Elimination in Operator Algebras

By

Anen Lakhal

zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
(Dr. rer. nat.)
im Fachbereich Mathematik
der Universität Kassel

Ph.D. thesis supervised by:

Prof. Dr. Wolfram Koepf
University of Kassel, Germany

July 2014

UNI KASSEL
VERSITÄT

Tag der mündlichen Prüfung
14. Juli 2014

Erstgutachter

Prof. Dr. Wolfram Koepf
Universität Kassel

Zweitgutachter

Prof. Dr. Werner Seiler
Universität Kassel

Acknowledgement

Completing my PhD degree is probably the most challenging activity of my life. The best and worst moments of my doctoral journey have been shared with many people, whose contribution is obvious. My first debt of gratitude must go to my dear advisor, Prof. Dr. Wolfram Koepf for supervising my work. He provided me with insightful discussions, a lot of encouragement and the necessary advices and suggestions in order to proceed through the doctorial program. I am greatly indebted for his invaluable support with helping me to find financial resources for my PhD. Special thanks to all my colleagues at the Department of Mathematics at the University of Kassel for their unconditional help and especially for the friendly atmosphere that we shared for years. At the end, I wish to thank my dear parents, Omar Lakhal and Manana Mahjoub and my two brothers Aref and Majdi. Their love and their perpetual support were my driving force in this special academic and personal experience. In particular I would like to thank my soul mate, my wife Meriam Gazzah. Her encouragement, quiet patience, big love and her faith in my abilities were the engine of my doctoral journey. During these past several years and thanks to my PhD we both learned how to be strong and determined to reach our target and to live life to the fullest.

Contents

1 Introduction **1**

2 Ore Algebras **3**

2.1 Ore extensions . . . 3

2.2 Action of Ore algebras . . . 8

2.3 Skew Euclidean Division . . . 10

3 Noncommutative Gröbner Bases **17**

3.1 Algebras of Solvable Type . . . 17

3.1.1 Notations and Preliminaries . . . 17

3.1.2 Definition and Properties of Solvable Algebras . . . 18

3.1.3 Examples of Solvable Algebras . . . 22

3.1.4 Hilbert's Basis Theorem for Solvable Algebras . . . 23

3.2 Gröbner bases in Solvable Algebras . . . 24

3.2.1 Left Reduction . . . 24

3.2.2 Left Buchberger's Algorithm . . . 28

3.2.3 Applications . . . 32

4 ∂-Finite and Holonomic Functions **39**

4.1 ∂-Finite Functions . . . 39

4.1.1 Definition and Characterization . . . 39

4.1.2 Closure Properties . . . 41

4.2 Holonomic functions . . . 50

4.3 Conclusion . . . 59

5 Algorithms for Summation and Integration **63**

5.1 Creative Telescoping . . . 63

5.1.1 Zeilberger's Slow Algorithm . . . 65

5.1.2 Takayama's Algorithm . . . 73

5.2 Chyzak's Algorithm . . . 82

5.2.1 Indefinite case . . . 83

5.2.2 Definite Case . . . 88

Bibliography **95**

List of Algorithms **99**

Chapter 1

Introduction

Computer algebra is a scientific area that refers to the study and development of algorithms and software for manipulating mathematical expressions and mathematical objects. A particularly interesting class of objects is the class of holonomic functions which are solutions of systems of linear difference-differential equations with polynomial coefficients. Holonomic functions are interesting since they can be uniquely determined by a finite number of informations, namely the coefficients of the equations and a finite number of initial conditions. Holonomic systems were introduced and studied by Bernstein and they form the foundation of Zeilberger's approach to prove special function identities using the method of creative telescoping. Later, Frédéric Chyzak extended this approach to the closely related class of ∂-finite functions which concerns all kinds of operators and not only differential and difference operators and allows the algorithmic execution of their closure properties like addition, multiplication and certain substitutions.

In his holonomic systems approach, Doron Zeilberger used an adaptation of Sylvester's dialytic elimination method which is time and space consuming. Finally, he stated that the use of Buchberger's concept of Gröbner bases in the noncommutative context of operator algebras may make it possible to perform this elimination much faster.

Since the work of Zeilberger in the early 1990s, many packages performing noncommutative Gröbner basis computations were implemented in different computer algebra systems, for instance, *Ncpoly* in *Reduce* (by Herbert Melenk), *Groebner* in Maple and *Plural* in Singular (by Viktor Levandovskyy). The Maple package is rather designed for computations in operator algebras (Ore algebras), whereas *Ncpoly* and *Plural* deal with more general algebras. This progress opened the door to the exploration of noncommutative Gröbner basis applications in different fields of mathematics.

This thesis deals with the exploration of elimination via Gröbner bases techniques and its applications in the special case of Ore algebras.

To carry out the main concepts of Gröbner bases theory from commutative to noncommutative algebras, the latter should satisfy certain conditions which make the computability of Gröbner bases possible. Towards an implementation in a computer algebra system, there are at least two approaches of the conditions that these algebras should satisfy. The first approach gives rise to the algebras of solvable type which were introduced by Kandri-Rody and Weispfenning. Later, Heinz Kredel generalized their axiomatic description by allowing that the variables operate on the coefficients. The second approach was introduced by Viktor Levandovskyy and deals with G-algebras. These two approaches apparently seem to be different. However, they are complementary and this apparent difference comes from the application scopes of each. An important part of this thesis is dedicated, after examination of these approaches, to the presentation of Gröbner bases theory in a way taking advantage of each and fitting them to the context and the purposes of this thesis.

Relying on Zeilberger's holonomic system approach and the theoretical as well as the algorithmic extension of this approach made by Frédéric Chyzak, one objective of this thesis was the implementation in the computer algebra system *Singular*, using its kernel extension *Plural*, of a package in the spirit of *Mgfun* of *Maple*. This package should deal with the algorithmic aspect of ∂-finite and holonomic functions. The main reason that prevented the achievement of this task is that the implementation of Gröbner bases in *Plural* does not take in consideration rational noncommutativity, for instance rational Ore algebras which are crucial for the algorithmic treatment of the class of ∂-finite functions. For this reason, a part of this thesis is restricted to the reinvestigation of the notions of ∂-finite and holonomic functions focusing on the presentation of the main definitions, properties and algorithms related to them in an algebraic point of view of modules and graded modules.

The method of *creative telescoping* consists of computing recurrence or differential equations satisfied by sums or integrals of special functions. The idea of this method is originally due to Celine Fasenmyer and was brought into a more general setting by Doron Zeilberger. A part of this thesis is devoted to investigate and to test how far noncommutative Gröbner basis elimination techniques may be efficiently applied to perform this method in the general setting described by Zeilberger.

This thesis is organized as follows. Chapter 2 deals with Ore algebras, where an overview of their main properties are given and a special focus is made on their action on the algebras of functions. This allows the description of these functions through their annihilating ideals which is the framework of Gröbner bases techniques. Chapter 3 gives an overview on how the main concepts and results of Gröbner bases theory can be generalized from commutative to noncommutative algebras and illustrates with examples some applications of Gröbner elimination techniques in Ore algebras. Chapter 4 introduces the notions of ∂-finite and holonomic functions, where their main algebraic characterizations are presented and the algorithmic aspect of the computation of their closure properties is investigated. The need to introduce these two classes is explained at the end of this chapter. Chapter 5 presents algorithms for summation and integration of functions lying in the intersection of both classes of holonomic and ∂-finite functions and which make use of the method of creative telescoping.

Chapter 2

Ore Algebras

The theory of Ore algebras gives the opportunity to consider linear ordinary differential, difference, q-difference equations and mixed ones from a uniform point of view. This theory was proposed by Ore [Ore33], who described, in particular, a uniform theory of operator factorization which generalizes the differential case. In this dissertation we present algorithms which apply to summation, as well as integration, and may involve operators of different kinds. This fact makes the study of Ore algebras attractive since it not only allows to prove more general statements concerning operators of various kinds, but also gives the ability to design algorithms in a unified way in terms of Ore operators which are afterwards adjustable to specific forms of operators. Bronstein and Petkovšek used the theory of Ore algebras in computer algebra in [BP96], where a general-purpose algorithm for factorization in an arbitrary Ore algebra over a field was described. In the first section of this chapter, we present how polynomial Ore algebras are defined and give an overview of their main properties. In the second section, we are concerned with the action of such polynomial Ore algebras on algebras of functions. This action allows to consider those functions as algebraic objects through their annihilating ideals which are the framework of Gröbner bases techniques. Finally, in the last section, we describe the Euclidean algorithm in Ore Algebras and how it can be used for elimination purposes.

2.1 Ore extensions

Linear Ore operators generalize among other the differential and difference operators. In their action on a product of two elements of a given ring, they satisfy a **skew** Leibniz rule which is a generalization of the differential product rule. For a better perception of the ideas behind this generalization, let's first observe and analyze the differential and finite difference cases.
On one hand, the product rule for two functions f and g of a differential algebra $\mathbb{A}$ states

$$(fg)' = fg' + f'g. \tag{2.1}$$

For $\mathbb{A} = \mathbb{K}[x]$ we define the differential operator with respect to x by

$$D : \mathbb{A} \to \mathbb{A}, \quad g \mapsto D(g) = Dg = \frac{dg}{dx}.$$

For a given function $f \in \mathbb{A}$ we define the operators

$$f : \mathbb{A} \to \mathbb{A}, \quad g \mapsto f(g) = fg$$

and

$$D_f : \mathbb{A} \to \mathbb{A}, \quad g \mapsto D_f(g) = D_f g = \frac{df}{dx} g$$

which stand for the multiplication by f and $\frac{df}{dx}$, respectively. With this notation the product rule (2.1) is equivalent to

$$(D \circ f)(g) = (f \circ D)(g) + D_f(g) \tag{2.2}$$

which may be expressed in terms of operators to be applied to g as follows

$$Df = fD + D_f, \tag{2.3}$$

where the product represents the composition of operators. On the other hand, in the case of finite differences we have the following functional identity

$$(fg)(x+1) - (fg)(x) = f(x+1)[g(x+1) - g(x)] + [f(x+1) - f(x)]g(x) \tag{2.4}$$

which is equivalent to

$$(\Delta \circ f)(g) = (S_f \circ \Delta)(g) + (\Delta_f)(g), \tag{2.5}$$

where S, Δ, f, Sf and Δf denote the following operators

$$S : \mathbb{A} \to \mathbb{A}, \quad g \mapsto S(g) = Sg = g(x+1),$$

$$\Delta : \mathbb{A} \to \mathbb{A}, \quad g \mapsto \Delta(g) = \Delta g = g(x+1) - g(x) = (S-1)g,$$

$$f : \mathbb{A} \to \mathbb{A}, \quad g \mapsto f(g) = fg,$$

$$S_f : f : \mathbb{A} \to \mathbb{A}, \quad g \mapsto S_f(g) = f(x+1)g(x)$$

and

$$\Delta_f : \mathbb{A} \to \mathbb{A}, \quad g \mapsto \Delta_f(g) = [f(x+1) - f(x)]g(x).$$

In terms of operators to be applied to g the rule (2.5) is expressed as follows

$$\Delta f = S_f \Delta + \Delta_f. \tag{2.6}$$

In a similar way the shift operator satisfies the equation

$$[S(fg)](x) = f(x+1)g(x+1) \tag{2.7}$$

which reads in terms of operators to be applied to g as follows

$$Sf = S_f S \tag{2.8}$$

If we observe the commutation rules (2.3), (2.6) and (2.8), we note that they satisfy the following pattern

$$\partial f = \sigma(f)\,\partial + \delta(f). \tag{2.9}$$

Since we are essentially interested in linear operators, σ and δ should be also linear. Moreover, the composition of operators which we expressed as a product in the previous commutation rules must be associative, i.e.

$$(\sigma f)\,g = \sigma\,(fg)$$

which by applying (2.9) leads to

$$\sigma(f)\,\partial g + \delta(f)g = \sigma(f)\,\sigma(g)\,\partial + \sigma(f)\,\delta(g) + \delta(f)\,g = \sigma(fg)\,\partial + \delta(fg). \tag{2.10}$$

Equating the coefficients of ∂ on the left and on the right hand side of the second equation of (2.10), we deduce that σ should be a ring homomorphism. Furthermore, δ is linear and it is related to σ as follows

$$\delta(fg) = \sigma(f)\,\delta(g) + \delta(f)\,g. \tag{2.11}$$

Now, we are ready to formulate the general setting of Ore extension. Let $\mathbb{K}$ be a field and $\mathbb{A}$ a ring (**not necessarily commutative**) which is endowed with a $\mathbb{K}$-algebra structure. Moreover, we consider an **injective** ring endomorphism $\sigma : \mathbb{A} \to \mathbb{A}$, then

Definition 2.1 (σ-Derivation)
A derivation with respect to σ is a $\mathbb{K}$-**linear** map $\delta : \mathbb{A} \to \mathbb{A}$ satisfying the following **skew** Leibniz rule

$$\delta(ab) = \sigma(a)\,\delta(b) + \delta(a)b \quad \text{for any} \quad a, b \in \mathbb{A}. \tag{2.12}$$

Definition 2.2
The set of constants with respect to σ and δ is

$$\mathrm{Const}_{\sigma,\delta} := \{a \in \mathbb{A};\ \sigma(a) = a,\ \delta(a) = 0\}$$

which is a subfield of $\mathbb{K}$.

Example 2.3
Let $\mathbb{K}$ be any subfield of $\mathbb{C}$ and $\mathbb{A} = \mathbb{K}[x]$. We denote by $1_{\mathbb{A}}$ the identity map of $\mathbb{A}$.

