**(FALSE); **}**

Remark 2.1.1 In COMP_t and COMP_b, we do not specifically describe how to compute the degree and coefficients of a subresultant, since the ground domain $\mathcal{R}$ is merely a commutative domain. Of course, determinants can always be computed by minor expansion [16, §9.4]. The subresultant algorithm may be used in COMP_t if exact division in $\mathcal{R}$ is computable. Note that we need only decide whether some determinants are equal to zero in both COMP_t and COMP_b.

Next, we study the Θ-compatibility of two Ore polynomials in $\mathcal{R}[X]$.

Definition 2.1.2 For A and B in $\mathcal{R}[X]$, the subresultant $\mathrm{sres}_0(A, B)$ is called *the (right) Sylvester resultant* of A and B and denoted by $\mathrm{res}(A, B)$.

This definition is an extension of the (right) Sylvester-like resultant for two linear differential operators in [1, 5].

Theorem 2.1.2 *For A and B in $\mathcal{R}[X]$, the left ideal $[A, B]$ does not contain any element of degree 0 if and only if $\mathrm{res}(A, B)$ is equal to zero.*

Proof If $[A, B]$ does not contain any element of degree 0, then $\mathrm{res}(A, B)$ is equal to zero because $\deg(\mathrm{res}(A, B)) \leq 0$ and $\mathrm{res}(A, B) \in [A, B]$. Conversely, if $\mathrm{res}(A, B)$ is equal to zero, the last member of $\mathcal{S}_1(A, B)$ is of degree greater than 0. Thus, $[A, B]$ does not contain any elements of degree 0 because $\mathcal{S}_1(A, B)$ is a PRS. □

2.2 Greatest Common Right Divisors

Notation *In the remainder of this chapter, A and B belong to the Ore polynomial ring $\mathcal{R}[X]$.*

The goal of this section is to describe the relation between the GCRDs and subresultants of two Ore polynomials. In order to describe (right) divisibility, we feel it convenient to consider Ore polynomials with coefficients in a commutative field. For this purpose we extend σ and δ to the quotient field of $\mathcal{R}$.

Proposition 2.2.1 If $\mathcal{F}$ is the quotient field of $\mathcal{R}$, then the conjugate operator σ and pseudo-derivation δ can be uniquely extended to $\mathcal{F}$ by letting

$$\sigma\left(\frac{a}{b}\right) = \frac{\sigma a}{\sigma b} \tag{2.1}$$

and

$$\delta\left(\frac{a}{b}\right) = \frac{b(\delta a) - a(\delta b)}{(\sigma b)b}, \tag{2.2}$$

for $a, b \in \mathcal{R}$ with $b \neq 0$.

Proof We have to verify the following:

1. σ is an injective endomorphism of the field $\mathcal{F}$.

2. δ is an endomorphism of the additive group $\mathcal{F}$.

3. For all $r, s \in \mathcal{F}$, $\delta(rs) = \sigma(r)\delta(s) + \delta(r)s$.

From the identity $\delta(ab) = \delta(ba)$, for $a, b \in \mathcal{R}$, and (1.2) in Proposition 1.2.1, it follows that

$$b(\delta a) - a(\delta b) = (\sigma b)(\delta a) - (\sigma a)(\delta b). \tag{2.3}$$

Let $a/b = c/d$, where $a, b, c, d \in \mathcal{R}$ and $bd \neq 0$. Then $\sigma(d)\sigma(a) = \sigma(c)\sigma(b)$ since σ is a ring homomorphism, hence, σ is well defined on $\mathcal{F}$. Applying δ to the equality $da = cb$ yields

$$(\sigma d)(\delta a) + (\delta d)a = (\sigma c)(\delta b) + (\delta c)b,$$

consequently,

$$(\sigma d)(\delta a) - (\sigma c)(\delta b) = (\delta c)b - (\delta d)a.$$

Multiplying both sides of the previous equality by $(\sigma b)d$ yields

$$d((\sigma d)(\sigma b)(\delta a) - \sigma(bc)(\delta b)) = (\sigma b)(bd(\delta c) - (ad)(\delta d)),$$

which, together with the equation $da = cb$, implies that

$$d(\sigma d)((\sigma b)(\delta a) - (\sigma a)(\delta b)) = b(\sigma b)(d(\sigma c) - c(\delta d)).$$

It then follows from equation (2.3) that

$$\frac{b(\delta a) - a(\delta b)}{b(\sigma b)} = \frac{d(\sigma c) - c(\delta d)}{d(\sigma d)}.$$

Hence δ is well defined on $\mathcal{F}$.

Clearly, σ is a ring endomorphism of $\mathcal{F}$. The distributivity of δ with respect to addition is proved by the following calculation: for $a, b, c, \in \mathcal{R}$ with $b \neq 0$,

$$\delta\left(\frac{a}{b} + \frac{c}{b}\right) = \delta\left(\frac{a+c}{b}\right) = \frac{b\delta(a+c) - (a+c)\delta(b)}{(\sigma b)b} = \delta\left(\frac{a}{b}\right) + \delta\left(\frac{c}{b}\right).$$

It remains to verify the multiplicative rule, that is, for all $a, b, c, d \in \mathcal{R}$, with $bd \neq 0$,

$$\delta\left(\frac{a}{b}\frac{c}{d}\right) = \sigma\left(\frac{a}{b}\right)\delta\left(\frac{c}{d}\right) + \delta\left(\frac{a}{b}\right)\left(\frac{c}{d}\right).$$

We calculate

$$\begin{aligned}
&bd\sigma(bd)\left(\sigma\left(\frac{a}{b}\right)\delta\left(\frac{c}{d}\right) + \left(\frac{c}{d}\right)\delta\left(\frac{a}{b}\right)\right) = \\
&= \quad b\sigma(a)(d\delta(c) - c\delta(d)) + c\sigma(d)(b\delta(a) - a\delta(b)) \\
&= \quad bd\sigma(a)\delta(c) - ac\sigma(d)\delta(b) - cb\sigma(a)\delta(d) + cb\sigma(d)\delta(a) \\
&= \quad bd(\sigma(a)\delta(c) + c\delta(a) - c\delta(a)) - ac(\sigma(d)\delta(b) + b\delta(d) - b\delta(d)) - cb\sigma(a)\delta(d) + cb\sigma(d)\delta(a) \\
&= \quad bd\delta(ac) - ac\delta(bd) + cb(\sigma(d) - d)\delta(a) - cb(\sigma(a) - a)\delta(d) \\
&= \quad bd\delta(ac) - ac\delta(bd) + cb(\,\sigma(d)\delta(a) - \sigma(a)\delta(d) - d\delta(a) + a\delta(d)\,) \\
&= \quad bd\delta(ac) - ac\delta(bd) \quad \text{(by equation (2.3))} \\
&= \quad bd\sigma(bd)\delta\left(\frac{a}{b}\frac{c}{d}\right).
\end{aligned}$$

The multiplicative rule holds.

If σ' is a conjugate operator extending σ and δ' is a pseudo-derivation (with respect to σ') extending δ, then, for every non-zero b in $\mathcal{R}$,

$$\sigma'\left(b\frac{1}{b}\right) = \sigma(b)\sigma'\left(\frac{1}{b}\right) = 1,$$

so $\sigma'(1/b) = 1/\sigma(b)$. From the property that $\delta(1) = 0$ (see Remark 1.2.2), we deduce

$$\delta'\left(b\frac{1}{b}\right) = \sigma(b)\delta'\left(\frac{1}{b}\right) + \frac{\delta(b)}{b} = 0.$$

Thus

$$\delta'\left(\frac{1}{b}\right) = -\frac{\delta(b)}{b\sigma(b)}.$$

The uniqueness then follows from the multiplicative rule of σ' and δ'. $\square$

By Propositions 1.2.1 and 2.2.1, the Ore operator Θ on $\mathcal{R}[X]$ can be uniquely extended to $\mathcal{F}[X]$. In the rest of this chapter, the vector space $\mathcal{F}[X]$ is regarded as an Ore polynomial ring whose Ore operator, conjugate operator, and pseudo-derivation are also denoted by Θ, σ, and δ, respectively.

Definition 2.2.1 A non-zero polynomial in $\mathcal{F}[X]$ of highest degree, which divides both A and B on the right, is called a *GCRD* of A and B.

Lemma 2.2.2 If G_1 and G_2 are two GCRDs of A and B, then G_1 and G_2 are similar over $\mathcal{F}$. If the sequence A, B, A_3, $\ldots$, A_k is a PRS, then A_k is a GCRD of A and B.

Proof See Ore [33, p. 484]. $\square$

Example 2.2.2 Let D be the differential operator on $\mathbf{Z}[t]$ that sends t^n to nt^{n-1}, for all $n \in \mathbf{N}^+$. Then $(\mathbf{Z}[t][D], \mathbf{1}, D)$ is an Ore polynomial ring. One can easily verify that $tD^3 = D^2(tD-2)$. Thus, $(tD-2)$ is a GCRD of D^3 and $(tD-2)$. Note that the product of two primitive polynomials is not necessarily primitive. Moreover, there does not exist A in $\mathbf{Z}[t][X]$ such that $D^3 = A(tD-2)$.

Example 2.2.3 Let E be the shift operator on $\mathbf{Z}[t]$ that sends t^n to $(t+1)^n$, for all $n \in \mathbf{N}^+$. Then $(\mathbf{Z}[t][E], E, \mathbf{0})$ is an Ore polynomial ring. If

$$A = t(t+1)E^2 - 2t(t+2)E + (t+1)(t+2) \quad \text{and} \quad B = (t-1)E^2 - (3t-2)E + 2t,$$

then a GCRD of A and B is $G = tE - (t+1)$. Note that $\gcd(\mathrm{lc}(A), \mathrm{lc}(B)) = 1$, but $\mathrm{lc}(G) = t$.

Now, we describe the relation between the GCRDs and subresultants of two Ore polynomials.

Proposition 2.2.3 If d is the degree of the GCRDs of A and B, then the dth subresultant of A and B is a GCRD of A and B.

Proof As A and B are in $\mathcal{R}[X]$, $\mathcal{S}_2(A, B)$ is a PRS of A and B by Corollary 1.4.4. If d is the degree of the GCRDs of A and B, then the last member in $\mathcal{S}_2(A, B)$ is $\mathrm{sres}_d(A, B)$. $\square$

Remark 2.2.4 Proposition 2.2.3 can be directly proved by induction on the degree of B (see Li and Nemes [25]).

Proposition 2.2.4 If d is the degree of the GCRDs of A and B, then the matrix

$$\mathrm{mat}(X^{n-1}A, \ldots, XA, A, X^{m-1}B, \ldots, XB, B)$$

has rank $(m+n-d)$.

Proof Let M be $\mathrm{mat}(X^{n-1}A, \ldots, XA, A, X^{m-1}B, \ldots, XB, B)$. Since $\mathrm{sres}_d(A, B)$ is nonzero, the rows of M represented by

$$X^{n-d-1}A, \ldots, A, X^{m-d-1}B, \ldots, B$$

are $\mathcal{F}$-linearly independent. Therefore, the rows of M represented by

$$X^{n-d-1}A, \ldots, A, X^{m-1}B, \ldots, X^{m-d-1}B, \ldots, B$$

are $\mathcal{F}$-linearly independent. We then have $\mathrm{rank}(M) \geq m+n-d$. On the other hand, there are non-zero $U, V \in \mathcal{F}[X]$ such that $A = U\mathrm{sres}_d(A, B)$ and $B = V\mathrm{sres}_d(A, B)$ by Proposition 2.2.3. Therefore, the polynomials X^iA $(0 \leq i \leq n-1)$ and X^jB $(0 \leq j \leq m-1)$ are $\mathcal{F}$-linear combinations of $X^{m+n-d-1}\mathrm{sres}_d(A, B), \ldots, X\mathrm{sres}_d(A, B), \mathrm{sres}_d(A, B)$, and hence $\mathrm{rank}(M) \leq m+n-d$. □

Corollary 2.2.5 If d is the degree of the GCRDs of A and B, then $\mathrm{lc}(\mathrm{sres}_d(A, B))$ is a multiplicative bound for the denominators of the coefficients in the monic GCRD of A and B.

Proof If G is the monic GCRD of A and B, then G and $\mathrm{sres}_d(A, B)$ are similar over $\mathcal{F}$. Therefore, $\mathrm{sres}_d(A, B)$ is equal to $\mathrm{lc}(\mathrm{sres}_d(A, B))G$ because G is monic. □

The rest of this section is devoted to proving the theorem (Theorem 2.2.8) that describes the relation between the subresultant sequence of two Ore polynomials and that of their two left cofactors. This theorem will explain some experimental results in the next chapter. Chardin [5] proved this theorem when $\mathcal{R}[X]$ is a ring of differential operators. Johnson [19] proved this theorem when $\mathcal{R}[X]$ is a ring of algebraic polynomials. Our proof is inspired by Johnson's. First, we give two lemmas.

Lemma 2.2.6 If G is a non-zero polynomial in $\mathcal{F}[X]$, then $\mathrm{lc}(BG) = \mathrm{lc}(B)\sigma^n(\mathrm{lc}(G))$.

Proof Since $\mathrm{lc}(BG) = \mathrm{lc}(B)\mathrm{lc}(X^nG)$, the lemma follows from the extended Leibniz rule. □

Lemma 2.2.7 If G is a non-zero polynomial in $\mathcal{F}[X]$, then

$$\mathrm{prem}(AG, BG) = (\sigma^n\mathrm{lc}(G))^{[m-n+1]}\mathrm{prem}(A, B)G. \tag{2.4}$$

Proof By the pseudo-remainder formula (1.6) we have

$$\mathrm{lc}(B)^{[m-n+1]}A = PB + \mathrm{prem}(A, B) \tag{2.5}$$

and

$$\mathrm{lc}(BG)^{[m-n+1]}AG = QBG + \mathrm{prem}(AG, BG), \tag{2.6}$$

where P and Q belong to $\mathcal{F}[X]$. By (2.5) we obtain

$$\sigma^n(\mathrm{lc}(G))^{[m-n+1]}\mathrm{lc}(B)^{[m-n+1]}AG = \sigma^n(\mathrm{lc}(G))^{[m-n+1]}PBG + \sigma^n(\mathrm{lc}(G))^{[m-n+1]}\mathrm{prem}(A, B)G,$$

so

$$\mathrm{lc}(BG)^{[m-n+1]}AG = \sigma^n(\mathrm{lc}(G))^{[m-n+1]}PBG + \sigma^n(\mathrm{lc}(G))^{[m-n+1]}\mathrm{prem}(A, B)G$$

by Lemma 2.2.6. Comparing this equation and (2.6) yields (2.4), because the pseudo-remainder of AG and BG is unique. □

Theorem 2.2.8 *If G is a non-zero polynomial in $\mathcal{F}[X]$, with degree k, then*

$$\mathrm{sres}_{k+i}(AG, BG) = \left(\sigma^{i+1}\mathrm{lc}(G)\right)^{[m+n-2i-1]}\mathrm{sres}_i(A, B)G \qquad (n-1 \geq i \geq 0).$$

Proof Denote $\mathrm{lc}(G)$ by g, $\mathrm{sres}_{k+i}(AG, BG)$ by S_{k+i}, and $\mathrm{sres}_i(A, B)$ by T_i, for $i = n-1,\ n-2,\ \ldots,\ 0$. Put $\alpha_{k+i} = \mathrm{lc}(S_{k+i})$, $\lambda_i = \mathrm{lc}(T_i)$, for $i = n,\ n-1,\ \ldots,\ 0$, $\beta_{n+k} = (\sigma\mathrm{lc}(S_{n+k}))^{[m-n]}$, $\mu_n = (\sigma\mathrm{lc}(T_n))^{[m-n]}$, $\beta_{i+k} = \sigma\mathrm{lc}(S_{k+i})$, and $\mu_i = \sigma\mathrm{lc}(T_i)$, for $i = n-1,\ n-2,\ \ldots,\ 0$. With the new notation, we need to show

$$S_{k+i} = (\sigma^{i+1}g)^{[m+n-2i-1]}T_iG, \qquad (n-1 \geq i \geq 0).$$

If the sequence $A, B, A_3, \ldots, A_l$ is a PRS of A and B, then the sequence AG, BG, A_3G, $\ldots$, A_lG is a PRS of AG and BG by Lemma 2.2.7. Hence, $\mathcal{S}(AG, BG)$ and $\mathcal{S}(A, B)$ have the same gap-structure by Theorem 1.4.2. In particular, $S_{k+i} = 0$ if and only if $T_i = 0$, for $n-1 \geq i \geq 0$. Accordingly, we need only prove that the theorem holds for non-zero subresultants.

Let $\deg T_{n-1} = r \geq 0$. First, we prove that the theorem holds for $i = n-1$ and $i = r$. The theorem holds for $i = n-1$ because of the following calculation:

$$\begin{aligned} S_{k+n-1} &= (-1)^{m-n+1}\mathrm{prem}(AG, BG) \quad \text{(by Lemma 1.3.5)} \\ &= (-1)^{m-n+1}(\sigma^n g)^{[m-n+1]}\mathrm{prem}(A, B)G \quad \text{(by Lemma 2.2.7)} \\ &= (\sigma^n g)^{[m-n+1]}T_{n-1}G \quad \text{(by Lemma 1.3.5).} \end{aligned} \tag{2.7}$$

By Theorem 1.4.2 we find

$$\beta_{k+n}^{[n-1-r]} S_{k+r} = \beta_{k+n-1}^{[n-1-r]} S_{k+n-1} \quad \text{and} \quad \mu_n^{[n-1-r]} T_r = \mu_{n-1}^{[n-1-r]} T_{n-1}.$$

Combining these two equations with equation (2.7) yields

$$\mu_{n-1}^{[n-1-r]} \beta_{k+n}^{[n-1-r]} S_{k+r} = \mu_n^{[n-1-r]} \beta_{k+n-1}^{[n-1-r]} (\sigma^n g)^{[m-n+1]} T_r G.$$

It remains to prove that

$$\left(\frac{\mu_n^{[n-1-r]}}{\beta_{k+n}^{[n-1-r]}} \right) \left(\frac{\beta_{k+n-1}^{[n-1-r]}}{\mu_{n-1}^{[n-1-r]}} \right) (\sigma^n g)^{[m-n+1]} = (\sigma^{r+1} g)^{[m+n-2r-1]}. \tag{2.8}$$

Denote by L the σ-factorial expression on the left-hand side of (2.8). Since

$$\frac{\beta_{k+n}}{\mu_n} = (\sigma^{n+1} g)^{[m-n]} \quad \text{and} \quad \frac{\beta_{k+n-1}}{\mu_{n-1}} = (\sigma^{n+1} g)^{[m-n+1]} (\sigma^{r+1} g)$$

by Lemma 2.2.6 and (2.7), we deduce

$$\begin{aligned} L &= \left(\frac{(\sigma^{n+1} g)^{[m-n+1]} (\sigma^{r+1} g)}{(\sigma^{n+1} g)^{[m-n]}} \right)^{[n-1-r]} (\sigma^n g)^{[m-n+1]} \\ &= \left((\sigma^{m+1} g)(\sigma^{r+1} g) \right)^{[n-1-r]} (\sigma^n g)^{[m-n+1]} \\ &= (\sigma^{r+1} g)^{[n-1-r]} (\sigma^n g)^{[m-n+1]} (\sigma^{m+1} g)^{[n-1-r]} \; = \; (\sigma^{r+1} g)^{[m+n-2r-1]}. \end{aligned}$$

This proves (2.8).

So far we have proved that the theorem holds for all i such that $n-1 \geq i \geq r$, because all the subresultants with orders between $n-1$ and r are all equal to zero. In particular, the theorem holds when $n = 1$. Our induction hypothesis is that the theorem holds when $\deg B < n$. Assume that $\deg B = n$. To complete the induction, we have to prove that

$$S_{k+i} = (\sigma^{i+1} g)^{[m+n-2i-1]} T_i G, \quad (r-1 \geq i \geq 0).$$

By equation (1.19) in Lemma 1.4.1, we have, for $r-1 \geq i \geq 0$,

$$\mathrm{lc}(BG)^{[r-i]} \left(\sigma \mathrm{lc}(BG) \right)^{[n-1-i][m-n]} S_{k+i} = \mathrm{sres}_{k+i}(BG, S_{k+n-1}) \tag{2.9}$$

and

$$\mathrm{lc}(B)^{[r-i]} \left(\sigma \mathrm{lc}(B) \right)^{[n-1-i][m-n]} T_i = \mathrm{sres}_i(B, T_{n-1}). \tag{2.10}$$

From equation (2.9) we deduce

$$
\begin{aligned}
\mathrm{lc}(BG)^{[r-i]}\,(\sigma \mathrm{lc}(BG))^{[n-1-i][m-n]}\,S_{k+i} &= \mathrm{sres}_{k+i}(BG, S_{k+n-1}) = \\
&= \mathrm{sres}_{k+i}\left(BG, (\sigma^n g)^{[m-n+1]} T_{n-1} G\right) \quad \text{(by (2.7))} \\
&= (\sigma^n g)^{[m-n+1][n-i]} \mathrm{sres}_{k+i}\,(BG, T_{n-1}G) \quad \text{(by Lemma 1.3.3)} \\
&= (\sigma^n g)^{[m-n+1][n-i]} (\sigma^{i+1} g)^{[n+r-2i-1]} \mathrm{sres}_{k+i}\,(B, T_{n-1})\,G \quad \text{(by the induction hypothesis)} \\
&= (\sigma^n g)^{[m-n+1][n-i]} (\sigma^{i+1} g)^{[n+r-2i-1]} \mathrm{lc}(B)^{[r-i]}\,(\sigma \mathrm{lc}(B))^{[n-1-i][m-n]}\,T_i G \quad \text{(by (2.10))}.
\end{aligned}
$$

It remains to prove

$$
\frac{(\sigma^n g)^{[m-n+1][n-i]} (\sigma^{i+1} g)^{[n+r-2i-1]} \mathrm{lc}(B)^{[r-i]}\,(\sigma \mathrm{lc}(B))^{[n-1-i][m-n]}}{\mathrm{lc}(BG)^{[r-i]}\,(\sigma \mathrm{lc}(BG))^{[n-1-i][m-n]}} = (\sigma^{i+1} g)^{[m+n-2i-1]}. \tag{2.11}
$$

Denote by L' the left-hand side of (2.11). By Lemma 2.2.6 it holds that

$$
L' = \frac{(\sigma^n g)^{[m-n+1][n-i]} (\sigma^{i+1} g)^{[n+r-2i-1]}}{(\sigma^n g)^{[r-i]} (\sigma^{n+1} g)^{[n-1-i][m-n]}}.
$$

The fourth assertion of Lemma 1.2.5 implies

$$
(\sigma^n g)^{[m-n+1][n-i]} = (\sigma^n g)^{[m-i]} (\sigma^{n+1} g)^{[m-n][n-i-1]},
$$

from which it follows that

$$
L' = \left(\frac{(\sigma^n g)^{[m-i]}}{(\sigma^n g)^{[r-i]}}\right) (\sigma^{i+1} g)^{[n+r-2i-1]} = (\sigma^{i+1} g)^{[n+r-2i-1]} (\sigma^{[n+r-i]} g)^{[m-r]} = (\sigma^{i+1} g)^{[m+n-2i-1]}.
$$

Equation (2.11) is proved. □

This theorem reveals that, if one uses the subresultant algorithm to compute the GCRD of two Ore polynomials, one gets the subresultant sequence of the first kind of the two left cofactors as a byproduct.

Corollary 2.2.9 With the notation introduced in Theorem 2.2.8, we have

$$
\mathrm{lc}(\mathrm{sres}_k(AG, BG)) X^0 = \mathrm{lc}(G)^{[m+n]} \mathrm{res}(A, B). \tag{2.12}
$$

Proof Setting $i = 0$ in Theorem 2.2.8, we get $\mathrm{sres}_k(AG, BG) = (\sigma g)^{[m+n-1]} \mathrm{res}(A, B) G$. Equating the leading coefficients of both sides of this equation yields (2.12). □

2.3 Least Common Left Multiples

Throughout this section the quotient field of $\mathcal{R}$ is denoted by $\mathcal{F}$ and $(\mathcal{F}[X], \Theta, \sigma, \delta)$ is denoted by $\mathcal{F}[X]$. We assume that the degree of the GCRDs of A and B, denoted by d, is less than n.

Definition 2.3.1 A non-zero polynomial in $\mathcal{F}[X]$ of lowest degree, which is right-hand divisible by A and B, is called an *LCLM* of A and B.

Obviously, two LCLMs of A and B are similar over $\mathcal{F}$. Ore [33] proved the existence of LCLMs using the Euclidean algorithm. As a convenience for later references, we state his theorem [33, Theorem 8] in terms of polynomial remainder sequences.