1. If $\sigma = 1_{\mathbb{A}}$ then $\delta = \frac{d}{dx}$ and the rule (2.12) is the classical Leibniz rule for derivation w.r.t. x. In this case the pair $(\mathbb{A}, \delta)$ is called a **differential** $\mathbb{K}$-algebra.

2. Let σ be any injective endomorphism of $\mathbb{A}$, then it is easy to prove that the map $\delta_\alpha = \alpha(\sigma - 1_{\mathbb{A}})$ for $\alpha \in \mathbb{K}$ defines a σ-derivative called in the literature **inner derivation**. In the special case of $\alpha = 1$, σ induces a **difference** $\mathbb{K}$-algebra $(\mathbb{A}, \delta_1)$ with the associated difference operator denoted by $\theta = \sigma - 1_{\mathbb{A}}$.

The following lemma describes the relationship between σ and δ.

Lemma 2.4
Let $\mathbb{A}$ be a $\mathbb{K}$-algebra and δ a σ-derivation of $\mathbb{A}$.

i) If $\sigma \neq 1$, then there is an element $\alpha \in \mathbb{A}$ such that $\delta = \alpha(\sigma - 1)$.

ii) if $\delta \neq 0$, then there is an element $\beta \in \mathbb{A}$ such that $\sigma = \beta\delta - 1$

An **Ore extension** adds a new symbol ∂ to the base ring $\mathbb{A}$ to get a new polynomial ring

Definition 2.5 (Skew polynomial ring)
A univariate skew polynomial ring $\mathbb{S} = \mathbb{A}[\partial; \sigma, \delta]$ over $\mathbb{A}$ given by σ and δ is the ring of polynomials in ∂ over $\mathbb{A}$ with the usual polynomial addition and a multiplication given by the rule

$$\partial a = \sigma(a)\,\partial + \delta(a) \quad \text{for any} \quad a \in \mathbb{A}. \tag{2.13}$$

Remark 2.6

- An element $p(\partial) \in \mathbb{A}[\partial; \sigma, \delta]$ is of the form $p(\partial) = \sum_{i=0}^{m} a_i\,\partial^i$ with $a_i \in \mathbb{A}$ for $i = 1, \ldots, m$.

- If we define a degree function on $\mathbb{A}[\partial; \sigma, \delta]$ to be $\deg(p) = \max\{i; a_i \neq 0\}$, then this function should satisfy

  1. $\deg(p) \geq 0$ if $p \neq 0$ and $\deg(0) = -\infty$.
  2. $\deg(p + q) \leq \max(\deg(p), \deg(q))$.
  3. $\deg(pq) = \deg(p) + \deg(q)$.

  The third property of the degree function is ensured only if σ is an injective endomorphism of $\mathbb{A}$.

- The multiplication defined by (2.13) can be uniquely extended to an associative multiplication on monomials by

$$(a\,\partial^{\alpha})(b\,\partial^{\beta}) = (a\,\partial^{\alpha-1})(\partial b)\,\partial^{\beta} = (a\,\partial^{\alpha-1})(\sigma(b)\,\partial^{\beta+1} + \delta(b)\,\partial^{\beta}) \quad \text{for } \alpha > 0. \tag{2.14}$$

  Furthermore, let $A,\ B \in \mathbb{A}[\partial; \sigma, \delta]\backslash\{0\}$ with leading monomials $a\,\partial^{\alpha}$ and $b\,\partial^{\beta}$, respectively. Then, by (2.14), the leading monomial of AB is $a\,\sigma^{\alpha}(b)\,\partial^{\alpha+\beta}$.

- Let $\mathbb{A}$ be a **commutative** ring and we denote by $\mathbb{A}\langle\,\partial\rangle$ the free associative $\mathbb{A}$-algebra generated by ∂, then the skew ring $\mathbb{S} = \mathbb{A}[\partial; \sigma, \delta]$ is isomorphic to $\mathbb{A}\langle\,\partial\rangle/(\partial - \sigma\partial - \delta)$. Furthermore, if σ is the identity and δ is the trivial derivation 0, then $\mathbb{S} = \mathbb{A}[\partial; 1, 0]$ is the polynomial ring $\mathbb{A}[\partial]$.

- Let $\boldsymbol{\partial} = \{\partial_1, \ldots, \partial_n\}$ be a set of symbols, $\boldsymbol{\sigma} = \{\sigma_1, \ldots, \sigma_n\}$ a set of ring endomorphisms of $\mathbb{A}$ and $\boldsymbol{\delta} = \{\delta_1, \ldots, \delta_n\}$ the set of the corresponding σ_i-derivatives. We denote the multivariate or also iterated skew polynomial ring by $\mathbb{A}[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}] := \mathbb{A}[\partial_1; \sigma_1, \delta_1] \cdots [\partial_n; \sigma_n, \delta_n]$, whose elements are finite sums of the form $p(\boldsymbol{\partial}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^n} a_{\boldsymbol{\alpha}}\,\boldsymbol{\partial}^{\boldsymbol{\alpha}}$.

  In the rest of this dissertation we will mean by a skew polynomial ring (also $\mathbb{K}$-algebra) such an iterated skew polynomial ring.

Now we are interested in a subclass of skew polynomial rings, where we require other conditions between the operators $\boldsymbol{\partial}$, $\boldsymbol{\sigma}$ and $\boldsymbol{\delta}$. They are called Ore Algebras.

Definition 2.7 (Ore algebras)
A skew polynomial ring $\mathbb{A}[\boldsymbol{\partial};\boldsymbol{\sigma},\boldsymbol{\delta}] = \mathbb{A}[\partial_1;\sigma_1,\delta_1]\cdots[\partial_n;\sigma_n,\delta_n]$ is called **Ore** $\mathbb{K}$-algebra and denoted by $\mathbb{O}$ when

i) the σ_is and δ_js commute for any $1 \leq i, j \leq n$ with $i \neq j$

ii) and satisfy $\sigma_i(\partial_j) = \partial_j$ and $\delta_i(\partial_j) = 0$ for $i > j$.

Remark 2.8

- In the previous definition the conditions on σ_i's and δ_i's imply that the ∂_i's should also commute which means $\partial_i\partial_j = \partial_j\partial_i$ for all $1 \leq i, j \leq n$.
- In the general definition of Ore Algebras, it is not required that the ∂_i's commute and they satisfy the following commutator relations:

$$\partial_i\partial_j = \sigma(\partial_j)\partial_i + \delta_i(\partial_j).$$

  We opt for Definition 2.7 because it fits better to the kind of operator algebras that we will consider throughout this dissertation.
- We use $\mathbb{O}_p$ and $\mathbb{O}_r$ to denote **polynomial** ($\mathbb{A} = \mathbb{K}[x_1, \ldots, x_s]$) and **rational** ($\mathbb{A} = \mathbb{K}(x_1, \ldots, x_s)$) Ore Algebras, respectively.

Example 2.9 (Weyl algebras)
The Weyl algebras are the associative algebras of linear differential operators with polynomial coefficients denoted by $\mathcal{A}_r(\mathbb{C}) = \langle x_1, \ldots, x_r, D_{x_1}, \ldots, D_{x_r}\rangle$, where D_{x_i} denotes the partial derivative with respect to x_i. They are noncommutative and given by the relations $D_{x_i}x_i = x_iD_{x_i} + 1$. In fact they represent also a special case of polynomial Ore Algebras as follows

$$\mathcal{A}_r(\mathbb{C}) = \mathbb{K}[x_1, \ldots, x_r][D_{x_1}; 1, D_{x_1}]\cdots[D_{x_r}; 1, D_{x_r}].$$

Example 2.10 (Shift Ore algebras)
Let $\mathbb{A}$ be $\mathbb{K}[n_1, \ldots, n_r]$ or $\mathbb{K}(n_1, \ldots, n_r)$ with $n_i \in \mathbb{Z}$ for $1 \leq i \leq r$. Moreover, by S_{n_i} we denote the forward shift operator defined by $S_{n_i}f(n_i) = f(n_i + 1)$ which satisfies in terms of operators the relation $S_{n_i}n_i = n_iS_{n_i} + S_{n_i}$. Hence, the corresponding Shift Ore algebra is $\mathbb{A}[S_{n_1}; S_{n_1}, 0]\cdots[S_{n_r}; S_{n_r}, 0]$.

Example 2.11 (q-Shift Ore algebra)
Let $\mathbb{A}$ be $\mathbb{K}(q_1, \ldots q_r, x_1, \ldots, x_r)$. We denote by Q_i the q-shift operator defined by $Q_if(x) = f(qx)$ which in terms of operator satisfies the relation $Q_ix = qxQ_i$. Hence, the corresponding Ore algebra is $\mathbb{A}[Q_1; Q_1, 0]\cdots[Q_r; Q_r, 0]$.

Table 2.1: Examples of Ore operators

$\mathbb{A}$	Case	σ	δ
$\mathbb{Q}(x)$	Differential	1	$\frac{d}{dx}$
	Eulerian	1	$x\frac{d}{dx}$
	Shift	S	0
	Difference	S	$S-1$
$\mathbb{Q}(q,x)$	q-shift	Q	0
	q-difference	Q	$Q-1$
	q-differential	Q	$\frac{Q-1}{x(q-1)}$

2.2 Action of Ore algebras

Let $\mathcal{F}$ be a $\mathbb{K}$-algebra of functions (e.g., sequences, power series etc) and $\mathbb{A}$ a subalgebra of $\mathcal{F}$. In this section, we describe the action of the Ore algebra $\mathbb{O} = \mathbb{A}[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$ on $\mathcal{F}$. In other words, how a multiplication of an element of $\mathbb{O}$ by an element $\mathcal{F}$ should be defined so that $\mathcal{F}$ turns into an $\mathbb{O}$-module. This module structure allows to see an element $f \in \mathcal{F}$ as an algebraic object by means of the ideal in $\mathbb{O}$ generated by the annihilating operators of f.

Definition 2.12 (Left-module)
A left R-module M over a ring R is an **abelian group** $(M,+)$ with an operation $\cdot : R \times M \to M$ called **scalar multiplication** such that for all r, s in R and x, y in M we have

1. $r\cdot(x+y) = r\cdot x + s\cdot y$.
2. $(r+s)\cdot x = r\cdot\, x + s\cdot x$.
3. $(rs)\cdot x = r\cdot(s\cdot x)$.
4. $1_R\cdot x = x$ if R has a multiplicative identity 1_R.

Depending on which operation we want to emphasize in $\mathcal{F}$ we may define the action of $\mathbb{O}$ on $\mathcal{F}$ as follows.

Definition 2.13
Let $\mathcal{F}$, $\mathbb{A}$ and $\mathbb{O}$ be as before. The operation $\cdot : \mathbb{O} \times \mathcal{F} \to M$ given by

$$\begin{aligned} a\cdot f &= af \quad \text{for all} \quad a \in \mathbb{A} \\ \partial_i\cdot f &= \delta(f) \quad \text{if} \quad \delta \neq 0 \\ \partial_i\cdot f &= \sigma(f) \quad \text{if} \quad \delta = 0 \end{aligned}$$

defines a scalar multiplication acting by composition of operators which gives a left-module structure to $\mathcal{F}$.

The interpretation of an algebra of functions $\mathcal{F}$ as $\mathbb{O}$-module motivates the study of the ideals of the Ore algebra $\mathbb{O}$. We are especially interested in the annihilating ideal of an element $f \in \mathcal{F}$ which is defined as follows.

Definition 2.14
Let $\mathcal{F}$ be a $\mathbb{K}$-algebra of functions, $\mathbb{O} = \mathbb{A}[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$ an Ore algebra acting on $\mathcal{F}$ by a scalar multiplication "$\bullet$" and $f \in \mathcal{F}$, then the set

$$\mathrm{Ann}_{\mathbb{O}}(f) = \{P \in \mathbb{O};\ P \bullet f \equiv 0\}$$

is an ideal in $\mathbb{O}$ called **annihilating ideal** of f.

Remark 2.15

- By means of the ideal $\mathrm{Ann}_{\mathbb{O}}(f)$ we correspond to f an algebraic structure that characterizes it and enables its interpretation as an algebraic object. Moreover, this ideal structure plays a crucial role in the application of Groebner bases techniques to answer questions (sometimes coming form analysis) related to f, for instance, to verify or prove identities satisfied by f. In most algorithms that we present in this dissertation, we are concerned among others with the determination, using noncommutative Groebner bases methods, of a set of generators of $\mathrm{Ann}_{\mathbb{O}}(f)$ satisfying special properties.

- The action of $\mathbb{O}$ on a function f defines a left **submodule** $\mathbb{O}\bullet f$ of $\mathbb{O}$ which is **isomorphic** to $\mathbb{O}/\mathrm{Ann}_{\mathbb{O}}(f)$.

- All the algorithms that we present in this dissertation do not necessarily require the concrete existence of an algebra $\mathcal{F}$. They may be formulated at the level of ideals $\mathcal{I}$ of $\mathbb{O}$ and submodules $\mathbb{O}/\mathcal{I}$, where the role of f is played by the generator 1 of $\mathbb{O}/\mathcal{I}$.

Example 2.16 (Legendre polynomials)
A class of orthogonal polynomials are the Legendre polynomials

$$P_n(x) = \sum_{k=0}^{n} \binom{n}{k} \binom{-n-1}{k} \Big(\frac{1-x}{2}\Big)^k. \tag{2.15}$$

They are elements of the algebra of formal power series $\mathcal{F} = \mathbb{Q}(n)[[x]]$ and solutions of a second order **differential equation**

$$(x^2-1)P_n''(x) + 2xP_n'(x) - n(n+1)P_n'(x) = 0, \tag{2.16}$$

a second order **recurrence equation**

$$(n+2)P_{n+2}(x) - (2n+3)xP_{n+1}(x) + (n+1)P_n(x) = 0, \tag{2.17}$$

and a mixed **difference-differential equation**

$$(x^2-1)P_{n+1}'(x) - (1+n)(xP_{n+1}(x) - P_n(x)) = 0. \tag{2.18}$$

If we act on $\mathcal{F}$ the Ore algebra

$$\mathbb{O}_r = \mathbb{A}[S_n; S_n, 0][D_x; 1, D_x] \quad \text{with} \quad \mathbb{A} = \mathbb{C}(n, x),$$

then the operators

$$G_1 = (x^2 - 1)D_x^2 + 2xD_x - n(n+1), \tag{2.19}$$

$$G_2 = (n+2)S_n^2 - (2n+3)xS_n + (n+1), \tag{2.20}$$

$$G_3 = (x^2 - 1)S_nD_x - (1+n)(xS_n - 1). \tag{2.21}$$

are elements of $\text{Ann}_{\mathbb{O}_r}(P_n)$ and translate the above equations in terms of operators. Furthermore, with the initial conditions

$$P_0(0) = 1, \quad P_1(0) = 0, \qquad P_0'(0) = 0, \quad P_1'(0) = 1 \tag{2.22}$$

the operators G_1 and G_2 describe the Legendre polynomials **uniquely**.

2.3 Skew Euclidean Division

We have seen in Remark 2.6 that in a skew ring $\mathbb{S} = \mathbb{A}[\partial; \sigma, \delta]$ a degree function is well defined, since the endomorphism σ is injective. Such a degree function makes possible to define an **Euclidean division** and thereafter to compute the greatest common divisor or the least common multiple of two elements of $\mathbb{S}$. The main results of this section are due to Oystein Ore [Ore33] and the explicit formulation of the Euclidian algorithm was made by Frédéric Chyzak in [CS96]. Firstly, we consider the Euclidean division in the special case of $\mathbb{S} = \mathbb{K}[\partial; \sigma, \delta]$, where the field $\mathbb{K}$ may be noncommutative and additionally satisfies a certain computability condition. Secondly, we present a generalization to the case of $\mathbb{S} = \mathbb{A}[\partial; \sigma, \delta]$.

Definition 2.17 (Skew field)
A field $\mathbb{K}$ which may be noncommutative is called **skew** if the following two conditions are satisfied.

a) The usual ring operations (addition and multiplication) are computable.

b) For all $0 \neq \alpha, \beta \in \mathbb{K}$ we can compute α', β' such that $\alpha'\alpha - \beta'\beta = 0$.

In the commutative case, the second condition is obtained by taking, for example, $\alpha' = \beta$ and $\beta' = \alpha$.

Let $\mathbb{S} = \mathbb{K}[\partial; \sigma, \delta]$ be a skew ring over an effective field and A and B two polynomials in $\mathbb{S} \setminus \{0\}$ such that

$$A = a\,\partial^\alpha + \text{ lower order terms w.r.t. } \partial \quad \text{with } a \neq 0 \tag{2.23}$$

$$B = b\,\partial^\beta + \text{ lower order terms w.r.t. } \partial \quad \text{with } b \neq 0. \tag{2.24}$$

Moreover, w.l.g. we suppose that $\alpha \geq \beta$ and we denote $\partial^\gamma = \partial^{\alpha-\beta}$. By a left multiplication of B by ∂^γ we get a new polynomial with leading monomial (see Remark 2.6-3) given by

$$\partial^\gamma b\, \partial^\beta = \sigma^\gamma(b)\, \partial^\alpha.$$

Moreover, since $\mathbb{K}$ is skew, we may compute u and v such that

$$ua - v\sigma^\gamma(b) = 0$$

and we get

$$uA - v\partial^\gamma B = R_0, \quad \text{where } R_0 \in \mathbb{S} \text{ and } \deg_\partial(R_0) < \alpha.$$

We apply recursively this division step to B and R_0 and so on until we obtain

$$uA = QB + R \quad \text{with } \deg_\partial(R) < \deg_\partial(B) \tag{2.25}$$

Definition 2.18 (Skew euclidean right division)
The process described above is called **skew Euclidean right division**, where Q and R denote the **right quotient** and the **right remainder** of A and B, respectively.

Remark 2.19

- Since $\mathbb{K}$ is a field, in the division step described previously, we may divide by u to get a more familiar expression of Euclidean division $A = QB + R$. However, we opt for the notation (2.25) which is also appropriate for the general case of a skew ring over a ring.
- We can analogously define a skew left Euclidean division by applying, in the division step, a right multiplication by ∂^γ to obtain $uA = BQ + R$.
- For A and B one can find the greatest common right divisor gcrd(A, B) by the right euclidean algorithm and least common left multiple lclm(A, B) by the extended Euclidean algorithm which also returns u and v such that gcrd$(A, B) = uA + vB$.
- The computation of the greatest common left divisor gcld and the least common right multiple lcrm can be reduced to the computation of the gcrd and the lclm by the **adjoint** which is presented in the following definition.

Definition 2.20
Let $\mathbb{S} = \mathbb{K}[\partial; \sigma, \delta]$ be a skew ring and we suppose that σ is an automorphism of $\mathbb{K}$. The adjoint of $\mathbb{S}$ is the skew ring $\mathbb{S}^* = \mathbb{K}[\partial; \sigma^*, \delta^*]$, where

$$\sigma^* = \sigma^{-1}, \qquad \delta^* = -\delta\sigma^{-1}. \tag{2.26}$$

Let $A = \sum_{i=0}^{n} a_i\partial^i \in \mathbb{S}$. The adjoint polynomial A^* is defined by the formula

$$A^* = \sum_{i=0}^{n} \partial^i a_i \in \mathbb{S}^* \tag{2.27}$$

Note that the product $\partial^i a_i$ must be computed in $\mathbb{S}^*$. It is easy to show that

$$Const_{\sigma,\delta}(\mathbb{K}) = Const_{\sigma^*,\delta^*}(\mathbb{K}), \quad (\sigma^*)^* = \sigma, \quad (\sigma^*)^* = \delta. \tag{2.28}$$

One can also verify that the adjoint is a linear (over $Const_{\sigma,\delta}(\mathbb{K})$) bijective mapping and

$$(A^*)^* = A, \qquad (AB)^* = A^*B^*. \tag{2.29}$$

Moreover,

$$\text{gcld}(A, B) = (\text{gcrd}(A^*, B^*))^*, \quad \text{lcrm}(A, B) = (\text{lclm}(A^*, B^*))^*. \tag{2.30}$$

Table 2.2: Examples of adjoint Ore operators

$\mathbb{A}$	Case	σ^*	δ^*
$\mathbb{Q}(x)$	Differential	1	$-\frac{d}{dx}$
	Eulerian	1	$-x\frac{d}{dx}$
	Shift	S^{-1}	0
	Difference	S^{-1}	$S^{-1}-1$
$\mathbb{Q}(q,x)$	q-shift	Q^{-1}	0
	q-difference	Q^{-1}	$Q^{-1}-1$
	q-differential	Q^{-1}	$\frac{Q^{-1}-1}{x(q-1)}$

We summarize the results of this part in the following theorem which was proved by Ore [Ore33] in the case of a commutative field $\mathbb{K}$ and is still true in the noncommutative case.

Theorem 2.21 (Oystein Ore)
Let A and B be two elements of the skew polynomial ring $\mathbb{S} = \mathbb{K}[\partial; \sigma, \delta]$ over an effective field $\mathbb{K}$, then the extended Euclidean algorithm computes polynomials u, v, D and $U, V \neq 0$ such that

$$uA + vB = D \qquad \text{and} \qquad UA + VB = 0, \tag{2.31}$$

where $D = \text{gcrd}(A, B)$, $UD = B$, $-VD = A$ and $UA = -VB = \text{lclm}(A, B)$.

Definition 2.22 (Left Ore ring)
A ring $\mathbb{A}$ is called a **left Ore ring** if the ideal intersection $\mathbb{A}a \cap \mathbb{A}b$ is not empty, that is, there exist at least two elements $0 \neq U, V$ in $\mathbb{A}$ such that $Ua = Vb$.

Remark 2.23

- In the sense of Definition 2.22 and according to Theorem 2.21, every skew polynomial ring over an effective field is also a left Ore ring.

The proof of Theorem 2.21 in [Ore33] leads to the following result

Corollary 2.24 (Oystein Ore)
If $\mathbb{A}$ is a left Ore ring, then $\mathbb{S} = \mathbb{A}[\partial; \sigma, \delta]$ is a left Ore ring as well.

Analogously to an effective field, we may define an effective Ore ring as follows

Definition 2.25 (Effective Ore ring)
A ring $\mathbb{A}$ is called effective if the usual ring operations are computable as well as the pair (U, V) involved in the equation (2.31).

Corollary 2.26
Let $\mathbb{S} = \mathbb{A}[\partial; \sigma, \delta]$ be a polynomial skew ring over an effective ring $\mathbb{A}$ and A and B two elements of $\mathbb{S}$. If there exist $(u, v) \in \mathbb{S}^2$ and $c \in \mathbb{A} \setminus \{0\}$ such that

$$uA + vB = c, \tag{2.32}$$

then the triple (u, v, c) may be computed by the extended Euclidean algorithm.

Remark 2.27

- It is obvious to note that the triple (u, v, c) may be computed only if A and B are coprime, that is $\text{gcrd}(A, B) = 1$.

In the following example we show how the Euclidean algorithm, throughout the results of Corollary 2.24 , may be applied for elimination purposes.

Example 2.28 (Euclidean algorithm and elimination)
We reconsider the annihilating operators of the Legendre polynomials involved in Example 2.16

$$\begin{aligned} G_1 &= (x^2-1)D_x^2 + 2xD_x + n(n-1) && (2.33) \\ G_2 &= (n+2)S_n^2 - (2n+3)xS_n + (n+1) && (2.34) \\ G_3 &= (x^2-1)D_xS_n - (n+1)(xS_n - 1) && (2.35) \\ &= ((x^2-1)D_x - (n+1)x)S_n + (n+1) && (2.36) \end{aligned}$$

which are elements of the left Ore ring $\mathbb{S} = \mathbb{Q}(n, x)[S_n; S_n, 0][D_x; 1, D_x]$. In what follows we explain how we may proceed by means of the Euclidean algorithm to eliminate the variable "S_n" between G_2 and G_3 to get a pure differential operator like G_1 (i.e., up to a multiple factor in $\mathbb{Q}(n, x)$). We consider G_2 and G_3 as polynomials in the left Ore ring

$$\mathbb{A}[S_n; 0, S_n], \quad \text{where } \mathbb{A} = \mathbb{Q}(n, x)[D_x; 1, D_x].$$

Since G_2 and G_3 are coprime, Corollary 2.26 states that we may compute, by application of the (extended) Euclidean algorithm, a polynomial $G \in \mathbb{A} \setminus \{0\}$ such that

$$uG_2 + vG_3 = G, \quad \text{where } u, v \in \mathbb{A}[S_n; 0, S_n].$$

Let's see how G may be concretely computed. We note that the polynomials G_2 and G_3 considered as polynomials in S_n with coefficients in $\mathbb{A}$ have degrees 2 and 1, respectively. First, we left multiply G_3 by S_n to get a polynomial with leading monomial of degree 2 in S_n and cancel it by means of skew division with the leading monomial in S_n of $G2$.

$$S_nG_3 = ((x^2-1)Dx - (n+2)x)S_n^2 + (n+2)S_n \tag{2.37}$$

and apply the (extended) Euclidean algorithm to get

$$\alpha G_2 + \beta S_n G_3 = C_1 S_n + C_2, \tag{2.38}$$

where

$$\begin{aligned} \alpha &= (x^2-1)D_x - (n+2)x, && (2.39)\\ \beta &= -(n+2), && (2.40)\\ C_1 &= -(2n+3)x(x^2-1)D_x + (n+1)((2n+3)x^2-(n+1)) \quad \text{and} && (2.41)\\ C_2 &= (n+1)(x^2-1)D_x - (n+2)(n+1)x && (2.42)\end{aligned}$$

are elements of $\mathbb{A} = \mathbb{Q}(n,x)[D_x;1,D_x]$ and $C_1S_n + C_2$ is a linear combination of G_2 and S_nG_3 of same degree in S_n as G_3.
In the next step we want to eliminate the variable "S_n" between G_3 and $C_1S_n + C_2$. We denote by $C_3 \in \mathbb{A}$ the leading coefficient of G_3 in S_n

$$C_3 = (x^2-1)D_x - (n+1)x. \tag{2.43}$$

Therefore, by applying once again the (extended) Euclidean algorithm but this time in $\mathbb{A}$ we obtain (see Theorem 2.21)

$$\alpha' C_1 + \beta' C_3 = 0, \quad \text{where } \alpha', \beta' \in \mathbb{A}. \tag{2.44}$$

Since $\mathrm{gcrd}(C_1, C_3) = 1$ then

$$\alpha' = C_3 \quad \text{and} \quad \beta' = -C1. \tag{2.45}$$

Finally, the polynomial

$$\alpha'(C_1S_n + C_2) + \beta' G_3 = \alpha\alpha' G_2 + (\alpha'\beta S_n + \beta')G_3 \tag{2.46}$$

is free of the variable "S_n" since the monomials involving "S_n" are canceled by the relation (2.44). Hence, the polynomial G we are looking for is

$$\begin{aligned} G &= (n+1)(x^2-1)^2D_x + 2(n+1)(x^2-1)xD_x - n(n+1)^2(x^2-1) && (2.47)\\ &= (n+1)(x^2-1)G_1. && (2.48)\end{aligned}$$

The previous example illustrates how the `skew_elim` command of the Maple package `Ore_algebra` internally works. In fact, `skew_elim(p,q,x,A)` tries to eliminate the indeterminate x between the skew polynomials p and q. It returns a nonzero polynomial $ap + bq$ free of x, if such a polynomial exists. Otherwise, a nonzero polynomial $ap + bq$ of least possible degree in x is returned.