Theorem 2.3.1 *With $A = A_1$ and $B = A_2$, assume that the sequence*

$$A_1, A_2, A_3, \ldots, A_k$$

is a PRS of A and B. Then the polynomial

$$L = A_{k-1}A_k^{-1}A_{k-2}A_{k-1}^{-1} \cdots A_3A_4^{-1}A_2A_3^{-1}A_1 \tag{2.13}$$

is an LCLM of A and B.

Ore proved that L in (2.13) was a well-defined polynomial and an LCLM of A and B. The proof of the following corollary can also be found in [33, p. 486].

Corollary 2.3.2 If L is an LCLM of A and B, then

$$\deg A + \deg B = d + \deg L.$$

Another way to compute LCLMs of A and B is to use the extended Euclidean algorithm to find U and V in $\mathcal{F}[X]$ such that $\deg U = \deg(B) - d$, $\deg V = \deg(A) - d$, and $UA + VB = 0$. Then both UA and VB are LCLMs of A and B by Corollary 2.3.2.

We shall now present a determinant formula for LCLMs. Let $S_j = \text{sres}_j(A, B)$, for $j = n - 1$, $n - 2$, ..., -1. By Remark 1.3.3 we have

$$S_j = u_{n-j-1}X^{n-j-1}A + \cdots + u_1XA + u_0A + v_{m-j-1}X^{m-j-1}B + \cdots + v_1XB + v_0B,$$

where each of the u's and v's belongs to $\mathcal{R}$. Using multiplication in $\mathcal{F}[X]$ we write

$$U_jA + V_jB = S_j, \tag{2.14}$$

where $U_j = u_{n-j-1}X^{n-j-1} + \cdots + u_1X + u_0$ and $V_j = v_{m-j-1}X^{m-j-1} + \cdots + v_1X + v_0$. The polynomials U_j and V_j can be expressed by replacing the last column in the determinant of S_j (see Remark 1.3.3 and Lemma 1.3.1) by the transposes of

$$(X^{n-j-1}, X^{n-j-2}, \ldots, X^0, \underbrace{0, 0, \ldots, 0}_{m-j})$$

and

$$(\underbrace{0, 0, \ldots, 0}_{n-j}, X^{m-j-1}, X^{m-j-2}, \ldots, X^0),$$

respectively.

Proposition 2.3.3 Both $U_{d-1}A$ and $V_{d-1}B$ are LCLMs of A and B.

Proof Since S_d is a GCRD of A and B by Proposition 2.2.3, we see that $S_{d-1} = 0$. Therefore, $U_{d-1}A + V_{d-1}B = 0$ by (2.14). Since the coefficient of X^{n-d} in U_{d-1} is nonzero by (2.15) and Proposition 2.2.3, $\deg U_{d-1} = n - d$, consequently, $\deg U_{d-1}A = m + n - d$. Thus $U_{d-1}A$ is an LCLM of A and B by Corollary 2.3.2. □

In the next chapter we show that the GCRDs of A and B can be obtained without computing any PRS of A and B when $\mathcal{R}$ is $\mathbf{Z}[t]$. In this case we need only to expand a determinant to compute LCLMs. As the leading coefficient of an LCLM is particularly important for proving identities of holonomic functions, we give a multiplicative bound for the denominators of the coefficients in the monic LCLM of A and B.

Note that, for $j = n-2, n-3,, \ldots, -1$,

$$u_{n-j-1} = (-1)^{m-j}\sigma^{m-j-1}(\mathrm{lc}(B))\mathrm{coeff}(S_{j+1}, X^{j+1}), \tag{2.15}$$

where $\mathrm{coeff}(S_{j+1}, X^{j+1})$ stands for the coefficient of X^{j+1} in S_{j+1}.

Corollary 2.3.4 If L is the monic LCLM of A and B, then $bL \in \mathcal{R}[X]$, where

$$b = \left(\sigma^{m-d}\mathrm{lc}(B)\right)\left(\sigma^{n-d}\mathrm{lc}(A)\right)\mathrm{lc}(\mathrm{sres}_d(A, B)).$$

Proof By (2.15) we have

$$\mathrm{lc}(U_{d-1}A) = (-1)^{m-j}\left(\sigma^{m-d}\mathrm{lc}(B)\right)\left(\sigma^{n-d}\mathrm{lc}(A)\right)\mathrm{lc}(\mathrm{sres}_d(A, B))$$

because $\deg U_{d-1} = n - d$. The lemma then follows from the fact that any two LCLMs of A and B are similar over $\mathcal{F}$. □

In the rest of this section, we present an algorithm for computing $U_{d-1}A$, Let $d_i = \deg A_i$, where A_i is given in Theorem 2.3.1, for $i = 1, 2, \ldots, k$. Theorem 1.4.7 and Corollary 1.4.4 imply that $\mathcal{S}_1(A, B)$ consists of

$$A, B, S_{d_2-1}, S_{d_3-1}, \ldots, S_{d_{k-1}-1}$$

and that $\mathcal{S}_2(A, B)$ consists of

$$A, B, S_{d_3}, S_{d_4}, \ldots, S_{d_k}.$$

Lemma 2.3.5 For all i with $2 \le i \le k$, $\deg U_{d_i-1} = n - d_i$ and $\deg V_{d_i-1} = m - d_i$, where U_{d_i-1} and V_{d_i-1} are defined in (2.14).

Proof It follows from (2.15) and the fact that S_{d_i} is regular. □

Theorem 2.3.6 *For $i = 2, 3, \ldots, k-1$, let a_{i+1} be $\mathrm{lc}(S_{d_i-1})$ and U_{d_i-1} be the same as those defined by (2.14). Let $e_3, e_4, \ldots, e_k$ form the sequence satisfying*

$$\begin{aligned} e_3 S_{d_2-1} &= \mathrm{prem}(A, B), && (2.16)\\ e_4 S_{d_3-1} &= \mathrm{prem}(B, S_{d_2-1}), && (2.17)\\ e_i S_{d_{i-1}-1} &= \mathrm{prem}(S_{d_{i-3}-1}, S_{d_{i-2}-1}), && (2.18)\\ & \quad \text{for } i = 5, 6, \ldots, k, \end{aligned}$$

then

$$\begin{aligned} U_{d_2-1} &= (-\mathrm{lc}(B))^{[m-n+1]} && (2.19)\\ U_{d_3-1} &= -e_4^{-1} Q_4 U_{d_2-1} && (2.20)\\ U_{d_{i-1}-1} &= e_i^{-1}\left(a_{i-1}^{[d_{i-2}-d_{i-1}+1]} U_{d_{i-3}-1} - Q_i U_{d_{i-2}-1}\right), && (2.21)\\ & \quad \text{for } i = 5, 6, \ldots, k \end{aligned}$$

where Q_4 is the left pseudo-quotient of B and S_{d_2-1}, and each of the Q_i's is the left pseudo-quotient of $S_{d_{i-3}-1}$ and $S_{d_{i-2}-1}$. Furthermore, if $H_k = a_k^{[d_{k-1}-d_k+1]} U_{d_{k-2}-1} - Q_{k+1} U_{d_{k-1}-1}$, where Q_{k+1} is the left pseudo-quotient of $S_{d_{k-2}-1}$ and $S_{d_{k-1}-1}$, then

$$U_{d_k-1} = (-1)^{m-d_k+1} \mathrm{lc}(S_{d_k}) \left(\sigma^{m-d_k} \mathrm{lc}(B)\right) \mathrm{lc}(H_k)^{-1} H_k. \quad (2.22)$$

Proof We say that two Ore polynomials F and G are congruent modulo an Ore polynomial M on the right if $F - G$ is right-divisible by M, which is denoted by $F \equiv G \,(\mathrm{mod}\, M)$.

The equality (2.19) holds by the definition of U_{d_2-1}. Assume that $k>3$. By (2.16) and (2.17) we have

$$U_{d_2-1}A \equiv S_{d_2-1} \ (\operatorname{mod} B) \quad \text{and} \quad Q_4 S_{d_2-1} \equiv -e_4 S_{d_3-1} \ (\operatorname{mod} B).$$

Combining the two equations just derived yields $-e_4^{-1}Q_4U_{d_2-1}A \equiv S_{d_3-1} \ (\operatorname{mod} B)$. On the other hand, the definition of U_{d_3-1} implies that $U_{d_3-1}A \equiv S_{d_3-1} \ (\operatorname{mod} B)$. It follows that

$$C_4 A \equiv 0 \quad (\operatorname{mod} B),$$

where $C_4 = -e_4^{-1}Q_4U_{d_2-1} - U_{d_3-1}$. But $C_4 = 0$, otherwise C_4A would be a non-zero left common multiple of A and B, with degree $\leq (m+n-d_3)$, which contradicts Corollary 2.3.2. Equation (2.20) holds.

Consider the case in which $k>4$. For $5 \leq i \leq k$, the congruent relations

$$U_{d_{i-3}-1}A \equiv S_{d_{i-3}-1} \ (\operatorname{mod} B) \quad \text{and} \quad U_{d_{i-2}-1}A \equiv S_{d_{i-2}-1} \ (\operatorname{mod} B).$$

hold by (2.14). Furthermore, equation (2.18) can be written as

$$a_{i-1}^{[d_{i-2}-d_{i-1}+1]} S_{d_{i-3}-1} = Q_i S_{d_{i-2}-1} + e_i S_{d_{i-1}-1}.$$

Combining the two congruent equations and the equation just given yields

$$e_i^{-1}\left(a_{i-1}^{[d_{i-2}-d_{i-1}+1]} U_{d_{i-3}-1} - Q_i U_{d_{i-2}-1}\right) A \equiv S_{d_{i-1}-1} \ (\operatorname{mod} B).$$

The congruence just proved and $U_{d_{i-1}-1}A \equiv S_{d_{i-1}-1} \ (\operatorname{mod} B)$ imply that

$$C_i A \equiv 0 \ (\operatorname{mod} B),$$

where $C_i = U_{d_{i-1}-1} - e_i^{-1}\left(a_{i-1}^{[d_{i-2}-d_{i-1}+1]} U_{d_{i-3}-1} - Q_i U_{d_{i-2}-1}\right)$. Since

$$\deg U_{d_{i-1}-1} = \deg\left(a_{i-1}^{[d_{i-2}-d_{i-1}+1]} U_{d_{i-3}-1} - Q_i U_{d_{i-2}-1}\right) = n - d_{i-1}$$

by Lemma 2.3.5, we have $C_i = 0$, otherwise the degree of the LCLMs of A and B would be not greater than $(n-d_{i-1})$, which contradicts to Corollary 2.3.2. Equation (2.21) holds.

To prove (2.22), we let $C_k = (-1)^{m-d_k+1}\left(\sigma^{m-d_k}\mathrm{lc}(B)\right)\mathrm{lc}(S_{d_{k-1}-1})\mathrm{lc}(H_k)^{-1}H_k$. Observe that

$$a_k^{[d_{k-1}-d_k+1]} S_{d_{k-2}-1} = Q_{k+1} S_{d_{k-1}-1},$$

because $S_{d_{k-1}-1}$ is the last member of $\mathcal{S}_1(A,B)$. It then follows from the congruent equations

$$U_{d_{k-2}-1}A \equiv S_{d_{k-2}-1} \ (\operatorname{mod} B) \quad \text{and} \quad U_{d_{k-1}-1}A \equiv S_{d_{k-2}-1} \ (\operatorname{mod} B)$$

that $H_k A \equiv 0 \pmod{B}$. The degree of H_k is equal to $(n - d_k)$ since

$$\deg(Q_{k+1} U_{d_{k-1}-1}) = (d_{k-1} - d_k) + (n - d_{k-1}) = n - d_k \quad \text{and} \quad \deg U_{d_{k-2}-1} = n - d_{k-2}.$$

Accordingly, the product $H_k A$ is an LCLM of A and B. Notice that $\mathrm{lc}(C_k) = \mathrm{lc}(U_{d_k-1})$. Hence C_k and U_{d_k-1} are equal because $C_k A$ and $U_{d_k-1} A$ are similar over $\mathcal{F}$. $\square$

As the e_i's in Theorem 2.3.6 can be constructed by the subresultant algorithm, we perform the subresultant algorithm, and record both left pseudo-quotient Q_i and extraneous factor e_i after each pseudo-division. Compute U_{d_i-1}, by (2.19), (2.20) and (2.21). Ultimately, we get both $U_{d_{k-2}-1}$ and $U_{d_{k-1}-1}$. The leading coefficient of S_{d_k} can be obtained from the formula given in Lemma 1.4.6, so U_{d_k-1} is computed by (2.22).