Maple Session 2.1 (Skew elimination: Legendre polynomials $P_n(x)$)

We load the package.

```
  >  with(Ore_algebra):
```

We define the skew algebra where the skew elimination will be performed.

```
>   A:=skew_algebra(shift=[Sn,n],diff=[Dx,x]);
```

$$A := Ore\_algebra$$

We consider the following annihilating operators of the Legendre polynomials.

```
>   G1:=(x^2-1)*Dx^2+2*x*Dx-n*(n+1);
```

$$G1 := \left(x^2-1\right) Dx^2 + 2\,xDx - n\,(n+1)$$

```
>   G2:=(n+2)*Sn^2-(2*n+3)*x*Sn+(n+1);
```

$$G2 := (n+2)\,Sn^2 - (2\,n+3)\,xSn + n + 1$$

```
>   G3:=(x^2-1)*Dx*Sn-(n+1)*x*Sn +n+1;
```

$$G3 := \left(x^2-1\right) Dx\;Sn - (n+1)\,xSn + n + 1$$

Now, we apply `skew_elim` to $G1$ and $G3$ to deduce up to a multiple factor the operator $G2$

```
>   collect(skew_elim(G1,G3,Dx,A),{Sn},factor);
```

$$(-n-2)\,Sn^2 + (2\,n+3)\,xSn - n - 1$$

which is the operator $-G2$.
Similarly, we can apply `skew_elim` to $G2$ and $G3$ to deduce up to a multiple factor the operator $G1$

```
>   collect(skew_elim(G2,G3,Sn,A),{Dx},factor);
```

$$-(x-1)\,(x+1)\,Dx^2 - 2\,xDx + n\,(n+1)$$

which is the operator $-G1$.

In the following Maple session, we show that skew elimination can be performed on more complicated orthogonal polynomials such as Jacobi polynomials $P_n^{(a,b)}(x)$, where additionally symbolic parameters a and b occur.

Maple Session 2.2 (Skew elimination: Jacobi polynomials $P_n^{(a,b)}(x)$)

The Jacobi polynomials may be expressed as a formal power series, where the summand is:

```
>   jacobiterm:=1/2^n*binomial(n+a,k)*binomial(n+b,n-k)*
>   (x-1)^(n-k)*(x+1)^k;
```

$$jacobiterm := \frac{\binom{n+a}{k}\binom{n+b}{n-k}\,(x-1)^{n-k}\,(x+1)^k}{2^n}$$

To compute an annihilating operator system of the Jacobi polynomials, we load the package `hsum15`.

```
>   read "hsum15.mpl";
      `Package "Hypergeometric Summation", Maple V - Maple 15`
     `Copyright 1998-2012, Wolfram Koepf, University of Kassel`
```

First we compute a recurrence equation

```
>   sumrecursion(jacobiterm,k,S(n));
```

$$\begin{aligned}&2\,(n+2)\,(n+a+2+b)\,(a+2+b+2\,n)\,S(n+2) - \\ &(2\,n+3+a+b)\,(xa^2+a^2+2\,bxa+4\,xan+6\,xa+8\,x+4\,xn^2+6\,bx-b^2+ \\ &b^2x+12\,xn+4\,bxn)S(n+1)+ \\ &2\,(4+2\,n+a+b)\,(n+b+1)\,(n+a+1)\,S(n)=0\end{aligned}$$

then a differential equation

```
>   sumdiffeq(jacobiterm,k,S(x));
```

$$(x-1)(x+1)\left(\frac{d^2}{dx^2}S(x)\right)+(2x+a+bx-b+xa)\frac{d}{dx}S(x)-n(n+a+1+b)S(x)=0$$

and finally a derivative rule

```
>  sumdiffrule(jacobiterm,k,S(n,x));
```

$$\frac{\partial}{\partial x}S(n,x)=-\frac{(n+a+1+b)(xa+a+2x+bx+2xn-b)S(n,x)}{(x-1)(x+1)(a+2+b+2n)}+2\frac{(n+1)(n+a+1+b)S(n+1,x)}{(x-1)(x+1)(a+2+b+2n)}$$

In operator notation, the above equations lead to the following operator system:

```
>  DE := (1-x^2)*Dx^2+(b-a-(a+b+2)*x)*Dx+n*(n+a+b+1);
```

$$DE := \left(1-x^2\right)Dx^2+(b-a-(2+a+b)x)Dx+n(n+a+1+b)$$

```
>  RE :=  2*(n+2)*(n+a+2+b)*(a+2+b+2*n)*Sn^2-
>  (2*n+3+a+b)*(x*a^2+a^2+4*x*a*n+2*x*a*b+6*x*a+4*x*n*b+4*x*n^2+
>  6*b*x-b^2+8*x+b^2*x+12*x*n)*Sn 2*(2*n+a+4+b)*(n+b+1)*(n+a+1);
```

$$\begin{aligned}RE := {}& 2\,(n+2)(n+a+2+b)(a+2+b+2n)Sn^2-\\ &(2n+3+a+b)(xa^2+a^2+2bxa+4xan+6xa+8x+4xn^2+6bx-b^2+\\ &b^2x+12xn+4bxn)Sn+2\,(4+2n+a+b)(n+b+1)(n+a+1)\end{aligned}$$

and by multiplying the derivative rule by the common dominator we get

```
>  DR := (x-1)*(x+1)*(b+2+a+2*n)*Dx-2*(n+1)*(n+a+b+1)*Sn+
>  (n+a+b+1)*(2*n*x+x*a+x*b+2*x+a-b);
```

$$\begin{aligned}DR := {}&(x-1)(x+1)(a+2+b+2n)Dx-2\,(n+1)(n+a+1+b)Sn+\\ &(n+a+1+b)(xa+a+2x+bx+2xn-b)\end{aligned}$$

We load the package `Ore_algebra` and define the skew algebra in which the skew elimination will be performed.

```
>  with(Ore_algebra):
>  A:=skew_algebra(shift=[Sn,n],diff=[Dx,x],polynom={a,b}):
```

We apply `skew_elim` to the operators RE and DR to get:

```
>  G1:=collect(skew_elim(RE,DR,Sn,A),{Dx},factor);
```

$$\begin{aligned}G1 := {}&(x-1)(x+1)(a+b+4+2n)(a+2+b+2n)^2Dx^2+\\ &(a+b+4+2n)(a+2+b+2n)^2(2x+xa+a+bx-b)Dx-\\ &n(a+b+4+2n)(n+a+1+b)(a+2+b+2n)^2\end{aligned}$$

which is the operator DE up to the multiple factor

```
>  normal(G1/DE);
```

$$-(a+b+4+2n)(a+2+b+2n)^2$$

Applying `skew_elim` to the operators DE and DR leads to:

```
>  G2:=collect(skew_elim(DE,DR,Dx,A),{Sn},factor);
```

$$\begin{aligned}G2 := {}&2\,(n+2)(a+2+b+2n)(n+a+2+b)(n+a+1+b)Sn^2-\\ &(2n+3+a+b)(n+a+1+b)(xa^2+a^2+6xa+2xab+\\ &4xan+8x+4xn^2+4xnb+6bx+b^2x-b^2+12xn)Sn+\\ &2\,(n+b+1)(n+a+1)(a+b+4+2n)(n+a+1+b)\end{aligned}$$

which is the operator RE up to the multiple factor $(n+a+b+1)$.

Chapter 3

Noncommutative Gröbner Bases

In this chapter, we will give, based on the works of H. Kredel [Kre93], F. Chyzak [Chy98] and V. Levandovskyy [Lev05], an overview on how the main concepts and results of Gröbner bases theory may be generalized from commutative to non commutative algebras. This generalization concerns the so called *solvable algebras* (also denoted by G-algebras in [Lev05]) which is a class of algebras satisfying certain conditions making Gröbner basis computable.
First, we will give an overview on characterization and main properties of solvable algebras. Then, we will show how Buchberger's algorithm may be carried out to these algebras. Finally, we will give some applications of Gröbner elimination techniques in Ore algebras which is a subclass of solvable algebras.

3.1 Algebras of Solvable Type

3.1.1 Notations and Preliminaries

Let $\mathbb{K}$ be a skew field, that is a not necessarily a commutative field. Let R be the polynomial ring $R = \mathbb{K}[x_1, \ldots, x_n]$ over the field $\mathbb{K}$ in the commuting variables $x_1, \ldots, x_n$ for some $n \in \mathbb{N}, n \geq 0$. All elements of $\mathbb{K}$ are assumed to commute with the variables $x_1, \ldots, x_n$ but $\mathbb{K}$ need not to be itself commutative.

Definition 3.1
Let T denote the set of terms (power-products of the variables) in the variables $x_1, \ldots, x_n$

$$T := \{x^e = x_1^{e_1} \ldots x_n^{e_n}; e_i \in \mathbb{N}, 1 \leq i \leq n\}.$$

For a subset $X \subset \{x_1, \ldots, x_n\}$ we denote by $T(X)$ the set of terms in the variables of X, and for an element $f \in R$ we denote by $T(f)$ the set of terms occurring in f with non-zero coefficient.

Definition 3.2

(i) Let $\mathbb{N}^n$ denote the set of all n-tuples of natural numbers. A **componentwise** partial order $\leq_n$ on $\mathbb{N}^n$ is defined as follows

$$\text{for } e = (e_1, \ldots, e_n), e' = (e'_1, \ldots, e'_n) \in \mathbb{N}^n, e \leq_n e' \;\Leftrightarrow\; e_i \leq e'_i \text{ , for all } 1 \leq i \leq n.$$

(ii) Let x^e, x^f be two terms of T. We say that x^e divides x^f and denote $x^e \mid x^f$ if there exists $x^{e'} \in T$ sucht that $x^f = x^{e'}x^e$ which is equivalent to $e \leq_n f$.

An important property of the divisibility relation of terms and the componentwise order of $\mathbb{N}^n$ is known as Dickson's lemma (see [BKW93]) which is fundamental for most termination proofs of polynomial algorithms.

Lemma 3.3 (Dickson 1913)
Let $\mathbb{N}^n$ be endowed with the componentwise order $\leq_n$. Then every subset M of $\mathbb{N}^n$ has a finite subset B of $\mathbb{N}^n$ such that for every $m = (m_1, \ldots, m_n) \in M$, there exists $b = (b_1, \ldots, b_n) \in B$ with $b \leq_n m$.

There is a bijection $e : T \to \mathbb{N}^n$ which describes a one-to-one correspondence between the elements of T and their exponents in $\mathbb{N}^n$, defined by $e(x_1^{e_1} \ldots x_n^{e_n}) = (e_1, \ldots, e_n)$. By means of this bijection and Definition 3.2-(ii) one may prove the following corollary.

Corollary 3.4
Let T be the set of terms partially ordered by divisibility $\mid$. Then every subset S of T has a finite subset V of T such that for every $s \in S$ there exists $v \in V$ with $v \mid s$.

Definition 3.5 (Admissible order)
A linear order $\prec$ on the set of terms T is called admissible if for all $r, s, t \in T$

(i) $1 \prec r$.

(ii) $r \prec s \implies rt \prec st$ (compatible with term-multiplication).

Definition 3.6
Let $\prec$ be an admissible order on T. Any non-zero element $f \in R = \mathbb{K}[x_1, \ldots, x_n]$ can be written uniquely as $f = cx^e + g$, with $c \neq 0$ and $x^{e'} \prec x^e$ for any non-zero monomial $c'x^{e'}$ of g. We define

$$\begin{aligned} \mathrm{lt}(f) &= x^e, && \text{the **leading term** of } f \\ \mathrm{lm}(f) &= cx^e, && \text{the **leading monomial** of } f, \\ \mathrm{lc}(f) &= c, && \text{the **leading coefficient** of } f, \\ \mathrm{le(f)} &= e, && \text{the **leading exponent** of } f. \end{aligned}$$

3.1.2 Definition and Properties of Solvable Algebras

Kandri-Rody and Weispfenning in [KRW90] introduced a polynomial solvable algebra as a polynomial ring $R = \mathbb{K}[x_1, \ldots, x_n]$ equipped with a new noncommutative multiplication $\star$ satisfying certain axioms. These axioms guarantee that the difference between $x_i \star x_j$ and a suitable scalar multiple of x_ix_j is smaller than x_ix_j w. r. t. an arbitrary but fixed admissible term order. In their approach, they supposed that the field $\mathbb{K}$ is commutative and the elements of $\mathbb{K}$ commute with the variables $x_1, \ldots, x_n$. Heinz Kredel in [Kre93] generalized the axiomatic description of solvable algebras in which he allowed that the variables operate on the coefficients, by requiring an additional condition on the product $x_i \star a$, where $a \in \mathbb{K}$. By this condition, the field $\mathbb{K}$ need not to be commutative and its elements should not commute with the variables $x_1, \ldots, x_n$.

Definition 3.7 (Solvable polynomial algebra)
Let $\mathbb{K}$ be a skew field and $R = \mathbb{K}[x_1, \ldots, x_n]$. For a fixed term order $\prec$ on the set of terms T, $(R, \star)$ is called **solvable polynomial algebra** for an admissible term order $\prec$ if $\star$ satisfies the following axioms:

(i) $(R, \star)$ is an associative ring with 1.