Chapter 3

Modular Algorithm for Computing GCRDs over $\mathbf{Z}[t]$

[1] Recent years have seen a rapid development of the algorithms for manipulating the functions that are annihilated by linear operational polynomials [42, 38, 2, 6]. This development motivates us to design an efficient algorithm for computing GCRDs over $\mathbf{Z}[t]$. The GCRD-calculation plays an important role in the computation of linear operational polynomials. For instance, if L_1 and L_2 are two linear differential operators, then their GCRD corresponds to the intersection of the solution spaces of L_1 and L_2. To represent the sum of the two solution spaces, one needs an LCLM of L_1 and L_2, which can be expressed as a determinant with entries being the derivatives of coefficients of L_1 and L_2, as long as the GCRD is obtained. The greatest common left divisor of L_1 and L_2 can be obtained from the GCRD of their adjoint operators.

We will extend the techniques used in the modular algorithm for computing usual commutative polynomial GCDs as much as we can (see Brown [3] and Geddes et al [16]). Two new problems that cannot be tackled by the classical techniques, are that

- evaluation mappings are *not* Ore ring homomorphisms;
- the normalization of leading coefficients is different from that in the algebraic case.

The first problem will be solved by the subresultant theory for Ore polynomials; the second one by rational number and rational function reconstructions. To the author's knowledge the present

[1]This chapter reports joint work with István Nemes.

algorithm is the first modular algorithm for computing Ore polynomial GCRDs. The non-modular algorithms are the Euclidean algorithm [33] and subresultant algorithm. Grigor'ev [18] presents a method for computing the GCRD of several linear differential operators by Gaussian elimination.

We will work in Ore polynomial rings whose coefficient domains are algebraic polynomial rings. Throughout this section, p is a prime and $\mathbf{Z}_p$ is the Galois field with p elements. For an indeterminate t, $\mathbf{Z}[t]$ and $\mathbf{Z}_p[t]$ are the rings of algebraic polynomials in t over $\mathbf{Z}$ and $\mathbf{Z}_p$, respectively. Let X be a new indeterminate. For non-zero F in $\mathbf{Z}[t][X]$ or $\mathbf{Z}_p[t][X]$, the leading coefficient of F in X is denoted by $\mathrm{lc}(F)$, the leading coefficient of $\mathrm{lc}(F)$ in t is called the *head coefficient* of F and denoted by $\mathrm{hc}(F)$, the degree of F in X is denoted by $\deg F$, and the degree of F in t is denoted by $\deg_t F$.

We assume that $(\mathbf{Z}[t][X], \sigma, \delta)$ is an Ore polynomial ring over $\mathbf{Z}[t]$. For brevity we denote this ring by $\mathbf{Z}[t][X]$. If A and B are in $\mathbf{Z}[t][X]$, then the *normalized* GCRD of A and B is the GCRD of A and B, which is in $\mathbf{Z}[t][X]$ and primitive with respect to X, and has positive head coefficient. If A and B are in the Ore polynomial ring $\mathbf{Z}_p[t][X]$, then the *normalized* GCRD of A and B is the GCRD of A and B, which is in $\mathbf{Z}_p[t][X]$ and primitive with respect to X, and has head coefficient 1. The normalized GCRD of A and B, where A and B are in $\mathbf{Z}[t][X]$ or $\mathbf{Z}_p[t][X]$, is denoted by $\mathrm{GCRD}(A, B)$.

The idea of our algorithm is as follows.

1. Use sufficiently many modular homomorphisms to reduce GCRD problem in $\mathbf{Z}[t][X]$ to a series of GCRD problems in $\mathbf{Z}_p[t][X]$.

2. Use sufficiently many evaluation mappings to reduce GCRD problem in $\mathbf{Z}_p[t][X]$ to a series of the problems of finding evaluation homomorphic images of the monic associate of the sought-after GCRD.

3. Use Chinese Remainder Algorithm (CRA) and rational function reconstruction to combine the lucky evaluation homomorphic images.

4. Use CRA and rational number reconstruction to combine the lucky modular homomorphic images.

This chapter is organized as follows. In Section 3.1, we study the modular and evaluation mappings. Section 3.2 is devoted to presenting the algorithm for computing the evaluation homomorphic images of the monic associate of the GCRD of two Ore polynomials in $\mathbf{Z}_p[t][X]$. In Section 3.3, we review the rational number and function reconstructions. The modular algorithms for computing GCRDs in $\mathbf{Z}_p[t][X]$ and in $\mathbf{Z}[t][X]$ are described in Section 3.4 and Section 3.5, respectively. Experimental results are given in Section 3.6

3.1 Modular Mappings and Evaluation Mappings

A modular mapping ϕ_p from $\mathbf{Z}[t][X]$ to $\mathbf{Z}_p[t][X]$ is a module homomorphism defined for a prime p by $\phi_p(A) = A \bmod p$, for $A \in \mathbf{Z}[t][X]$. An evaluation mapping ϕ_{t-k} from $\mathbf{Z}_p[t][X]$ to $\mathbf{Z}_p[X]$ is a module homomorphism defined for an element k of $\mathbf{Z}_p$ by $\phi_{t-k}(A(t,X)) = A(k,X)$, for $A \in \mathbf{Z}_p[t][X]$. In this section, we investigate whether modular and evaluation mappings can be regarded as Ore ring homomorphisms.

The next lemma clearly holds because σ and δ are endomorphisms of the additive group $\mathbf{Z}[t]$,

Lemma 3.1.1 If $f, g \in \mathbf{Z}[t]$ and $f \equiv g \bmod p$, then $\sigma(f) \equiv \sigma(g) \bmod p$ and $\delta(f) \equiv \delta(g) \bmod p$. $\square$

This lemma allows us to define two operators σ_p and δ_p on $\mathbf{Z}_p[t]$ by the respective rules:

$$\sigma_p(\phi_p(f)) = \phi_p(\sigma(f)) \quad \text{and} \quad \delta_p(\phi_p(f)) = \phi_p(\delta(f)), \quad \text{for all } f \in \mathbf{Z}[t].$$

It is clear that σ_p is an endomorphism of the domain $\mathbf{Z}_p[t]$ and that δ_p is a pseudo-derivation with respect to σ_p if σ_p is injective.

Lemma 3.1.2 If p is not a divisor of $\text{hc}(\sigma(t))$, then σ_p is injective.

Proof Since $\sigma(m) = m$, for $m \in \mathbf{Z}$, $\deg_t \sigma(t) > 0$. Let $f = f_n t^n + \cdots + f_0 \in \mathbf{Z}[t]$. If $\sigma_p(\phi_p(f)) = 0$, then $\phi_p\left(f_n \sigma(t)^n + \cdots + f_0\right) = 0$ by the definition of σ_p. Since $\phi_p\left(\text{hc}(\sigma(t))\right) \neq 0$, $\phi_p(\sigma(t))$ is of positive degree in t, and hence $\phi_p(f_i) = 0$, $0 \leq i \leq n$. $\square$

The above two lemmas assert that σ_p is a conjugate operator and δ_p is a pseudo-derivation with respect to σ_p if p does not divide $\text{hc}(\sigma(t))$. Thus, $(\mathbf{Z}_p[t][X], \sigma_p, \delta_p)$ is an Ore polynomial ring with the multiplication defined by $Xa = \sigma_p(a)X + \delta_p(a)$, for all $a \in \mathbf{Z}_p[t]$ by Proposition 1.2.1 and Theorem 1.2.2.

When σ_p is injective, the Ore polynomial ring $(\mathbf{Z}_p[t][X], \sigma_p, \delta_p)$ is said to be the *Ore polynomial ring induced from* $\mathbf{Z}[t][X]$ *by the modular homomorphism* ϕ_p. The next corollary is evident.

Corollary 3.1.3 If $(\mathbf{Z}_p[t][X], \sigma_p, \delta_p)$ is the Ore polynomial ring induced from $\mathbf{Z}[t][X]$ by the modular mapping ϕ_p, then ϕ_p is an Ore polynomial ring homomorphism from $\mathbf{Z}[t][X]$ to $\mathbf{Z}_p[t][X]$.

If $(\mathbf{Z}_p[X], \sigma', \delta')$ is an Ore polynomial ring, then σ' must be the identity mapping and δ' must be the null mapping since $\mathbf{Z}_p$ is generated by 1 as an additive group. Accordingly, the multiplication induced by σ' and δ' is the usual commutative one. Therefore, an evaluation mapping from an

Ore polynomial ring $\mathbf{Z}_p[t][X]$ to $\mathbf{Z}_p[X]$ is *not* always an Ore ring homomorphism. This fact tells us that the algebraic modular method in [3] cannot be directly applied to Ore polynomials. We will overcome this difficulty by Propositions 2.2.3 and 2.2.4.

3.2 Evaluation Homomorphic Images of GCRDs

In this section, let $(\mathbf{Z}_p[t][X], \sigma_p, \delta_p)$ be an Ore polynomial ring. Fix an element k of $\mathbf{Z}_p$ and the evaluation mapping ϕ_{t-k}. Assume that A and B are in $\mathbf{Z}_p[t][X]$, with $\deg A = m$ and $\deg B = n$, where $m \geq n \geq 1$. Let M be the matrix $\mathrm{mat}(X^{n-1}A, \ldots, XA,\, A,\, X^{m-1}B, \ldots, XB,\, B)$. We show how to use the arithmetic in $\mathbf{Z}_p$ to compute the monic associate of $\phi_{t-k}(\mathrm{GCRD}(A,B))$.

Lemma 3.2.1 Let G be $\mathrm{GCRD}(A,B)$ with degree d and let S_d be $\mathrm{sres}_d(A,B)$. If $\phi_{t-k}(\mathrm{lc}(S_d))$ is nonzero, then

$$\frac{\phi_{t-k}(G)}{\phi_{t-k}(\mathrm{lc}(G))} = \frac{\phi_{t-k}(S_d)}{\phi_{t-k}(\mathrm{lc}(S_d))}.$$

Proof By Proposition 2.2.3 there exists a non-zero r in $\mathbf{Z}_p[t]$ such that $rG = S_d$. Since $\phi_{t-k}(\mathrm{lc}(S_d))$ is nonzero, $\phi_{t-k}(\mathrm{lc}(G))$ is nonzero. Applying ϕ_{t-k} to

$$\frac{G}{\mathrm{lc}(G)} = \frac{S_d}{\mathrm{lc}(S_d)}$$

yields the lemma. $\square$

Definition 3.2.1 Let d be the degree of $\mathrm{GCRD}(A,B)$. The evaluation point k is *unlucky* for A and B if either

$$\phi_{t-k}\left(\prod_{i=0}^{m-1}\left(\sigma_p^i \mathrm{lc}(B)\right)\right) = 0 \quad \text{or} \quad \phi_{t-k}(\mathrm{lc}(\mathrm{sres}_d(A,B))) = 0.$$

One way to compute the image of the monic associate of $\mathrm{GCRD}(A,B)$ under ϕ_{t-k} is as follows. We compute the image of M under ϕ_{t-k}, denoted by M_{t-k}, and compute the rank of M_{t-k}. If k is not unlucky, then M and M_{t-k} have the same rank, so the degree of $\mathrm{GCRD}(A,B)$, say, d, is equal to $(m+n-\mathrm{rank}(M_{t-k}))$ by Proposition 2.2.3. Thus, the monic associate of $\phi_{t-k}(\mathrm{sres}_d(A,B))$ is the monic associate of $\phi_{t-k}(\mathrm{GCRD}(A,B))$ by Lemma 3.2.1. This method has two computational tasks, namely, calculating $\mathrm{rank}(M_{t-k})$ and $\phi_{t-k}(\mathrm{sres}_d(A,B))$. These two tasks can be combined into one Gaussian elimination when the pivot rows are chosen properly. These considerations lead to the algorithm GCRD_e.

algorithm GCRD_e

Input: A prime p, a residue $k \in \mathbf{Z}_p$, and $A, B \in \mathbf{Z}_p[t][X]$ with $\deg A \geq \deg B \geq 0$.

Output: $g \in \mathbf{Z}_p[X]$. If k is not unlucky, then g is the monic associate of $\phi_{t-k}(\mathrm{GCRD}(A, B))$. Otherwise, g is 0 or of degree greater than d.

[initialize]

1. $m \leftarrow \deg A$; $n \leftarrow \deg B$;

2. **for** $i = 0$ **to** $m-1$ **do** **{** $C_i \leftarrow \phi_{t-k}(X^i B)$; **if** $\deg C_i < n + i$ **then** **return**(0); [unlucky k] **}**

[nested elimination]

3. **for** $i = 0$ **to** $n - 1$ **do** **{**

4. $R_i \leftarrow \phi_{t-k}(X^i A)$;

5. **for** $j = m - n + i$ **to** 0 **do** **{** **if** $\deg R_i = \deg C_j$ **then** $R_i \leftarrow R_i - \mathrm{lc}(R_i)\mathrm{lc}(C_j)^{-1} C_j$; **}**

6. **while** $\exists\, l$, $0 \leq l \leq i - 1$ and $\deg R_i = \deg R_l \geq 0$ **do** **{** $R_i \leftarrow R_i - \mathrm{lc}(R_i)\mathrm{lc}(R_l)^{-1} R_l$; **}** **}**

[compute the rank]

7. $r \leftarrow m + n$;

8. **for** $i = 0$ **to** $n - 1$ **do** **{** **if** $R_i = 0$ **then** $r \leftarrow r - 1$; **}**

[guess the degree]

9. $b \leftarrow m + n - r$;

[compute the image]

10. $g \leftarrow$ the polynomial of least degree in the set $\{R_0, R_1, \ldots, R_{n-b-1}, C_0\}$;

11. **if** $\deg g = b$ **then** $g \leftarrow \mathrm{lc}(g)^{-1} g$; **else** $g \leftarrow 0$;

12. **return**(g);

Proposition 3.2.2 The algorithm GCRD_e is correct.

Proof Let $d = \deg \mathrm{GCRD}(A, B)$ and $S_d = \mathrm{sres}_d(A, B)$. We exclude the case when $\phi_{t-k}(\sigma^i \mathrm{lc}(B))$ is zero, for some i with $0 \leq i \leq m - 1$. Thus, $\deg C_i = n + i$, for $i = 0, 1, \ldots, m - 1$. According to lines 5 and 6, all of the non-zero R_j's have distinct degrees less than n, for $j = 0, 1, \ldots, n - 1$. Hence r obtained from line 8 is the rank of the matrix

$$N_{t-k} = \mathrm{mat}(R_{n-1}, \ldots, R_0, C_{m-1}, \ldots, C_0).$$

Let

$$M_{t-k} = \mathrm{mat}(\phi_{t-k}(X^{n-1}A), \ldots, \phi_{t-k}(A), \phi_{t-k}(X^{m-1}B), \ldots, \phi_{t-k}(B)).$$

Then $r = \text{rank}(M_{t-k})$ because N_{t-k} is computed by row reduction on M_{t-k} in lines 3, 4, 5 and 6. Note that $r \leq \text{rank}(M)$. Consequently, the tentative degree b obtained from line 9 is not less than d by Proposition 2.2.4. Notice that the polynomial g obtained from line 10 is the polynomial with smallest degree among the polynomials

$$R_{n-b-1}, \ldots, R_0, C_{m-b-1}, \ldots, C_0,$$

all of which are $\mathbf{Z}_p$-linear combinations of

$$\phi_{t-k}(X^{n-b-1}A), \ldots, \phi_{t-k}(A), \phi_{t-k}(X^{m-b-1}B), \ldots, \phi_{t-k}(B),$$

and *vice versa*. Therefore, g and $\phi_{t-k}(\text{sres}_b(A, B))$ are similar over $\mathbf{Z}_p$ by Lemma 1.3.4.

If k is not unlucky, then the polynomials

$$\phi_{t-k}(X^{n-d-1}A), \ldots, \phi_{t-k}(A), \phi_{t-k}(X^{m-d-1}B), \ldots, \phi_{t-k}(B)$$

are $\mathbf{Z}_p$-linearly independent, because $\phi_{t-k}(\text{lc}(S_d))$ is nonzero. Accordingly, the polynomials

$$\phi_{t-k}(X^{n-d-1}A), \ldots, \phi_{t-k}(A), \phi_{t-k}(X^{m-1}B), \ldots, \phi_{t-k}(X^{m-d-1}B), \ldots, \phi_{t-k}(B)$$

are $\mathbf{Z}_p$-linearly independent. Hence $r \geq m + n - d$. But $\text{rank}(M) = m + n - d$ by Proposition 2.2.4. Consequently, $r = \text{rank}(M)$, so $b = d$. Since g and $\phi_{t-k}(S_d)$ are similar over $\mathbf{Z}_p$, g returned in line 12 is the monic associate of $\phi_{t-k}(\text{GCRD}(A, B))$ by Lemma 3.2.1.

If k is unlucky, then there are two cases, namely, $b > d$ or $b = d$. In the former case, g is either 0 or a polynomial of degree greater than d. In the latter case, $\deg g < b$ since g and $\phi_{t-k}(S_d)$ are similar over $\mathbf{Z}_p$, and $\deg \phi_{t-k}(S_d) < d$. Therefore, g is set to be 0 in line 11. □

3.3 Rational Number and Rational Function Reconstructions

To use CRA to combine the evaluation homomorphic images of the monic GCRD of two Ore polynomials, say, A and B in $\mathbf{Z}_p[t][X]$, we need to know a multiplicative bound for the denominator of the monic GCRD of A and B. In the algebraic case, such a bound is the GCD of $\text{lc}(A)$ and $\text{lc}(B)$. However, there are counterexamples showing that neither the GCD nor the LCM of $\text{lc}(A)$ and $\text{lc}(B)$ is the desired multiplicative bound. One multiplicative bound is the leading coefficient of the dth subresultant of A and B if $\text{GCRD}(A, B)$ has degree d by Corollary 2.2.5. Unfortunately, this multiplicative bound tends to be loose. Inspired by the work of Encarnación [14, 15], we use rational function reconstruction to combine the evaluation homomorphic images of GCRD(A, B). A similar problem arises when A and B are in $\mathbf{Z}[t][X]$. Thus, the rational number reconstruction is also needed.

The algorithm for reconstructing rational numbers [40] is given below.

algorithm RECON_n

Input: A modulus $m \in \mathbf{N}^+$ and a non-zero residue $r \in \mathbf{Z}_m = \{0, 1, \ldots, m-1\}$.

Output: A pair (a, b) of integers, s.t. $ab^{-1} = r$ in $\mathbf{Z}_m$, $|a| < \sqrt{m/2}$, and $0 < b < \sqrt{m/2}$ if such a and b exist. Otherwise, NIL is returned.

1. $a_1 \leftarrow m$; $a_2 \leftarrow r$; $v_1 \leftarrow 0$; $v_2 \leftarrow 1$; $i \leftarrow 2$;
2. **while** true **do** **{**
3. **if** $|v_i| \geq \sqrt{m/2}$ **then return**(NIL);
4. **if** $|a_i| < \sqrt{m/2}$ and GCD$(a_i, v_i) = 1$ **then return** $((\mathrm{sign}(v_i)a_i, \mid v_i \mid))$;
5. $q \leftarrow$ integral quotient of a_{i-1} and a_i;
6. $a_{i+1} \leftarrow a_{i-1} - qa_i$; $v_{i+1} \leftarrow v_{i-1} - qv_i$; $i \leftarrow i + 1$; **}**

According to [12], we added the condition GCD$(a_i, v_i) = 1$ in line 4 in RECON_n, because Wang's original algorithm does not guarantee that GCD$(b, m) = 1$. The reader is advised to consult [12] for more detailed discussion and recent progress on rational number reconstruction.

In the library of the computer algebra system *Maple*, there is an implementation solving the general problem of rational function reconstruction. As we could not find any proof of the correctness of this implementation in the literature, we present the problem of rational function reconstruction and a modified version of the algorithm, named RECON_f, for our use,

We are concerned with the following problem.

Problem RFR: Let $\mathcal{F}$ be a field and $\mathcal{F}[t]$ the algebraic polynomial ring over $\mathcal{F}$.

Given: $M \in \mathcal{F}[t]$ with $\deg_t M > 0$, and a non-zero $R \in \mathcal{F}[t]/(M)$.

Find: $A, B \in \mathcal{F}[t]$ with $\deg_t A \leq (\deg_t M)/2$, $\deg_t B < (\deg_t M)/2$, and GCD$(B, M) = 1$ such that $AB^{-1} = R$ in $\mathcal{F}[t]/(M)$.

The algorithm RECON_f solves Problem RFR.

algorithm RECON_f

Input: A modulus $M \in \mathcal{F}[t]$ and a non-zero residue $R \in \mathcal{F}[t]/(M)$.

Output: A pair (A, B) of polynomials in $\mathcal{F}[t]$, such that $AB^{-1} = R$ in $\mathcal{F}[t]/(M)$, $\mathrm{lc}(B) = 1$, $\deg_t A \le (\deg_t M)/2$, and $\deg_t B < (\deg_t M)/2$ if such A and B exist. Otherwise, NIL is returned.

1. $A_1 \leftarrow M$; $A_2 \leftarrow R$; $V_1 \leftarrow 0$; $V_2 \leftarrow 1$; $i \leftarrow 2$;
2. **while** true **do**
3. **if** $\deg_t V_i \ge (\deg_t M)/2$ **then return**(NIL);
4. **if** $\deg_t A_i \le (\deg_t M)/2$ and $\mathrm{GCD}(A_i, V_i) = 1$ **then return** $(((\mathrm{lc}_t V_i)^{-1} A_i,\ (\mathrm{lc}_t V_i)^{-1} V_i))$;
5. $Q \leftarrow$ polynomial quotient of A_{i-1} and A_i;
6. $A_{i+1} \leftarrow A_{i-1} - QA_i$; $V_{i+1} \leftarrow V_{i-1} - qV_i$; $i \leftarrow i + 1$;

Now, we prove the correctness of RECON_f.

Lemma 3.3.1 If (A, B) is a solution to Problem RFR, then the fraction A/B is uniquely determined, and the pair $(A/\mathrm{GCD}(A, B), B/\mathrm{GCD}(A, B))$ is also a solution.

Proof Suppose that (A', B') is another solution to Problem RFR. Then $BR \equiv A \bmod M$ and $B'R \equiv A' \bmod M$. It follows that $B'A \equiv BA' \bmod M$. Hence, $B'A = BA'$ because both $\deg_t AB'$ and $\deg_t A'B$ are smaller than $\deg_t M$.

There exists a polynomial C in $\mathcal{F}[t]$ such that $CM + BR = A$. Since $\mathrm{GCD}(B, M) = 1$, $\mathrm{GCD}(A, B)$ divides C. Thus $(A/\mathrm{GCD}(A, B), B/\mathrm{GCD}(A, B))$ is a solution to the same problem □

RECON_f is the half-extended Euclidean algorithm equipped with a different terminating condition. In order to prove the correctness of RECON_f, one has to prove that, if there exists a solution (A, B) to Problem RFR, A is similar to a member in a PRS of M and R over $\mathcal{F}$. By the algebraic subresultant theory it is sufficient to show that A is similar to a non-zero subresultant of M and R over $\mathcal{F}$.

Lemma 3.3.2 Let $\deg_t M = m$ and $\deg_t R = n$, where $m > n \ge 0$. Assume that (A, B) is a solution to Problem RFR and $\mathrm{GCD}(A, B) = 1$. If $\deg_t B = m - j - 1$, then $A \sim_{\mathcal{F}} \mathrm{sres}_j(M, R)$.

Proof There exists a polynomial C in $\mathcal{F}[t]$, with degree $(n - j - 1)$, such that

$$CM + BR = A. \tag{3.1}$$

Let $\deg_t A = d$ and write

$$A = \sum_{i=0}^{d} a_i t^i,\ B = \sum_{i=0}^{m-j-1} b_i t^i,\ \text{and } C = \sum_{i=0}^{n-j-1} c_i t^i,$$

where a_d, b_{m-j-1}, and c_{n-j-1} are all nonzero. Note that $d \le j$ since $m-j-1 < m/2$ and $d \le m/2$. Let

$$\mathbf{u} = (c_{n-j-1}, \ldots, c_0,\, b_{m-j-1}, \ldots, b_0)$$

and

$$\mathbf{v} = (\,\underbrace{0, \ldots, 0,}_{m+n-j-d-1}\ a_d, a_{d-1}, \ldots, a_0).$$

Moreover, let N be the $(m+n-2j) \times (m+n-j)$ matrix

$$\mathrm{mat}(t^{n-j-1}M, \ldots, M,\, t^{m-j-1}R, \ldots, R).$$

Then equation (3.1) can be written as the linear system

$$\mathbf{u}N = \mathbf{v}. \tag{3.2}$$

First, we prove that $\mathrm{sres}_j(M,R)$ is nonzero. Let N_d be the $(m+n-2j)\times(m+n-2j)$ submatrix whose first $(m+n-2j-1)$ columns are the same as those of N and whose last column is the $(m+n-j-d)$th column of N. Then $\det(N_d)$ is the coefficient of t^d in $\mathrm{sres}_j(M,R)$. Hence, it suffices to prove that $\det(N_d)$ is nonzero.