(ii) (a) For all $a, b \in \mathbb{K}$, $t \in T(x_1, \ldots, x_n)$, $a \star b \star t = a \star (bt) = (ab)t = abt$.
  (b) For all $1 \leq i \leq n$, $s \in T(x_1, \ldots, x_i)$, $t \in T(x_i, \ldots, x_n)$, $s \star t = st$.

(iii) For all $1 \leq i < j \leq n$ there exist $0 \neq c_{ij} \in \mathbb{K}$ and $d_{ij} \in R$ with $\operatorname{lt}(d_{ij}) \prec x_i x_j$ such that

$$x_j \star x_i = c_{ij} x_i x_j + d_{ij}. \tag{3.1}$$

(iv) For all $1 \leq i \leq n$ and all $\neq a \in \mathbb{K}$ there exist $0 \neq c_{ai} \in \mathbb{K}$ and $d_{ai} \in \mathbb{K}$ such that

$$x_i \star a = c_{ai} a x_i + d_{ai}. \tag{3.2}$$

Solvable polynomial algebras will be denoted by $\mathbb{K}\{x_1, \ldots, x_n; Q; Q'\}$, where Q and Q' denote the *commutation rules* (3.1) and (3.2), respectively.

Remark 3.8

- Any admissible order $\prec$ satisfying condition (iii) of Definition 3.7 is called **$\star$-compatible**.
- If we assume that the elements of $\mathbb{K}$ commute with the variables $x_1, \ldots, x_n$, that is in Q' $c_{ai} = 1$ and $d_{ai} = 0$, then we drop Q' and denote the solvable algebra by $\mathbb{K}\{x_1, \ldots, x_n; Q\}$.
- The set of all terms $x_{i_1} \star \cdots \star x_{i_k}$ with $i_1 \leq i_2 \leq \cdots \leq i_k$ is a $\mathbb{K}$-basis of $\mathbb{K}\{x_1, \ldots, x_n; Q; Q'\}$ and the map $x_{i_1} \star \cdots \star x_{i_k} \mapsto x_1 x_2 \cdots x_k$ is a $\mathbb{K}$-automorphism mapping the new basis into the standard basis.
- In the Definition 3.7 it is required for $\mathbb{K}$ to be a skew field. We may make this requirement less restrictive and define solvable polynomial algebras over any domain $\mathcal{R}$ satisfying the following condition: there exists a subfield $\mathbb{L} \subset \operatorname{cen}(\mathcal{R}) := \{a \in \mathcal{R};\ ar = ra,\ \text{for all } r \in \mathcal{R}\}$ such that all c_{ij}, c_{ai} are contained in $\mathbb{L}$. In other words, the c_{ij} and c_{ai} must be invertible and commute with all elements of $\mathcal{R}$. A solvable algebra $S = \mathcal{R}\{x_1, \ldots, x_n; Q, Q'\}$, over a ring $\mathcal{R}$ satisfying the previous conditions, is called **centred commutation rules**.

Definition 3.9
Let $\prec$ be the lexicographical order T defined by $x_1 < x_2 < \cdots < x_n$ and $S = \mathbb{K}\{x_1, \ldots, x_n; Q; Q'\}$ be a solvable algebra w. r. t. $\prec$. S is called of **strictly lexicographical type** if for the commutator relations Q in Definition 3.7 we have $d_{ij} \prec x_j$, that is $d_{ij} \in \mathbb{K}[x_1, \ldots, x_j]$. Furthermore, if $c_{ij} = 1$ then S is called of **strictly monic lexicographical type**.

Definition 3.7 describes the computation of $\star$-products of variables and coefficients, and now we are interested in the extension of this computation to arbitrary polynomials of R. The following lemma determines left multiplication with field elements and right multiplication by special terms.

Lemma 3.10
Let $S = \mathbb{K}\{x_1, \dots, x_n; Q; Q'\}$ be a solvable algebra.

1. Let $a \in \mathbb{K}$ and $f \in S$, then $a \star f = af$.
2. Let $1 \leq i \leq j \leq k \leq n$ and $f \in \mathbb{K}[x_i, \dots, x_j]$, $t \in T(x_j, \dots, x_k)$, then
$$f \star t = ft \in \mathbb{K}[x_i, \dots, x_k].$$

The following lemma considers products of polynomials with increasing set of variables

Lemma 3.11
Let $S = \mathbb{K}\{x_1, \dots, x_n; Q; Q'\}$ be a solvable algebra, $\prec$ be a $\star$-compatible admissible term order, $1 \leq i \leq j \leq k \leq n$, $f \in \mathbb{K}[x_i, \dots, x_j]$ and $g \in \mathbb{K}[x_i, \dots, x_k]$, then there exists $0 \neq c \in \mathbb{K}$ and $h \in \mathbb{K}[x_i, \dots, x_k]$ with $\mathrm{lt}(h) \prec \mathrm{lt}(fg)$ and such that
$$f \star g = cfg + h.$$
In particular $f \star g \in \mathbb{K}[x_i, \dots, x_k]$ and $\mathrm{lt}(f \star g) = \mathrm{lt}(fg)$

Remark 3.12

- The result of the previous lemma still true in the special case of $g = a \in \mathbb{K}$ and we get: $f \star a \in \mathbb{K}[x_i, \dots, x_j]$ and $\mathrm{lt}(f \star a) = \mathrm{lt}(f)$.
- If the coefficients commute with the variables, then $f \star g = fg$ and $f \star g \in \mathbb{K}[x_i, \dots, x_k]$.

Lemma 3.10 and Lemma 3.11 and a constructive proof by case distinction leads to the following result which describes the $\star$-multiplication of arbitrary two polynomials in $S = \mathbb{K}\{x_1, \dots, x_n; Q; Q'\}$.

Proposition 3.13
Let $S = \mathbb{K}\{x_1, \dots, x_n; Q; Q'\}$ be a solvable algebra, $\prec$ be a $\star$-compatible admissible term order and $f, g \in S$, then there exists $h \in S$ and $0 \neq c \in \mathbb{K}$ such that
$$f \star g = cfg + h \quad \text{and} \quad \mathrm{lt}(h) \prec \mathrm{lt}(fg).$$
Moreover, c and h are uniquely determined by f and g.

Corollary 3.14
Let $S = \mathbb{K}\{x_1, \dots, x_n; Q; Q'\}$ be a solvable algebra and $\prec$ be a $\star$-compatible admissible term order, then the $\star$-multiplication is uniquely determined by Q and Q'.

The following lemma plays a crucial role in a straightforward translation of many tools and properties of Gröbner basis theory from the classic commutative case to the case of solvable algebras (see Section 3.2). In this sense, solvable algebras are intermediate between general noncommutative free associative algebras and commutative polynomial algebras.

Lemma 3.15
Let $S = \mathbb{K}\{x_1, \ldots, x_n; Q; Q'\}$ be a solvable algebra, $\prec$ be a $\star$-compatible admissible term order and $f, g \in S$. Then

1. $\mathrm{lt}(f \star g) = \mathrm{lt}(fg) = \mathrm{lt}(f)\mathrm{lt}(g) = \mathrm{lt}(g)\mathrm{lt}(f) = \mathrm{lt}(gf) = \mathrm{lt}(g \star f)$,
2. there exists $0 \neq c \in \mathbb{K}$, such that $\mathrm{lm}(f \star g) = c\mathrm{lm}(f)\,\mathrm{lm}(g)$,
3. for $h \in S$, $\mathrm{lt}(f) \prec \mathrm{lt}(g)$ implies $\mathrm{lt}(f \star h) \prec \mathrm{lt}(g \star h)$ and $\mathrm{lt}(h \star f) \prec \mathrm{lt}(h \star g)$.

From the previous lemma follows immediately.

Corollary 3.16
Let $S = \mathbb{K}\{x_1, \ldots, x_n; Q; Q'\}$ be a solvable algebra and $\prec$ be a $\star$-compatible admissible term order, then S is an integral domain.

Remark 3.17 (Associativity)
The conditions (ii)-(iv) of Definition 3.7 alone do not guarantee the associativity of the $\star$-product. For this aim we need rather complicated conditions on Q. These conditions were implicitly described by Kredel in [Kre93, Proposition 3.3.6], and explicitly determined by Viktor Levandovskyy in [Lev05] who called them **non-degeneracy conditions**.

Remark 3.18 (Computability)
Computability means the existence of a Turing machine which takes the appropriate inputs, performs a finite number of steps and terminates after producing the searched output.

- A field $\mathbb{K}$ is computable means that its basic operations (addition, multiplication and division) may be performed algorithmically. Moreover, the computability of $\mathbb{K}$ implies the computability of a commutative polynomial ring over $\mathbb{K}$.
- A term order $\prec$ is computable (or decidable) means that for any two terms $t, s \in T$ it can be decided by means of an algorithm if $s = t$ or $s \prec t$.
- One can compute in a solvable algebra if the commutator relations Q and Q' are computable. This requires that $\mathbb{K}$ is computable and the maps $a \mapsto c_{ai}a$ and $a \mapsto d_{ij}$ may be given algorithmically.

3.1.3 Examples of Solvable Algebras

Ore Algebras

Let $\mathbb{O} = \mathbb{A}[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}] = \mathbb{A}[\partial_1, \sigma_1, \delta_1] \cdots [\partial_n, \sigma_n, \delta_n]$ be an iterated Ore algebra (see Definition 2.7). A multiplication $\star$ on $\mathbb{O}$ is defined by (see Definition 2.5)

$$\begin{aligned} (Q') : \partial_i \star a &= \sigma_i(a)\, \partial_i + \delta_i(a) \quad \text{for any } a \in \mathbb{A} \text{ and } 1 \le i \le n \\ &= c_{ai} a\, \partial_i + d_{ai}. \end{aligned}$$

Moreover, the conditions required on the σ_i's and δ_i's imply the commutation of ∂_i's (see Remark 2.6), that is,

$$(Q) : \partial_i \star \partial_j = \partial_j \star \partial_i = \partial_i\, \partial_j \quad \text{for any } 1 \le i, j \le n.$$

It follows that $\{\mathbb{O}; Q, Q'\}$ is a solvable algebra w.r.t. any term order $\prec$.

Poincaré-Birkoff-Witt Extensions

In this subsection we give an example of a solvable algebra, where we require that $\mathbb{K}$ is commutative and the variables x_i do not act on the elements of $\mathbb{K}$. That is, this example fits rather into the framework developed by Kandry-Rody and Weispfenning [KRW90] than the general framework of Kredel [Kre93].
Bell and Goodearl [BG88] introduced the Poincaré-Birkoff-Witt extension (for short **PBW extension**) for two associative rings $\mathcal{R} \subseteq \mathcal{P}$. We formulate the definition for the special case where $\mathcal{R}$ is a field $\mathbb{K}$ and we additionally require that the variables commute with the coefficients.

Definition 3.19 (PBW extension)
Let $\mathbb{K}$ be a commutative field and $(\mathcal{P}, *)$ an associative ring with 1 such that $\mathbb{K} \subseteq \mathcal{P}$. The ring $\mathcal{P}$ is called a finite **PBW extension** (also PBW algebra) of $\mathbb{K}$ if there exist $x_1, \ldots, x_n \in \mathcal{P}$ such that

(i) the standard monomials $x^e = x_1^{e_1} \cdots x_n^{e_n}$ form a $\mathbb{K}$-basis of $\mathcal{P}$ as a $\mathbb{K}$-vector space,

(ii) $x_i \star a = a \star x_i$ for each $1 \le i \le n$, and any $a \in \mathbb{K}$.

(iii) $x_i \star x_j - x_j \star x_i = d_{ij} \in \mathbb{K}x_1 + \cdots + \mathbb{K}x_n$.

We note in Definition 3.19-(iii) that $\deg(d_{ij}) \le 1 < 2 = \deg(x_i x_j)$. If we define the commutator relation (Q) as follows:

$$(Q) : x_j \star x_i = x_i x_j + d_{ij},$$

then $(\mathcal{P}, Q)$ is of solvable type w.r.t. any **degree** term order $\prec_{\deg}$.

Remark 3.20

- A classical example of a PBW extension is the **universal enveloping algebra** $U(L)$ of a finite-dimensional Lie-algebra L over $\mathbb{K}$. In fact, $U(L)$ is a unitary associative $\mathbb{K}$-algebra containing L. Moreover, The Poincaré-Birkoff-Witt Theorem [Jac62] asserts that the standard monomials form a $\mathbb{K}$-basis of $U(L)$ which also explains the name PBW extension.

- In the general definition of a PBW extension, given by Bell and Goodearl in [BG88], it is not required in condition (ii) of Definition 3.19 that the variables commute with the coefficients. According to that definition, a PBW extension is not necessarily of solvable type. For example, the skew enveloping algebras $\mathcal{R}\#U(L)$ where $\mathcal{R}$ is a $\mathbb{K}$-algebra on which the elements of L act as derivations is not of solvable type (see [MR87, Sect. 1.7.10]).

3.1.4 Hilbert's Basis Theorem for Solvable Algebras

Hilbert's basis theorem is an important and well known property of commutative polynomial rings $\mathcal{R}[x_1, \ldots, x_n]$. It plays a central role in the theory of Gröbner bases, in the sense that it is the decision criterion for the termination of Buchberger's algorithm. In this subsection, we will see that solvable algebras satisfy this property under certain conditions.

Definition 3.21 (Noetherian ring)
A ring $\mathcal{R}$ is called a left (right) **Notherian ring** if it satisfies the following equivalent conditions.

ACC There does not exist an infinite ascending chain of left (right) ideals, that is, an infinite sequence of left (right) ideals such that each ideal is properly contained in its successor.

MAX Every non-empty family of left (right) ideals of $\mathcal{R}$ has a maximal element.

HIB Every left (right) ideal of $\mathcal{R}$ is finitely generated.

If the ring $\mathcal{R}$ is both left and right noetherian it is simply called noetherian.

In the following we describe Hilbert's basis theorem in the general context of solvable algebras over a ring $\mathcal{R}$ with **centered commutator relations** (see Remark 3.8).

Theorem 3.22 (Hilbert's basis theorem [Kre93])
If $\mathcal{R}$ is a left (right) noetherian ring, then any polynomial ring of solvable type $S = \mathcal{R}\{x_1, \ldots, x_n; Q, Q'\}$ with centred commutation rules, such that the maps $\sigma_i : a \mapsto c_{ai}a$ are automorphisms of $\mathcal{R}$, is left (right) noetherian.

Remark 3.23

- The condition that the maps $\sigma_i : a \mapsto c_{ai}a$ are automorphisms of $\mathcal{R}$ is necessary for S to be Noetherian. In fact, Mc Connel and Robson gave in [MR87] a concrete counter-example of a univariate solvable polynomial ring violating this condition and which is neither left nor right noetherian.

- Throughout this dissertation, we will consider operator algebras (Ore algebras) of solvable type (see Subsection 3.1.3) with bijective σ_i's (see Tables 2.1 and 2.2), thereby they are noetherian.

3.2 Gröbner bases in Solvable Algebras

Throughout this section let $S = \mathbb{K}\{x_1, \ldots, x_n; Q; Q'\}$ denote a solvable algebra over a computable field $\mathbb{K}$ with respect to a fixed but arbitrary admissible and decidable term order $\prec$. T denotes the set of terms in the variables $x_1, \ldots, x_n$ (see Definition 3.1) and $\mathbb{K}^* = \mathbb{K} \setminus \{0\}$. Furthermore, since we are essentially concerned with operator algebras (Ore algebras) acting by left multiplication, in this section we will present an overview of the theory of left Gröbner bases in solvable algebras. A theory of right (two-sided) Gröbner bases is analogously described in [Kre93] and [Lev05].

3.2.1 Left Reduction

One of the pillars of the commutative Buchberger algorithm is the reduction which describes "the multivariate division". In this section, we present its left-analogue in the case of solvable algebras.

Definition 3.24 (Left reduction)

- For $f, g, r \in S$, we say that f is **left reducible** to r modulo g if there exists a monomial $m = cx^\alpha$ with $c \in \mathbb{K}^*$ such that

  $$\mathrm{lm}(f) = \mathrm{lm}(m \star g) \quad \text{and} \quad r = f - m \star g.$$

  We note that $\mathrm{lt}(r) \prec \mathrm{lt}(f)$. Moreover, if no such a monomial m exists, then f is **left irreducible** modulo g.

- Let $G \subset S$ be a set of polynomials. An element f is left reducible modulo G if it is left reducible modulo an element $g \in G$.

Definition 3.25 (Notations)
Throughout this section we use the following notations

1. $f \longrightarrow_g r$ if f is left reducible to r modulo g.

2. $f \longrightarrow_G r$ if f is left reducible to r modulo G.

3. $f \overset{*}{\longrightarrow}_G r$, if $f = r$ or there exists a finite sequence $g_1, \ldots, g_s$ of elements of S such that

   $$f \longrightarrow_{g_1} \cdots \longrightarrow_{g_s} r.$$

   Furthermore, if r is irreducible modulo G, then it is called a **left normal form** of f with respect to G and denoted by $\mathrm{LNF}(f \mid G)$.

4. The **left ideal** generated by a set of polynomials $G \subset S$ is denoted by

   $$_S\langle G\rangle := \{f = \sum_{i=1}^{s} s_i \star g_i \; ; s_i \in S, g_i \in G\}$$

5. The left ideal generated by the leading terms of the elements of a subset $G \subset S$ is denoted by
$$L(G) := {}_S\langle\{\text{lt}(g)\ ; g \in G\}\rangle,$$
and is called **leading ideal** of G. Analogously, we denote by $L(I)$ the leading ideal of a left ideal $I \subset S$.

Remark 3.26

- A left normal form LNF$(f \mid G)$ it not necessarily unique, every different finite sequence $g_1, \ldots, g_s$ may lead to a different LNF$(f \mid G)$. In Lemma 3.30, we will see when a left normal form is unique.

It is easy to check that a left normal form satisfies the following properties.

Lemma 3.27

1. LNF$(0 \mid G) = 0$.
2. LNF$(f \mid G) \neq 0 \quad \Rightarrow \quad$ lt(LNF$(f \mid G)) \notin L(G)$.
3. $f -$ LNF$(f \mid G) \in {}_S\langle G\rangle$.

Definition 3.28 (Reduced set)

- A subset $G \subset S$ is called **minimal**, if $0 \notin G$ and $\text{lt}(g) \notin L(G \setminus \{g\})$ for each $g \in G$.
- We say that an element $f \in S$ is **completely left reduced** with respect to G, if no monomial of f is contained in $L(G)$.
- A subset $G \subset S$ is called **left reduced**, if $0 \notin G$, each $g \in G$ is reduced with respect to $G \setminus \{g\}$ and $g - \text{lc}(g)\text{lt}(g)$ is reduced with respect to G.

Definition 3.29

Let $G = \{g_1, \ldots, g_s\} \subset S$ be a finite set of polynomials. A representation of $f \in {}_S\langle G\rangle$
$$f = \sum_{i=1}^{s} s_i \star g_i$$
satisfying $\text{lt}(s_i \star g_i) \preceq \text{lt}(f)$ for all $i = 1, \ldots, s$, is called a **left standard representation** of f with respect to G.

Lemma 3.30

Let $G \subset S$ be a set of polynomials.

1. $f \xrightarrow{*}_G 0$ implies that f has a left standard representation with respect to G. In particular $f \in {}_S\langle G\rangle$.
2. If LNF$(f \mid G)$ is completely left reduced, then it is unique.

Proof:

1. follows immediately by construction from Definition 3.24 and Definition 3.25-(3).

2. Let $f \in S$ and assume that r and r' are two reduced normal forms of f with respect to G, then it follows that $r - r' = (f - r') - (f - r) \in {}_S\langle G \rangle$ (see Lemma 3.27). If $r - r' \neq 0$, then $\text{lt}(r - r') \in \text{L}(G)$, which contradicts the fact $\text{lt}(r - r')$ is a term of either r or r' which should not be contained in $\text{L}(G)$ (see Definition 3.28).

□

Definition 3.31
Let x^α and x^β be two terms of T. For each $1 \leq i \leq n$, set $\mu_i = \max(\alpha_i, \beta_i)$ and $\mu = (\mu_1, \dots, \mu_n)$. Then the **lcm** of x^α and x^β, denoted by $\text{lcm}(x^\alpha, x^\beta)$, is the term x^μ which is in particular divisible by x^α and x^β.

An important tool of Buchberger's algorithm are the so called S-polynomials. They can be analogously defined in the context of solvable algebras.

Definition 3.32 (Left S-polynomials)
Let $f, g \in S \setminus \{0\}$ with $\text{lt}(f) = x^\alpha$ and $\text{lt}(g) = x^\beta$. Let $\mu \in \mathbb{N}^n$ such that $x^\mu = \text{lcm}(x^\alpha, x^\beta)$. The **left S-polynomial** of f and g is defined by

$$\text{LSP}(f, g) := x^{\mu-\alpha} \star f - \frac{\text{lc}(x^{\mu-\alpha} \star f)}{\text{lc}(x^{\mu-\beta} \star\ g)}\, x^{\mu-\beta} \star g.$$

We note that $\text{lt}(\text{SLP}(f, g)) \prec \text{lt}(f \star g)$.

Remark 3.33

- We note in Definition 3.32 that if $\text{lt}(f)$ is divisible by $\text{lt}(g)$, then $\text{lcm}(x^\alpha, x^\beta) = x^\alpha$ and the left S-polynomial of f and g is

  $$\text{LSP}(f, g) = f - \frac{\text{lc}(f)}{\text{lc}(x^{\mu-\beta} \star g)} x^{\alpha-\beta} \star g.$$

  Moreover, in this case $\text{LSP}(f, g)$ is nothing else but the left reduction of f modulo g.

- If the polynomial f is not monic (i.e. $\text{lc}(f) = 1$), then to perform a left reduction of f modulo a polynomial g we should divide by the leading coefficient of f. Hence, an implementation of an algorithm computing a left normal form by successive left reductions should manage fractions which may be a source of inefficiency. First, because dividing is time consuming compared to other arithmetic operations. Second, because managing fractions may lead to an intermediate exponential growth of the coefficients. For these reasons it is better, in each left reduction step, to get rid of the dominators. With this strategy the obtained left normal form is defined up to a factor $c \in \mathbb{K}$, that is, $\text{LNF}(f \mid G) = cf - \sum_{i=1}^{s} s_i \star g_i$.

Proof: (Algorithm 3.1).

Algorithm 3.1: **Left normal form: LNF**

```
Input  : $S$ a sovable algebra with respect to $\prec$, $f \in S$ and $G \subset S$ a finite subset of $S$;
Output: $r :=$ LNF$(f \mid G)$;
begin
    Initialization: $r \longleftarrow f$;
    while $r \neq 0$ and $G_r = \{g \in G;\ lt(g) \mid lt(r)\} \neq \emptyset$ do
        choose any $g \in G_h$;
        $r \longleftarrow$ LSP$(r, g)$;
    end
    return $r$
end
```

- **Termination**:
  Let $r_0 = f$ and in the i-th step of the while loop we compute $r_i =$ LSP(r_{i-1}, g). Since $\text{lt}(r_i) = \text{lt}(\text{LSP}(r_{i-1}, g)) \prec \text{lt}(r_{i-1})$ (see Definition 3.32), we obtain a set $\{\text{lt}(r_i)\}$ of leading terms of r_i, where $\text{lt}(r_{i+1}) \prec \text{lt}(r_i)$. Since $\prec$ is an admissible term order (see Definition 3.5), this set has a minimum, hence the algorithm terminates.

- **Correctness**:
  Suppose that this minimum is reached at the step s. Let $r = r_s$. Making a backward substitution we obtain the expressinon
  $$r = f - \sum_{i=1}^{s-1} s_i \star g_i,$$
  where s_i are monomials in S and $g_i \in G$. This expression satisfies
  $$\text{lt}(f) = \text{lt}(s_1 \star g_2) \succ \cdots \succ \text{lt}(s_i \star g_i) \succ \cdots \succ \text{lt}(r).$$
  By construction, if $r \neq 0$ then $\text{lt}(r) \notin \text{L}(G)$. Hence, correctness follows.

□

We can easily extend LNF algorithm (Algorithm 3.1) to a reduced LNF algorithm (Algorithm 3.2).

Proof: (Algorithm 3.2)

- **Termination**:
  Since $\text{lt}\big(g - \text{lc}(g)\text{lt}(g)\big) \prec \text{lt}(g)$ and $\prec$ is an admissible term order, the algorithm terminates.

- **Correctness**:
  Follows from the correctness of the LFN algorithm.

□

Algorithm 3.2: Reduced left normal form: RedLNF

```
Input  : $S$ a sovable algebra with respect to $\prec$, $f \in S$ and $G \subset S$ a finite subset of $S$;
Output: $r := \text{RedLNF}(f \mid G)$;
begin
    Initialization: $r \longleftarrow 0, g \longleftarrow f$;
    while $g \neq 0$ do
        $g \longleftarrow \text{LNF}(g \mid G)$;
        $r \longleftarrow r + \text{lc}(g)\text{lt}(g)$;
        $g \longleftarrow g - \text{lc}(g)\text{lt}(g)$;
    end
    return $r$
end
```

3.2.2 Left Buchberger's Algorithm

In 1965, Bruno Buchberger in ([Buc65], [Buc85]) introduced the notion of Gröbner bases for commutative polynomial ideals, and gave an algorithm for their computation, known as Buchberger's algorithm. In this section, we present the left analog of this algorithm in the noncommutative context of solvable algebras.

Definition 3.34 (Left Gröbner basis)
Let S be a solvable algebra with respect to $\prec$, $I \subset S$ a left ideal and $G \subset I$ a finite subset. Then, G is called a **left Gröbner basis** of I if and only if $L(G) = L(I)$ as vector spaces. In other words, for any $f \in I \setminus \{0\}$ there exists $g \in G$ satisfying $\text{lt}(g) \mid \text{lt}(f)$.

Lemma 3.35
Let S be a solvable algebra with respect to $\prec$, $I \subset S$ a left ideal and $G \subset I$ a Left Gröbner Basis of I. Then,

1. for any $f \in S$ we have $f \in I \Leftrightarrow \text{LNF}(f \mid G) = 0$.
2. If $J \subset S$ is a left ideal such that $I \subset J$, then $L(I) = L(J)$ implies $I = J$. In particular, G generates I as a left ideal of S.

Proof:

1. If $\text{LNF}(f \mid G) = 0$ then $f \in I$ (see Lemma 3.30).
   If $\text{LNF}(f \mid G) \neq 0$, then $\text{lt}\big(\text{LNF}(f \mid G)\big) \notin L(G) = L(I)$. Hence, $\text{LNF}(f \mid G) \notin I$ which implies that $f \notin I$.
2. We want to prove that $J \subset I$. Let $f \in J$ and we assume that $r := \text{LNF}(f \mid G) \neq 0$. Furthermore, from Lemma 3.27 it follows that $\text{lt}(r) \notin L(G) = L(I) = L(J)$, which contradicts the fact that $r = f - \sum s_i \star g_i$ is an element of J. Hence, $r = \text{LNF}(f \mid G) = 0$ and $f \in I$ by (1).