From (3.2) we see that

$$\mathbf{u}N_d = (\,\underbrace{0, \ldots, 0,}_{m+n-2j-1}\ a_d). \tag{3.3}$$

Suppose that $\mathbf{u}' = (c'_{n-j-1}, \ldots, c'_0,\, b'_{m-j-1}, \ldots, b'_0)$ is another solution of (3.3). Let

$$B' = \sum_{i=0}^{m-j-1} b'_i t^i, \quad C' = \sum_{i=0}^{n-j-1} c'_i t^i, \quad \text{and} \quad A' = C'M + B'R.$$

Then $0 \le \deg_t A' \le j$ because of (3.3).

Equation (3.1) and the definition of A' give rise to the congruent equations $BR \equiv A \bmod M$ and $B'R \equiv A' \bmod M$. Eliminating R from the two congruent equations, we get $B'A \equiv BA' \bmod M$. Hence $B'A = BA'$, since both $\deg_t B'A$ and $\deg_t BA'$ are less than m. Thus, B divides B' since $\mathrm{GCD}(A,B) = 1$. Consequently, there exists h in $\mathcal{F}$ such that $hB = B'$, because $\deg_t B'$ is not greater than $\deg_t B$. It follows that $hA = A'$, so $h = 1$ because a_d is the coefficient of t^d in both A and A'. Hence $A = A'$, $B = B'$, and, moreover, $C = C'$. We then conclude that $\mathbf{u} = \mathbf{u}'$, i.e., linear system (3.3) has a unique solution. Thus, $\det(N_d)$ is nonzero.

By Lemma 7.7.4 in [32, p. 255], there are polynomials C'' and B'' in $\mathcal{F}[t]$, with $\deg_t C'' \le n-j-1$ and $\deg_t B'' \le m-j-1$, such that $C''M + B''R = \mathrm{sres}_j(M,R)$. This equation and (3.1) give rise to the congruent equations $BR \equiv A \bmod M$ and $B''R \equiv \mathrm{sres}_j(M,R) \bmod M$. The same argument as in the previous paragraph proves that A and $\mathrm{sres}_j(M,R)$ are $\mathcal{F}$-linear dependent. Thus, A and $\mathrm{sres}_j(M,R)$ are similar over $\mathcal{F}$ because they are nonzero. □

We recall some basic properties of the extended Euclidean algorithm (see Knuth [20, Excercise 3 in §4.6.1]). Let A_1 and A_2 be in $\mathcal{F}[t]$ such that $\deg_t A_1 \geq \deg_t A_2 > 0$. The extended Euclidean algorithm with inputs A_1 and A_2 generates three sequences (in $\mathcal{F}[t]$):

$$A_1, A_2, \ldots, A_r, \quad U_1, U_2, \ldots, U_r, \quad \text{and} \quad V_1, V_2, \ldots, V_r,$$

with the properties that, for $i = 3, 4, \ldots, r$,

1. A_i is the remainder of A_{i-2} and A_{i-1};
2. $U_iA_1 + V_iA_2 = A_i$, where $\deg_t U_i < \deg_t A_2 - \deg_t A_i$ and $\deg_t V_i < \deg_t A_1 - \deg_t A_i$;
3. U_i and V_i are relatively prime.

The last property follows from the fact that $U_{i-1}V_i - U_iV_{i-1} = \pm 1$, for $i = 2, 3, \ldots, r$.

We are ready to prove the correctness of the algorithm RECON_f.

Proposition 3.3.3 Problem RFR has a solution if and only if RECON_f with inputs M and R returns a pair (A, B). If RECON_f returns a pair (A, B), then

$$AB^{-1} \equiv R \bmod M \quad \text{and} \quad \mathrm{GCD}(A, B) = 1.$$

Proof If $\deg_t R \leq (\deg_t M)/2$, then RECON_f returns the pair $(R, 1)$. If $\deg_t R \geq \deg_t M$, then the residue R can be replaced by the remainder of R and M. We may then assume that $(\deg_t M)/2 < \deg_t R < \deg_t M$. Suppose that RECON_f returns (A, B) in the ith iteration. Then we have $BR \equiv A \bmod M$, because RECON_f preserves the relation $V_jR \equiv A_j \bmod M$, for $2 \leq j \leq i$, where V_j and A_j are produced by RECON_f. Moreover, the relation $\mathrm{GCD}(V_i, A_i) = 1$ implies that $\mathrm{GCD}(V_i, M) = 1$, according to the third property of the extended Euclidean algorithm. The pair (A, B) is the desired solution.

Conversely, let (A, B) be the solution to Problem RFR with $\mathrm{GCD}(A, B) = 1$. Let M, R, A_3, $\ldots$, A_k be a PRS generated by the Euclidean algorithm. Then Lemma 3.3.2 implies that there is a non-zero element a in $\mathcal{F}$ such that $A = aA_l$. We then have the following congruent equations:

$$BR \equiv A \bmod M \quad \text{and} \quad aV_lR \equiv A \bmod M,$$

where $\deg_t V_l < \deg_t M - \deg_t A_l$ by the second property of the extended Euclidean algorithm. Eliminating R from the above congruent equations, we get $(aV_l - B)A \equiv 0 \bmod M$. Thus, $aV_l = B$ since $\deg_t(aV_l - B)A < \deg_t M$. As the degrees of the V_j's increase and the degrees of the A_j's decrease in RECON_f, the pair (A, B) is found in the lth iteration. □

Based on RECON_n and RECON_f we present the algorithms COEFF_n and COEFF_f that reconstruct the rational number and rational function coefficients of polynomials from the given residues, respectively. These two algorithms, together with CRA, will be used to combine modular and evaluation homomorphic images. As these two algorithms can be worked out easily, we only specify their inputs and outputs.

algorithm COEFF_n

Input: A modulus $m \in \mathbf{N}^+$ and a non-zero residue $R \in \mathbf{Z}_m[t][X]$.
Output: $A \in \mathbf{Q}[t][X]$, such that $A \equiv R \bmod m$ and the denominators and numerators of the rational coefficients in A range from $-\sqrt{m/2}$ to $\sqrt{m/2}$ if such a polynomial exists. Otherwise, NIL is returned.

algorithm COEFF_f

Input: A modulus $M \in \mathbf{Z}_p[t]$ with $\deg_t M > 0$, and a non-zero residue $R \in \mathbf{Z}_p[t][X]$.
Output: $A \in \mathbf{Z}_p(t)[X]$, such that $A \equiv R \bmod M$,
the denominators of the coefficients of A have degrees $< (\deg_t M)/2$,
and the numerators of the coefficients in A have degrees $\leq (\deg_t M)/2$,
if such a polynomial exists. Otherwise, NIL is returned.

3.4 Modular Algorithm for Computing GCRDs over $\mathbf{Z}_p[t]$

Let $(\mathbf{Z}_p[t][X], \sigma_p, \delta_p)$ be an Ore polynomial ring. We present the modular algorithm GCRD_p for computing GCRDs in this ring. We reduce the GCRD problem in $\mathbf{Z}_p[t][X]$ to a series of the problems of computing the evaluation homomorphic images in $\mathbf{Z}_p[X]$, which are solved by GCRD_e. The "lucky" evaluation homomorphic images are combined by CRA and COEFF_f. The termination of GCRD_p is determined by trial division. It is a rare, though possible case that there are not enough lucky evaluation points in $\mathbf{Z}_p$. If this happens, GCRD_p reports *failure*.

algorithm GCRD_p

Input: A prime p and $A, B \in \mathbf{Z}_p[t][X]$ with $\deg A \geq \deg B \geq 1$.

Output: C, where $C = \mathrm{GCRD}(A, B)$.

[initialize the modulus, residue, and degree]

1. $k \leftarrow 0$;

2. **repeat**

3. **if** $k = p$ **then** { *report failure;* } $R_k \leftarrow \mathrm{GRCD\_e}(p, k, A, B)$; $k \leftarrow k + 1$;

4. **until** $R_k \neq 0$

5. $d_k \leftarrow \deg R_k$;

6. **if** $d_k = 0$ **then return**(1);

7. $M \leftarrow t - k$; $R \leftarrow R_k$; $d \leftarrow d_k$; $C \leftarrow 0$;

[main loop]

8. **while** true **do** {

9. **repeat**

10. **if** $k = p$ **then** { *report failure*; } $R_k \leftarrow \mathrm{GRCD\_e}(p, k, A, B)$; $k \leftarrow k + 1$;

11. **until** $R_k \neq 0$

12. $d_k \leftarrow \deg R_k$;

[test for unlucky evaluation points]

13. **if** $d_k < d$ **then goto** line 7;

14. **if** $d_k = d$ **then** {

[combine]

15. $R \leftarrow \mathrm{CRA}(R, M, R_k, t - k)$; $M \leftarrow (t - k)M$;

16. $\bar{C} \leftarrow \mathrm{COEFF\_f}(M, R)$;

17. **if** $C \neq 0$ and $C = \bar{C}$ **then**

[trial division]

18. **if** $A \equiv 0 \bmod C$ and $B \equiv 0 \bmod C$ **then return**(the numerator of C);

19. $C \leftarrow \bar{C}$; } }

Proposition 3.4.1 The algorithm GCRD_p is correct.

Proof Let G be $\mathrm{GCRD}(A, B)$ with degree l. If $l = 0$, then GCRD_p returns 1 when there exists a lucky evaluation point in $\mathbf{Z}_p$. From now on, assume $l > 0$. If there are less than $(2l + 2)$ lucky points in $\mathbf{Z}_p$, GCRD_p reports failure. Assume that there are more than $(2l + 1)$ lucky points in $\mathbf{Z}_p$. Then the tentative degree d in GCRD_p will be eventually equal to l, because, for each unlucky point, GCRD_e returns either 0 or a polynomial of degree greater than l. Unlucky evaluation points can be detected in line 13 as soon as a lucky one is encountered. So we may suppose that d is equal to l. Then each R_k entering CRA in line 15 is equal to $\phi_{t-k}(G/\mathrm{lc}(G))$ by Proposition 3.2.1. Hence $R \equiv G/\mathrm{lc}(G) \bmod M$ in GCRD_p. Since the solution to Problem RFR is unique, COEFF_f in line 16 recovers $G/\mathrm{lc}(G)$ when $\deg_t M$ exceeds $2l$. COEFF_f produces $G/\mathrm{lc}(G)$ again when the next lucky evaluation point is encountered. At this point the condition $C = \bar{C}$ in line 17 is satisfied. Hence GCRD_p returns G after a trial division. □

The next lemma ensures that GCRD_p does not report failure if p is sufficiently large.

Lemma 3.4.2 If A and B are in $\mathbf{Z}_p[t][X]$, with respective degrees m and n, where $m \geq n \geq 1$, then there are at most

$$\deg_t \left(\prod_{i=0}^{m-1} \sigma_p^i(\mathrm{lc}(B)) \right) + m \deg_t B + n \deg_t A \tag{3.4}$$

unlucky evaluation points for A and B.

Proof If k is unlucky for A and B, k is a root of

$$\left(\prod_{i=0}^{m-1} \sigma_p^i(\mathrm{lc}(B)) \right) \mathrm{lc}(\mathrm{sres}_d(A, B)),$$

where d is the degree of GCRD(A, B), and (3.4) gives a degree bound for this polynomial. □

3.5 Modular Algorithm for Computing GCRDs over $\mathbf{Z}[t]$

In this section, we let A and B be in $\mathbf{Z}[t][X]$, with respective degrees m and n, where $m \geq n \geq 1$. Assume that $G = \mathrm{GCRD}(A, B)$ with degree k. Let us denote by ϕ_p the usual modular homomorphism from $\mathbf{Z}[t][X]$ to $\mathbf{Z}_p[t][X]$. Using modular homomorphisms we reduce the problem of computing G to a series of the problems of computing the monic associates of the modular homomorphic images of G. First, we define unlucky primes.

Definition 3.5.1 A prime p is *unlucky* for A and B if one of the following holds:

1. p is a divisor of $\mathrm{hc}(\sigma(t))\mathrm{lc}(A)\mathrm{lc}(B)$;
2. p is a divisor of $\mathrm{lc}(\mathrm{sres}_k(A, B))$;
3. p is a divisor of $\mathrm{hc}(G)$;

4. $\phi_p(G)$ is not primitive with respect to X.