□

Remark 3.36

- Let $I \subset S$ be a left ideal and $f \in S$. Decide whether f is an element of I or not is called the **left ideal membership problem (LIMP)**. From the first statement of Lemma 3.35, we note that the LIMP may be solved algorithmically by means of the LNF-algorithm (Algorithms 3.1 and 3.2) if the input is a Left Gröbner Basis of the ideal I.

The following theorem proves the correctness of Buchberger's algorithm for the computation of a Left Gröbner Basis.

Theorem 3.37 (Buchberger's left criterion)
Let S be a solvable algebra with respect to $\prec$, $I \subset S$ a left ideal a S and $G = \{g_1, \dots, g_s\} \subset I$ a finite subset of I. Then the following statements are equivalent

1. G is a Left Gröbner Basis of I.
2. LNF$(f \mid G) = 0$ for all $f \in I$.
3. Each $f \in I$ has a left standard representation with respect to G.
4. LNF$\big($LSP$(g_i, g_j) \mid G\big) = 0$ for $1 \leq i, j \leq s$.

Proof:

- The implication $(1 \Rightarrow 2)$ follows from Lemma 3.35-1.
- The implication $(2 \Rightarrow 3)$ follows from Lemma 3.30-1
- For the implication $(3 \Rightarrow 1)$, we see that if f has a left standard representation with respect to G: $f = \sum\limits_{i=1}^{s} s_i \star g_i$, then lt$(f)$ must occur as the leading term of $s_i \star g_i$ for some i. It means that lt$(g_i) \mid$ lt(f), hence by Definition 3.34 G is a Left Gröbner Basis of I.
- To prove $(3 \Rightarrow 4)$, we note first that $r_{ij} =$ LNF$\big($LSP$(g_i, g_j) \mid G\big) \in I$ which means if $r_{ij} \neq 0$ that lt$(r_{ij}) \in L(G)$ and contradicts the second property of LNF in Lemma 3.3.
- The implication $(4 \Rightarrow 1)$ is the left Buchberger criterion which allows checking and construction of Gröbner bases in a finite number of steps in the case where S is left noetherian (see Theorem 3.22). The proof of this implication uses *syzygies*, for more details the reader may refer to ([Lev05], Theorem 4.8).

□

Proof:

- **Termination**:
By Lemma 3.27-(2), we know that if $r_{ij} \neq 0$ then lt$(r_{ij}) \notin L(G)$. Therefore, ${}_S\langle G\rangle \subset {}_S\langle\{G, r_{ij}\}\rangle$ and we obtain a strictly ascending chain of left ideals in S. Since S is noetherian, this chain stabilizes (see Definition 3.21). This means that, after **finitely** many steps all LSPs reduce to zero and the set P becomes empty.

Algorithm 3.3: **Left Buchberger's algorithm**

```
Input  : $S$ a noetherian sovable algebra with respect to $\prec$, $F \subset S$ a finite subset of $S$;
Output: $G$ a Left Gröbner Basis of the left ideal $I = {}_S\langle F \rangle$;
begin
    Initialization:
    $G \longleftarrow F$;
    $P \longleftarrow \{(g_i, g_j);\ g_i, g_j \in G, g_i \neq g_j\}$;
    while $P \neq \emptyset$ do
        Select $(g_i, g_j) \in P$;
        $P \longleftarrow P \setminus \{(g_i, g_j)\}$;
        $r_{ij} \longleftarrow \text{LNF}\big(\text{LSP}(g_i, g_j) \mid G\big)$;
        if $r_{ij} \neq 0$ then
            $P \longleftarrow P \cup \{(r_{ij}, g_k);\ g_k \in G\}$;
            $G \longleftarrow G \cup \{r_{ij}\}$;
        end
    end
    return $G$;
end
```

- **Correctness**:
  Follows from Theorem 3.37

□

Remark 3.38 (Reduced left Gröbner basis)

- If in the While loop of Algorithm 3.3 we perform the RedLFN algorithm (Algorithm 3.2) and we have as input a reduced set F (see Definition 3.28), then we obtain a reduced Gröbner basis G as output. If F is not reduced, we may apply the RedLFN algorithm afterwards to $(g \mid G \setminus \{g\})$ for all $g \in G$ in order to obtain a reduced Gröbner basis.

Buchberger's algorithm has a high complexity which depends on the number of variables n of the polynomial algebra and the maximal degree d of the polynomials in the input. In the commutative case this complexity is $d^{O(n^2)}$ (see [Laz92], [FGDM93]). This high complexity is essentially due to the cost of reductions which depends on the size of the polynomials and may exponentially increase along the steps of the while loop. Hence, in order to optimize Buchberger's algorithm, it is natural to look for a **selection strategy** that predicts critical pairs before executing reductions. By critical pairs, we refer to those whose LSP reduces to zero and thereby, they do not intervene in the construction of the output or the pairs with a high reduction cost. In the following we will give an overview on some known selection strategies in the commutative case and discuss how far they may be generalized to the noncommutative context of solvable algebras.

Buchberger's Normal Strategy

The normal strategy of Buchberger consists of two criteria.

1. **Product criterion (Pcrit)**: for a pair $(g_i, g_j) \in G \times G$ we have

   $$\text{lcm}(\text{lt}(g_i), \text{lt}(g_j)) = \text{lt}(g_i)\text{lt}(g_j) \Rightarrow \text{LNF}\big(\text{LSP}(g_i, g_j) \mid G\big) = 0.$$

   It follows that in Algorithm 3.3 we omit the reduction of pairs $(g_i, g_j) \in P$ satisfying the condition of the previous statement.

2. **Chain criterion (Ccrit)**: in Algorithm 3.3, we omit the reduction of a LSP of a pair (g_i, g_j), if there exists $g_k \in G$ such that $\text{lt}(g_k)$ divides $\text{lcm}(\text{lt}(g_i), \text{lt}(g_j))$ **or** the pairs (g_i, g_k) and (g_k, g_j) have been already reduced by the algorithm.

Although the **Ccrit** may be applied in the context of solvable algebras, it is not quite the case for the **Pcrit**. We illustrate this fact in the following example.

Example 3.39
Let $\mathbb{K}[x][D_x; 1, D_x]$ be the Weyl algebra endowed with the lexicographical term order $\prec_{\text{lex}}$, where $x \prec_{\text{lex}} D_x$. Furthermore, $f = x$ and $g = D_x$ with $\text{lcm}(\text{lt}(f), \text{lt}(g)) = \text{lt}(f)\text{lt}(g)$. We note that $\text{LSP}(f, g) = D_x f - xg = 1$ does not reduce to zero, which contradicts the statement of the **Pcrit**.

The contradiction of the previous example with the statement of the **Pcrit** is due to the fact that $\text{lt}(f) = x$ and $\text{lt}(g) = D_x$ do not commute. Hence, we may extend the **Pcrit** to the noncommutative context, if we add the condition that $\text{lt}(g_i)$ and $\text{lt}(g_j)$ commute. That is, the variables of $\text{lt}(g_i)$ commute with all the variables of $\text{lt}(g_j)$, which may happen, for instance, in Ore algebras (see 3.1.3).

"Sugar" Strategy

The Buchberger normal strategy works fine if the term order is degree compatible, but it's quite bad for the lexicographical term order. This motivates the "Sugar" selection strategy presented by A. Giovini and T. Mora in [GMN+91]. Before we give an overview on this strategy, we need to introduce some definitions.

Definition 3.40 (The "Sugar")
Let $R = \mathbb{K}[x_1, \ldots, x_n]$ be a polynomial ring. The **"Sugar"** is defined to be the map

$$s : R \longrightarrow \mathbb{N}, \ f \longmapsto s(f) = d,$$

where d is the total degree of f. It is clear that the "Sugar" satisfies the following properties

(i) $s(fg) = s(f) + s(g)$

(ii) $s(f + g) = \max(s(f), s(g))$

Definition 3.41

(i) A polynomial $f \in \mathbb{K}[x_1, \ldots, x_n]$ is called **homogeneous** of degree d, if every term of f with nonzero coefficient has a total degree d.

(ii) A non-homogeneous polynomial $f \in R$ may be homogenized by introducing a new variable y and defining

$$h\big(f(x_1, \ldots, x_n)\big) = y^d f(\frac{x_1}{y}, \ldots, \frac{x_n}{y}),$$

where d is the total degree of f.

The ”Sugar selection strategy” is generally combined with Buchberger's normal strategy, and consists of two steps before each reduction in Algorithm 3.3.

1. Homogenize the set of polynomials G
2. Perform the normal strategy, and after that select the pair $\big(h(g_i), h(g_j)\big)$ such that LSP$(h(g_i), h(g_j))$ has the smallest ”Sugar”.

We note that the ”Sugar” selection strategy may be generalized to the context of solvable algebras, since the homogenization by a commutative variable does not interfere with the commutator relations.

Trace Lifting

The trace lifting strategy, presented by Traverso in [Tra89], has drastically improved Buchberger's algorithm in the case of polynomial rings over transcendental field extension, for instance, $\mathbb{K} = \mathbb{Q}(\alpha)$. This strategy can be applied in the commutative case, as well as in the noncommutative context of solvable algebras. The idea of this strategy is to give temporary a numeric value to α, say α_0, and perform reduction with this value. Obviously, the computation with a numeric α_0 is much faster than the generic computation in α. If the result of the reduction for α_0 is nonzero, then it is likely the case for a generic α thus, the reduction is immediately performed. Otherwise, the reduction is delayed.

3.2.3 Applications

Elimination of Variables

Definition 3.42
Let $S = \mathbb{K}\{x_1, \ldots, x_n; Q, Q'\}$ be a solvable algebra with respect to a $\star$-compatible admissible term order $\prec$. Let $0 \leq m \leq n$ and define

$$\begin{aligned} S_m &= \mathbb{K}\{x_{m+1}, \ldots, x_n; Q_m, Q'_m\}, \text{ where} \\ Q_m &= Q \cap \mathbb{K}[x_{m+1}, \ldots, x_n], \\ Q'_m &= Q' \cap \mathbb{K}[x_{m+1}, \ldots, x_n]. \end{aligned}$$

Then S_m is a solvable subalgebra of S, since by definition and Lemma 3.11 is closed under the $\star$ multiplication inherited from S. The algebra S_m is called the m-th **elimination subalgebra** of S.

Definition 3.43 (Elimination term order)
A term order $\prec$ on $T(x_1,\dots,x_n)$ is called **elimination term order** for $\{x_1,\dots,x_m\}$, if it satisfies $\{x_1,\dots,x_m\} \succ \{x_{m+1},\dots,x_n\}$. Such a term order is characterized by

$$f \in \mathbb{K}[x_{m+1},\dots x_n] \Leftrightarrow \mathrm{lt}(f) \in T(x_{m+1},\dots,x_n)$$

Definition 3.44 (Elimination problem)
Let $S = \mathbb{K}\{x_1,\dots,x_n;Q,Q'\}$ be a solvable algebra with respect to a $\star$-compatible admissible term order $\prec$, $S_m \subset S$ an elimination subalgebra and $I \subset S$ a finitely generated left ideal. The determination of a set of generators of the left ideal $I \cap S_m$ is called **elimination problem**.

The elimination problem can be solved algorithmically by means of the following lemma.

Lemma 3.45
Let $S = \mathbb{K}\{x_1,\dots,x_n;Q,Q'\}$ be a solvable algebra with respect to a $\star$-compatible elimination term order $\prec$ for $\{x_1,\dots,x_m\}$. Let $S_m \subset S$ be an elimination subalgebra of S and $I \subset S$ a left ideal given by a Gröbner basis G. Then, a Gröbner basis of $I \cap S_m$ is $G_m = G \cap S_m$.

In the previous lemma, we note that the elimination term order $\prec$ must be $\star$-compatible, that is, the commutator relations $x_j \star x_i = c_{ij}x_ix_j + d_{ij}$ must satisfy $\mathrm{lt}(d_{ij}) \prec x_ix_j$. Otherwise, no elimination is possible. We illustrate this fact in the following example.

Example 3.46
Let $\mathbb{O} = \mathbb{K}[x,u][D_x;1,D_x]$ be the Ore algebra defined as an extension of the Weyl Algebra $\mathbb{K}[x][D_x;1,D_x]$ by introducing an additional commutator relation

$$D_x \star u = uD_x - u^2.$$

An elimination term order for D_x should satisfy $D_x \prec u$. However such a term order is not $\star$-compatible, since $u^2 \succ uD_x$. Therefore, no elimination of D_x is possible.

Intersection of Ideals

The intersection of two left ideals of a solvable algebra S may be reduced to the intersection of an ideal with a subalgebra (elimination problem: Definition 3.44) by means of the following lemma.

Lemma 3.47
Let $S = \mathbb{K}\{x_1,\dots,x_n;Q,Q'\}$ be a solvable algebra with respect to a $\star$-compatible admissible term order $\prec$. Let I and J given by the left Gröbner bases $G_I = \{f_1,\dots,f_r\}$, $G_J = \{g_1,\dots,g_s\}$, respectively. We consider the left ideal $D := tI + (1-t)J$ in $S_t = S \otimes_{\mathbb{K}} \mathbb{K}[t]$ viewed as solvable algebra, where t commutes with S. If $\prec$ is an elimination term order for t on S_t, then $I \cap J = D \cap S$.