We shall present three technical lemmas to prove the correctness of the modular algorithm for computing GCRDs over $\mathbf{Z}[t]$.

The first lemma describes the relation between $\mathcal{S}(A, B)$ and $\mathcal{S}(\phi_p(A), \phi_p(B))$.

Lemma 3.5.1 If p is not a divisor of $\mathrm{hc}(\sigma(t))\mathrm{lc}(A)\mathrm{lc}(B)$, then

$$\mathrm{sres}_i(A, B) = \mathrm{sres}_i(\phi_p(A), \phi_p(B)),$$

for $i = n, n-1, \ldots, 0$.

Proof Since p is not a divisor of $\mathrm{hc}(\sigma(t))$, Lemmas 3.1.1 and 3.1.2 imply that $(\mathbf{Z}_p[t][X], \sigma_p, \delta_p)$ is the Ore polynomial ring induced from $\mathbf{Z}[t][X]$ by ϕ_p. Furthermore, ϕ_p is a ring homomorphism from $\mathbf{Z}[t][X]$ to $\mathbf{Z}_p[t][X]$ by Corollary 3.1.3. Since p is not a divisor of $\mathrm{lc}(A)\mathrm{lc}(B)$, $\deg A = \deg \phi_p(A)$ and $\deg(B) = \deg(\phi_p(B))$. Therefore,

$$\begin{aligned}
\mathrm{sres}_i(\phi_p(A), \phi_p(B)) &= \mid X^{n-i-1}\phi_p(A), \ldots, \phi_p(A),\ X^{m-i-1}\phi_p(B), \ldots, \phi_p(B) \mid \\
&= \mid \phi_p(X^{n-i-1}A), \ldots, \phi_p(A),\ \phi_p(X^{m-i-1}B), \ldots, \phi_p(B) \mid \\
&= \phi_p\left(\mid X^{n-i-1}A, \ldots, A,\ X^{m-i-1}B, \ldots, B \mid\right) \\
&= \phi_p(\mathrm{sres}_i(A, B)),
\end{aligned}$$

for $i = n-1, n-2, \ldots\ 0$. □

The second lemma describes the relation between $\phi_p(G)$ and $\mathrm{GCRD}(\phi_p(A), \phi_p(B))$ when p is not unlucky.

Lemma 3.5.2 If p is not unlucky and $\mathbf{Z}_p[t][X]$ is the Ore polynomial ring induced from $\mathbf{Z}[t][X]$ by ϕ_p, then

$$\mathrm{GCRD}(\phi_p(A), \phi_p(B)) = \phi_p(G)/\phi_p(\mathrm{hc}(G)). \tag{3.5}$$

Proof As $\deg A = \deg \phi_p(A)$, $\deg B = \deg \phi_p(B)$, and ϕ_p is an Ore ring homomorphism,

$$\mathrm{sres}_k(\phi_p(A), \phi_p(B)) = \phi_p((\mathrm{sres}_k(A, B)) \neq 0$$

by Lemma 3.5.1. So the degree of $\mathrm{GCRD}(\phi_p(A), \phi_p(B))$ is not greater than k, since every common right factor of A and B must be a right factor of their subresultants. On the other hand, Corollary 3.1.3 implies that $\phi_p(G)$ is a common right factor of $\phi_p(A)$ and $\phi_p(B)$. Thus, $\phi_p(G)$ is a GCRD of $\phi_p(A)$ and $\phi_p(B)$ because $\deg \phi_p(G) = k$. Hence (3.5) holds because $\phi_p(G)$ is primitive with respect to X. □

The last lemma describes the difference between $\phi_p(G)$ and $\mathrm{GCRD}(\phi_p(A), \phi_p(B))$ when p is unlucky.

Lemma 3.5.3 If p is unlucky and is not a divisor of $\mathrm{hc}(\sigma(t))\mathrm{lc}(A)\mathrm{lc}(B)$, then we have either

$$\deg \mathrm{GCRD}(\phi_p(A), \phi_p(B)) > k$$

or

$$\deg \mathrm{GCRD}(\phi_p(A), \phi_p(B)) = k \quad \text{and} \quad \deg_t \mathrm{GCRD}(\phi_p(A), \phi_p(B)) < \deg_t \mathrm{lc}(G).$$

Proof Let $G_p = \mathrm{GCRD}(\phi_p(A), \phi_p(B))$, $D_p = \deg G_p$ and $d_p = \deg_t G_p$. Notice that p satisfies one of the last three conditions listed in Definition 3.5.1, since p is not a divisor of $\mathrm{hc}(\sigma(t))\mathrm{lc}(A)\mathrm{lc}(B)$.

Assume that p is a divisor of $\mathrm{lc}(\mathrm{sres}_k(A, B))$. By Lemma 3.5.1, the degree of $\mathrm{sres}_k(\phi_p(A), \phi_p(B))$, say d, is less than k. First, we prove that $\mathrm{sres}_k(\phi_p(A), \phi_p(B))$ is zero. If $\mathrm{sres}_k(\phi_p(A), \phi_p(B))$ is nonzero, then $D_p \le d$ because G_p is a right divisor of $\mathrm{sres}_k(\phi_p(A), \phi_p(B))$. Lemma 2.2.3 then implies that the D_pth subresultant of $\phi_p(A)$ and $\phi_p(B)$ is nonzero. On the other hand, the D_pth subresultant of A and B is zero since $D_p < k$. So the D_pth subresultant of $\phi_p(A)$ and $\phi_p(B)$ is zero by Lemma 3.5.1. This contradiction proves that $\mathrm{sres}_k(\phi_p(A), \phi_p(B))$ is zero. Again, by Lemma 2.2.3 the D_pth subresultant of $\phi_p(A)$ and $\phi_p(B)$ is nonzero. Hence $D_p > k$.

Assume that p is *not* a divisor of $\mathrm{lc}(\mathrm{sres}_k(A, B))$ in the rest of the proof. Then $\mathrm{sres}_k(\phi_p(A), \phi_p(B))$ is of degree k. Since G_p is a right divisor of $\mathrm{sres}_k(\phi_p(A), \phi_p(B))$, we have $D_p \le k$. On the other hand, $\mathrm{sres}_k(\phi_p(A), \phi_p(B))$ is a right divisor of $\phi_p(A)$ and $\phi_p(B)$, because $\mathrm{sres}_k(A, B)$ is a right divisor of A and B by Lemma 2.2.3. Hence $D_p \ge k$. We have proved that $D_p = k$. From Lemma 2.2.3 and the definition of normalized GCRDs, it follows that $\mathrm{sres}_k(A, B) = a\,G$ and $\mathrm{sres}_k(\phi(A), \phi(B)) = b\,G_p$, for some $a \in \mathbf{Z}[t]$ and $b \in \mathbf{Z}_p[t]$. Hence $\phi_p(a)\phi_p(G) = b\,G_p$. We then have

$$\phi_p(G) = c\,G_p, \tag{3.6}$$

for some $c \in \mathbf{Z}_p[t]$, because G_p is primitive with respect to X. If p is a divisor of $\mathrm{lc}(G)$, then $\deg_t \mathrm{lc}(G) > \deg_t \mathrm{lc}(G_p)$ by (3.6). If $\phi_p(G)$ is not relatively prime with respect to X, then $\deg_t c$ is positive, and hence, $\deg_t \mathrm{lc}(G) > \deg_t \mathrm{lc}(G_p)$. □

Clearly, there are only finitely many unlucky primes for A and B. For each lucky prime p, $\phi_p(G)/\mathrm{hc}(\phi_p(G))$ can be obtained from $\mathrm{GCRD}(\phi_p(A), \phi_p(B))$ by Lemma 3.5.2. These considerations lead to the algorithm GCRD_m.

algorithm GCRD_m

Input: $A, B \in \mathbf{Z}[t][X]$.

Output: C, where $C = \mathrm{GCRD}(A, B)$.

[initialize]

1. **if** $\deg(A) \geq \deg(B)$ **then** $\{\ A_1 \leftarrow A;\ A_2 \leftarrow B;\ \}$

2. **else** $\{\ A_1 \leftarrow B;\ A_2 \leftarrow A;\ \}$

3. $A_1 \leftarrow$ the primitive part of A_1 w.r.t. X;

4. $A_2 \leftarrow$ the primitive part of A_2 w.r.t. X;

5. $b \leftarrow \mathrm{hc}(A_1)\mathrm{hc}(A_2)\mathrm{hc}(\sigma(t))$;

[initialize the modulus, residue, and degrees]

6. $p \leftarrow$ a large prime not dividing b;

7. $R_p \leftarrow \mathrm{GCRD\_p}(p, \phi_p(A_1), \phi_p(A_2))$;

8. $D_p \leftarrow \deg R_p$; $d_p \leftarrow \deg_t \mathrm{lc}(R_p)$;

9. **if** $D_p = 0$ **then return**(1);

10. $M \leftarrow p$; $R \leftarrow R_p$; $D \leftarrow D_p$; $d \leftarrow d_p$; $C \leftarrow 0$

[main loop]

11. **while** true **do** $\{$

12. $p \leftarrow$ a new large prime not dividing b;

13. $R_p \leftarrow \mathrm{GCRD\_p}(p, \phi_p(A_1), \phi_p(A_2))$;

14. $D_p \leftarrow \deg R_p$; $d_p \leftarrow \deg_t \mathrm{lc}(R_p)$;

[test for unlucky primes]

15. **if** $D_p < D$ **then goto** line 9;

16. **if** $D_p = D$ and $d_p > d$ **then goto** line 10;

[combine]

17. **if** $D_p = D$ and $d_p = d$ **then** $\{$

18. $R \leftarrow \mathrm{CRA}(R, M, R_p, p)$; $M \leftarrow pM$;

19. $\bar{C} \leftarrow \mathrm{COEFF\_n}(M, R)$;

20. **if** $C \neq 0$ and $C = \bar{C}$ **then**

[trial division]

21. **if** $A_1 \equiv 0 \bmod C$ and $A_2 \equiv 0 \bmod C$ **then** **return**(the numerator of C);

22. $C \leftarrow \bar{C};\ \}\ \}$

Remark 3.5.2 By "large prime" p, we mean that p is so large that GCRD_p does not report failure. It is always possible to choose such p by Lemma 3.4.2.

Proposition 3.5.4 The algorithm GCRD_m is correct.

Proof As b is assigned to be $\mathrm{hc}(A_1)\mathrm{hc}(A_2)\mathrm{hc}(\sigma(t))$ in line 5, If p is unlucky, then, by Lemma 3.5.3, GCRD_p can only result R_p in lines 7 and 13 such that either $\deg R_p > \deg G$ or $\deg_t \mathrm{lc}(R_p) < \deg_t \mathrm{lc}(G)$. Unlucky primes can be detected in lines 15 and 16 as soon as a lucky prime is encountered. Since there are only a finite number of unlucky primes, we may further assume that $D = \deg G$ and $d = \deg_t \mathrm{lc}(G)$. Accordingly, the polynomial R in line 18 satisfies the congruence $R \equiv G/\mathrm{hc}(G) \bmod M$ by Lemma 3.5.2. Then the polynomial $\bar{C}$ computed by COEFF_n in line 19 is equal to $G/\mathrm{hc}(G)$ as soon as $\sqrt{M/2}$ exceeds the absolute value of the maximum of the integral coefficients of G. Thus, GCRD_m returns G. □

The advantages of GCRD_m are clear. The problem of finding $\mathrm{GCRD}(A, B)$ is mapped to the domains in which the arithmetic does not cause any intermediate swelling. In addition, GCRD_m can recognize the case when $\mathrm{GCRD}(A, B)$ is trivial as soon as a lucky prime is encountered.

3.6 Experimental Results

This section presents experimental results to compare the algorithm GCRD_m, subresultant algorithm, and primitive Euclidean algorithm. We implemented in *Maple V* (Release 3) these three algorithms for the differential operator D and shift operator E with coefficients in $\mathbf{Z}[t]$, where D and E are defined in Examples 2.2.2 and 2.2.3, respectively.

The first suite was generated as follows. We used the Maple function `randpoly` to generate pairs of bivariate polynomials in $\mathbf{Z}[t, X]$, with total degree n and $n-1$, where $n = 5$, 10, and 15. These polynomials had five terms with coefficients ranging from -99 to 99. We then regarded these polynomials as differential operators in $\mathbf{Z}[t][D]$ and shift operators in $\mathbf{Z}[t][E]$, respectively, and computed the GCRD of each pair. The timings are summarized in Figure 3.1, in which the column labeled n gives the total degrees of the polynomials; the columns labeled DM, DS, DPE, give the respective computing times for GCRD_m, the subresultant algorithm, and primitive Euclidean algorithm whose inputs are differential operators; similarly, the columns labeled SM, SS, SPE, give the respective computing times for GCRD_m, subresultant algorithm, and primitive Euclidean algorithm whose inputs are shift operators. All the entries are Maple CPU time and given in seconds.

n	DM	DS	DPE	SM	SS	SPE
5	0.20	0.27	0.19	0.17	0.25	0.21
10	0.99	38.86	39.71	0.59	42.73	40.71
15	1.65	301.25	374.00	0.77	436.47	485.91

Figure 3.1: Computing times for the first suite

We see from Figure 3.1 that GCRD_m is considerably faster than non-modular ones when input polynomials are of total degree more than eight. This is not a surprise since the GCRD of two random polynomials is usually trivial. In practice, GCRD_m can detect the case when input polynomials have a trivial GCRD by one or two primes. The timings also indicate that the subresultant algorithm is slightly faster than the primitive Euclidean algorithm when input polynomials are chosen at random.

To construct the second suite, we used `randpoly` to generate three polynomials, say, A, B, and C in $\mathbf{Z}[t, X]$, with respective total degrees $n-2$, $n-3$, and 2, where $n = 5$, 10, and 15. The number of terms and length of coefficients were the same as those in the first suite. We took the differential (shift) products AC and BC as input polynomials. Thus, the GCRD of each pair of input polynomials was usually nontrivial. The timings are summarized in Figure 3.2, where a dash (–) indicates that our implementation of the primitive Euclidean algorithm took more than three hours without any output. This could happen because it took very long time to compute the primitive part of a polynomial in $\mathbf{Z}[t][X]$ when the content had large integral coefficients.

n	DM	DS	DPE	SM	SS	SPE
5	2.26	0.25	0.15	1.29	0.30	0.15
10	9.91	64.25	16.72	3.74	57.66	18.67
15	27.23	1348.83	–	6.46	1999.64	–

Figure 3.2: Computing times for the second suite

Again, the timings in Figure 3.2 indicate that GCRD_m is more efficient than non-modular ones. We also remark that the subresultant algorithm may be slower than the primitive Euclidean algorithm when input polynomials have a non-trivial GCRD. This is because the primitive Euclidean algorithm removes more extraneous factors after each division when the GCRD is not monic, according to Theorem 2.2.8.

Bibliography

[1] L. M. Berkovich and V. G. Tsirulik. Differential Resultants and Some of Their Applications. In *Differential'nye Uravneniya* **22**(5), 750–757, 1986.

[2] M. Bronstein and M. Petkovšek. On Ore Rings, Linear Operators and Factorization. *Programming and Comput. Software* **20**, 14–26, 1994.

[3] W. S. Brown. On Euclid's Algorithm and the Computation of Polynomial Greatest Common Divisors. *JACM* **18**, 478–504, 1971.

[4] W. S. Brown and J. F. Traub. On Euclid's Algorithm and the Theory of Subresultants. *JACM* **18**, 505–514, 1971.

[5] M. Chardin. Differential Resultants and Subresultants. In *Proceedings of Fundamentals of Computation Theory, Lecture Notes in Computer Science* **529**, 180–189, 1991.

[6] F. Chyzak. Holonomic Systems and Automatic Proofs of Identities. *Research report*, INRIA, Centre de Diffusion, BP 105–78153 Le Chesnay Cedex, France, 1994.

[7] R. M. Cohn. *Difference Algebra*. Interscience Publishers. 1965.

[8] G. E. Collins. Polynomial Remainder Sequences and Determinants. *American Mathematical Monthly* **73**(7), 709–712, 1966.

[9] G. E. Collins. Subresultant and Reduced Polynomial Remainder Sequences. *JACM* **16**, 708–712, 1967.

[10] G. E. Collins. The Calculation of Multivariate Polynomial Resultants. *JACM* **18**, 515–532, 1971.

[11] G. E. Collins. Quantifier Elimination for the Elementary Theory of Real Closed Fields by Cylindrical Algebraic Decomposition. In *Automata Theory and Formal Languages, 2nd GI Conference, Lecture Note in Computer Science* **33**, 234–183, Berlin, Springer-Verlag, 1975.

[12] G. E. Collins and M. J. Encarnación. Efficient Rational Number Reconstruction. *Technical Report*, no. 94-64, RISC-Linz, Johannes Kepler University, A-4040, Austria. To appear in *Journal of Symbolic Computation*.

[13] G. E. Collins and R. Loos. Real Zeros of Polynomials. In B. Buchberger, G. E. Collins, and R. Loos (eds.), *Computer Algebra, Symbolic and Algebraic Computation,* 83–94. Springer-Verlag, Wien-New York, 1982.

[14] M. J. Encarnación. On a Modular Algorithm for Computing Gcds of Polynomials over Algebraic Number Fields. In *Proceedings of the 1994 International Symposium on Symbolic and Algebraic Computation*, 58–65, ACM Press, 1994.

[15] M. J. Encarnación. *Faster Algorithms for Reconstructing Rationals, Computing Polynomial GCDs, and Factoring Polynomials.* Ph.D. Thesis, RISC-Linz, Johannes Kepler University, A-4040, Linz, Austria. 1995.

[16] K. O. Geddes, S. R. Czapor, and G. Labahn. *Algorithms for Computer Algebra.* Kluwer Academic Publishers. 1992.

[17] L. Gonzàlez-Vega. Determinantal Formula for the Solution Set of Zero-Dimensional Ideals. *Journal of Pure and Applied Algebra* **76**, 57–80, 1991.

[18] D. Yu. Grigor'ev. Complexity of Factoring and Calculating the GCD of Linear Ordinary Differential Operators. *Journal of Symbolic Computation* **10**(1), 7–37, 1990.

[19] J. Johnson. Private Communication, 1994

[20] D. E. Knuth. *The Art of Computer Programming* **2**. Addison-Wesley. 1981.

[21] E. R. Kolchin. *Differential Algebra and Algebraic Groups.* Pure and Applied Math **54**. Academic Press, New York-London, 1973.

[22] T. Y. Lam. *A First Course in Non-commutative Rings.* Graduate Texts in Mathematics **131**, Springer-Verlag. 1991.

[23] Z. Li. An Implementation of the Characteristic Set Method for Solving Algebraic Equations. *Technical Report,* no. 94-86, RISC-Linz, Johannes Kepler University, A-4040, Linz, Austria, 1994.

[24] Z. Li. A Subresultant Theory for Linear Ordinary Differential Polynomials, *Technical Report*, no. 95-35, RISC-Linz, Johannes Kepler University, A-4040, Linz, Austria, 1995.

[25] Z. Li and I. Nemes. A Modular Algorithm for Computing Greatest Common Right Divisors of Ore Polynomials. Submitted to *the 1996 International Symposium on Symbolic and Algebraic Computation.*

[26] R. Loos. Generalized Polynomial Remainder Sequence. In B. Buchberger, G. E. Collins, and R. Loos (eds.), *Computer Algebra, Symbolic and Algebraic Computation,* 115–137. Springer-Verlag, Wien-New York, 1982.

[27] A. M. Mandache. The Gröbner Basis Algorithm and Subresultant Theory. In *Proceedings of the 1994 International Symposium on Symbolic and Algebraic Computation*, 20–23, ACM Press, 1994.

[28] A. M. Mandache. *Gröbner Bases Computation and Gaussian Elimination.* Ph.D. Thesis, RISC-Linz, Johannes Kepler University, A-4040, Linz, Austria. 1995.

[29] L. M. Milne-Thomson. *The Calculus of Finite Differences.* Macmillan & Co. Ltd. 1960.

[30] D. Manocha and J. F. Canny. Algorithm for Implicitizing Rational Parametric Surfaces. *Journal of Computer Aided Geometric Design* **9**, 25–50, 1992.

[31] D. Manocha and J. F. Canny. Implicit Representations of Rational Parametric Surfaces. *Journal of Symbolic Computation* **13**(5) 485–510, 1992.

[32] B. Mishra. *Algorithmic Algebra.* Texts and Monographs in Computer Science, D. Gries D and F. B. Schneider (eds.), Springer-Verlag. 1993.

[33] O. Ore. Theory of Non-Commutative Polynomials. *Annals of Math* **34**, 480–508, 1933.

[34] Peter Paule and Volker Strehl. Symbolic Summation – Some Recent Developments. *Technical Report,* no. 95-11, RISC-Linz, Johannes Kepler University, Linz, A-4040, Austria. To appear in *Computer Algebra in Science and Engineering*, J. Fleascher, J. Grabmeier, F. Hehl, and W. Wüchlin (eds.), World Scientific, Singapore.

[35] E. G. C. Poole. *Introduction to the Theory of Linear Ordinary Differential Equations.* Dover Publications Inc., New York. 1936.

[36] J. F. Ritt. *Differential Algebra.* AMS. 1950.

[37] C. M. Rubald. *Algorithms for Polynomials over a Real Algebraic Number Field.* Ph.D. Thesis, Department of Computer Science, University of Wisconsin. 1973.

[38] B. Salvy, and P. Zimmermann. Gfun: A Maple Package for the Manipulation of Generating and Holonomic Functions in One Variable. *ACM Transactions on Mathematical Software* **20**, 163–177, 1994.

[39] B. Z. Shen. Solving a Congruence on a Graded Algebra by a Subresultant Sequence and Its Application. *Journal of Symbolic Computation* **14**(5), 505–522, 1992.

[40] P. S. Wang. A p-adic Algorithm for Univariate Partial Fractions. In Proceedings of the 1981 Symposium on Symbolic and Algebraic Computation, 212–217, ACM Press, 1981.

[41] P. S. Wang, M. J. T. Guy, and J. H. Davenport. p-adic Reconstruction of Rational numbers. *SIGSAM Bulletin* **16**, 2–3, 1982.

[42] H. S. Wilf and D. Zeilberger. An Algorithmic Proof of Theory for Hypergeometric (Ordinary and "q") Multisum / Integral Identities. *Inventiones Mathematicae* **108**, 575–633, 1992.

Ziming Li

Institute of Systems Science

Academia Sinica

Phone: +86 01 62541834

Email: `zmli@mmrc.iss.ac.cn`

Personal

Born June 6, 1962, Beijing, China. P. R. China citizen.

Education

- PhD., Research Institute for Symbolic Computation (RISC-Linz) Johannes Kepler University, Linz, Austria.
- M.S., Institute of Systems Science, Academia Sinica, Beijing, 1988
- B.S., Department of Applied Mathematics, Tsinghua University, Beijing, 1985.

Positions

- 6/96– Assistant Professor, Institute of Systems Science, Academia Sinica, Beijing, China.
- 2/92–4/96 Research assistant, Research Institute for Symbolic Computation, Johannes Kepler University, Linz, Austria.
- 7/88–1/92 Lecturer, Department of Applied Mathematics, Tsinghua University, Beijing.

Journal Publications

- Computations with Rational Parametric Equations (with S.C. Chou and X.S. Gao), *Computer Mathematics*, 86–111, World Scientific Pub., River Edg. NJ, 1993.
- Mechanical Theorem Proving in the Local Theory of Surfaces, *Annals of Mathematics and Artificial Intelligence*, **13**, 25–46. 1995.
- Determinantal Polynomial Sequences, *Ke Xue Tong Bao* **33**, 1369–1372, 1988.

Conference Papers

- Finding Roots of Unity among Quotients of the Roots of an Integral Polynomial (with K. Yokoyama and I. István), in *Proceedings of the 1995 International Symposium on Symbolic Computation*, 85–89, ACM Press, 1995

- An Implementation of the Characteristic Set Method for Solving Algebraic Equations, accepted by PoSSo Workshop on Software, Paris, 1995.

Technical Reports

- A Subresultant Theory for Linear Differential, Linear Difference, and Ore Polynomials, with Applications, RISC-Linz PhD Thesis, 1996.

- A Subresultant Theory for Linear Differential Polynomials, RISC-Linz Technical Report, no. 95-35, 1995.

- Computing Tighter Degree Bounds in Elimination Theory, Mathematics-Mechanization Research Preprints, **12**, 32–52, 1994.

- Automatic Implicitization of Parametric Objects, Mathematics-Mechanization Research Preprints, **4**, 54–62, 1989.

- On the Triangulation for a finite polynomial set(I), Mathematics-Mechanization Research Preprints, **2**, 48–54, 1987.

- A New Proof of Collins' Theorem, Mathematics-Mechanization Research Preprints, **1**, 33–37, 1987.