Proof:

- $D \cap S \subseteq I \cap J$
  Let $f \in D \cap S$. Then f can be presented as a sum

$$f = \sum_{i=1}^{r} t\, r_i \star f_i + \sum_{i=1}^{s} (1-t)\, s_i \star g_i, \text{ where } r_i, s_i \in S$$

If in this sum we substitute $t = 0$ we get

$$f = \sum_{i=1}^{s} s_i \star g_i \in J,$$

and if we substitute $t = 1$ we get

$$f = \sum_{i=1}^{r} r_i \star f_i \in I.$$

Hence, $f \in I \cap J$.

- $I \cap J \subseteq D \cap S$
  Let $f \in I \cap J$. Then f can be represented in two ways

  $$f = \sum_{i=1}^{r} r_i \star f_i = \sum_{i=1}^{s} s_i \star g_i.$$

  Since t commutes with S, consider

  $$f = tf + (1-t)f = \sum_{i=1}^{r} t\, r_i \star f_i + \sum_{i=1}^{s} (1-t)\, s_i \star g_i \in D \cap S.$$

  Hence, $f \in D \cap S$ and $I \cap J = D \cap S$.

□

At the end of the last chapter we mentioned the notion of *derivative rules* without explaining how they are defined and for which purposes they are used. A family of functions $f_n(x)$ is called *admissible* (see [Koe95]) if it satisfies a recurrence equation of the form

$$\sum_{k=0}^{m} r_k\, f_{n+k}(x) = 0 \tag{3.3}$$

with polynomial coefficients $r_k \in \mathbb{K}[n, x]$, and a derivative rule of the form

$$f_n'(x) = \frac{\partial}{\partial x} f_n(x) = \sum_{k=0}^{m-1} r_k\, f_{n-k}(x) \quad \text{or} \quad f_n'(x) = \sum_{k=0}^{m-1} r_k\, f_{n+k}(x), \tag{3.4}$$

where the derivative with respect to x is represented by a finite number of lower or higher indexed functions of the family, and where $r_k \in \mathbb{K}(n, x)$ are rational function in n and x. The two types of derivative rules are called *backward* and *forward* derivative rules, respectively. The rather rigid property of admissibility has many interesting consequences, that can be used to generate and verify identities for these functions by linear algebra techniques. In the next Maple sessions, we will show how Gröbner basis elimination techniques can be used to determine such derivative rules. Moreover, we will show that using Gröbner elimination we can get same results as those obtained by skew elimination.

Maple Session 3.1 (Gröbner elimination: Whittaker functions $M_{n,m}(x)$)

We load the packages we need

```
>  with(Ore_algebra):
>  with(Groebner):
```

and define the algebra where the Gröbner elimination will be performed.

```
>  A:=skew_algebra(shift=[Sm,m],diff=[Dx,x], comm=n,polynom=x);
```

$$A := \mathit{Ore\_algebra}$$

The Whittaker functions are annihilated by the following differential and recurrence operators:

```
>  DE:=4*x^2*Dx^2-x^2+4*n*x-4*m^2+1;
```

$$\mathit{DE} := 4\,x^2\,\mathit{Dx}^2 - x^2 + 4\,n\,x - 4\,m^2 + 1$$

```
>  RE:=(1+2*m)*(2*n-3-2*m)*(3+2*m+2*n)*x*Sm^2+16*(1+m)*(2+m)*(3+2*m)
>  *(2*n*x-3-8*m-4*m^2)*Sm+16*(1+m)*(2+m)*(1+2*m)*(3+2*m)^2*x;
```

$$\begin{aligned}\mathit{RE} := {} & (1+2\,m)\,(2\,n-3-2\,m)\,(3+2\,m+2\,n)\,x\,\mathit{Sm}^2\\ & + 16\,(1+m)\,(2+m)\,(3+2\,m)\,(2\,n\,x-3-8\,m-4\,m^2)\,\mathit{Sm}\\ & + 16\,(1+m)\,(2+m)\,(1+2\,m)\,(3+2\,m)^2\,x\end{aligned}$$

We compute a Gröbner basis w. r. t. an elimination term order $\prec_1$ that eliminates as far as possible the variable x and the derivative operator Dx. For this purpose, we choose $\prec_1$ to be the lexicographical term order with $Sm \prec_1 Dx \prec_1 x$.

```
>  T1:=MonomialOrder(A,plex(x,Dx,Sm));
```

$$\mathit{T1} := \mathit{monomial\_order}$$

```
>  GB1:=Basis([DE,RE],T1);
```

$$\begin{aligned}\mathit{GB1} := {} & [-864 + 16\,\mathit{Sm}^2\,m\,n^2 - 832\,\mathit{Sm}\,m\,n - 576\,\mathit{Sm}\,m^2\,n - 128\,\mathit{Sm}\,m^3\,n\\ & - 4752\,m - 10224\,m^2 - 11200\,m^3 - 6656\,m^4 - 2048\,m^5 - 9\,\mathit{Sm}^2\\ & + 4\,\mathit{Sm}^2\,n^2 - 48\,\mathit{Sm}^2\,m - 88\,\mathit{Sm}^2\,m^2 - 64\,\mathit{Sm}^2\,m^3 - 384\,\mathit{Sm}\,n\\ & + 16\,\mathit{Sm}^2\,n^2\,m^2 + 576\,\mathit{Dx}\,\mathit{Sm} - 16\,\mathit{Sm}^2\,m^4 + 2784\,\mathit{Dx}\,\mathit{Sm}\,m\\ & + 4960\,\mathit{Dx}\,\mathit{Sm}\,m^2 + 4160\,\mathit{Dx}\,\mathit{Sm}\,m^3 + 1664\,\mathit{Dx}\,\mathit{Sm}\,m^4\\ & + 256\,\mathit{Dx}\,\mathit{Sm}\,m^5 - 256\,m^6, -4\,x\,\mathit{Sm}^2\,n^2 + 9\,x\,\mathit{Sm}^2 + 30\,x\,\mathit{Sm}^2\,m\\ & + 28\,x\,\mathit{Sm}^2\,m^2 - 8\,x\,\mathit{Sm}^2\,m\,n^2 + 8\,x\,\mathit{Sm}^2\,m^3 - 192\,\mathit{Sm}\,n\,x + 288\,\mathit{Sm}\\ & + 1392\,\mathit{Sm}\,m + 2480\,\mathit{Sm}\,m^2 - 416\,\mathit{Sm}\,m\,n\,x + 2080\,\mathit{Sm}\,m^3\\ & - 288\,\mathit{Sm}\,m^2\,n\,x + 832\,\mathit{Sm}\,m^4 - 64\,\mathit{Sm}\,m^3\,n\,x + 128\,\mathit{Sm}\,m^5 - 288\,x\\ & - 1392\,x\,m - 2480\,x\,m^2 - 2080\,x\,m^3 - 832\,x\,m^4 - 128\,x\,m^5,\\ & -32\,m^4 - 80\,m^3 + 32\,x\,\mathit{Dx}\,m^3 - 4\,x\,\mathit{Sm}\,m^2 - 72\,m^2 + 64\,x\,\mathit{Dx}\,m^2\\ & + 16\,m^2\,n\,x - 28\,m + 40\,x\,\mathit{Dx}\,m - 4\,x\,\mathit{Sm}\,m + 24\,m\,n\,x - 4\\ & + 8\,x\,\mathit{Dx} - x\,\mathit{Sm} + 4\,\mathit{Sm}\,n^2\,x + 8\,n\,x]\end{aligned}$$

After observing the computed Gröbner basis, we can see that a derivative rule satisfied by the Whittaker function is:

```
>  DR1:=collect(GB1[1],[Dx,Sm],factor);
```

$$\begin{aligned}DR1 := {} & 32\,(1+2\,m)\,(2+m)\,(1+m)\,(3+2\,m)^2\,Sm\,Dx\\ & -(1+2\,m)^2\,(3+2\,m+2\,n)\,(2\,m+3-2\,n)\,Sm^2\\ & -64\,n\,(3+2\,m)\,(2+m)\,(1+m)\,Sm\\ & -16\,(1+2\,m)\,(2+m)\,(1+m)\,(3+2\,m)^3\end{aligned}$$

The derivative rule $DR1$ has an advantage that it is free of the variable x but it does not really coincide with the one mentioned in the paper [Koe95] (Section 8, Theorem 7).
Now if we choose a term order $\prec_2$ which eliminates in the first place the operator Dx, we get exactly the derivative rule of [Koe95].

```
>   T2:=MonomialOrder(A,plex(Dx,Sm,x));
```

$$T2 := monomial\_order$$

```
>   GB2:=Basis([DE,RE],T2);
```

$$\begin{aligned}GB2 := [& -4\,x\,Sm^2\,n^2+9\,x\,Sm^2+30\,x\,Sm^2\,m+28\,x\,Sm^2\,m^2-8\,x\,Sm^2\,m\,n^2\\ & +8\,x\,Sm^2\,m^3-192\,Sm\,n\,x+288\,Sm+1392\,Sm\,m+2480\,Sm\,m^2\\ & -416\,Sm\,m\,n\,x+2080\,Sm\,m^3-288\,Sm\,m^2\,n\,x+832\,Sm\,m^4\\ & -64\,Sm\,m^3\,n\,x+128\,Sm\,m^5-288\,x-1392\,x\,m-2480\,x\,m^2\\ & -2080\,x\,m^3-832\,x\,m^4-128\,x\,m^5,-32\,m^4-80\,m^3+32\,x\,Dx\,m^3\\ & -4\,x\,Sm\,m^2-72\,m^2+64\,x\,Dx\,m^2+16\,m^2\,n\,x-28\,m+40\,x\,Dx\,m\\ & -4\,x\,Sm\,m+24\,m\,n\,x-4+8\,x\,Dx-x\,Sm+4\,Sm\,n^2\,x+8\,n\,x,\\ & -864+16\,Sm^2\,m\,n^2-832\,Sm\,m\,n-576\,Sm\,m^2\,n-128\,Sm\,m^3\,n\\ & -4752\,m-10224\,m^2-11200\,m^3-6656\,m^4-2048\,m^5-9\,Sm^2\\ & +4\,Sm^2\,n^2-48\,Sm^2\,m-88\,Sm^2\,m^2-64\,Sm^2\,m^3-384\,Sm\,n\\ & +16\,Sm^2\,n^2\,m^2+576\,Dx\,Sm-16\,Sm^2\,m^4+2784\,Dx\,Sm\,m\\ & +4960\,Dx\,Sm\,m^2+4160\,Dx\,Sm\,m^3+1664\,Dx\,Sm\,m^4\\ & +256\,Dx\,Sm\,m^5-256\,m^6]\end{aligned}$$

```
>   DR2:=collect(GB2[2],[Dx,Sm],factor);
```

$$\begin{aligned}DR2 := {} & 8\,x\,(1+m)\,(1+2\,m)^2\,Dx-x\,(2\,m+1+2\,n)\,(2\,m+1-2\,n)\,Sm\\ & -4\,(1+2\,m)\,(1+m)\,(4\,m^2+4\,m+1-2\,n\,x)\end{aligned}$$

Now we will show that Gröbner basis elimination techniques can be used to get same results as by skew elimination (see Maple Sessions 2.3 and 2.3).
Proceeding by Gröbner elimination we deduce the differential operator DE from the operators $DR2$ and RE. We choose the lexicographical term order with $x \prec_3 Dx \prec_3 Sm$ which eliminates in the first place the operator Sm.

```
>   T3:=MonomialOrder(A,plex(Sm,Dx,x));
```

$$T3 := monomial\_order$$

We compute a Gröbner basis w. r. t. $\prec_3$

```
>   GB3:=Basis([DR2,RE],T3):
```

to get the differential operator DE as follows:

```
>   remove(has,GB3,Sm);
```

$$[4\,x^2\,Dx^2 - x^2 + 4\,n\,x - 4\,m^2 + 1]$$

We can similarly deduce RE from $DR2$ and DE.

```
>   T4:= MonomialOrder(A,plex(Dx,Sm,x));
```

$$T4 := monomial\_order$$

```
>   GB4:=Basis([DR2,DE],T4):
>   collect(remove(has,GB4,Dx),Sm,factor);
```

$$[x\,(1+2\,m)\,(3+2\,m+2\,n)\,(2\,m+3-2\,n)\,Sm^2$$
$$+\,16\,(3+2\,m)\,(2+m)\,(1+m)\,(4\,m^2+8\,m+3-2\,n\,x)\,Sm$$
$$-\,16\,(1+m)\,(2+m)\,(1+2\,m)\,(3+2\,m)^2\,x]$$

Remark 3.48
The derivative rule $DR1$ of the previous Maple session can be expressed in the following form:

$$DR1 : M'_{n,m+1}(x) = R_2(n,m)M_{n,m+2}(x) + R_1(n,m)M_{n,m+1}(x) + R_0(n,m)M_{n,m}(x), \quad (3.5)$$

where the coefficients R_i are elements of $\mathbb{Q}(n,m)$. $DR1$ expresses forward **shifted** derivatives of the Whittaker functions as a linear combination of it forward shifts.
If we shift back $DR1$ i.e. substitute m by $m-1$ we get

$$M'_{n,m}(x) = R_2(n,m-1)M_{n,m+1}(x) + R_1(n,m-1)M_{n,m}(x) + R_0(n,m-1)M_{n,m-1}(x) \quad (3.6)$$

which is a derivative rule free of x that expresses the derivatives of the Whittaker functions as a linear combination of its backward and forward shifts.
We note that the set $\{M_{n,m+1}(x), M_{n,m}(x), M_{n,m-1}(x)\}$ is linearly dependent over $\mathbb{Q}(n,m,x)$. In fact, if we consider the recurrence equation satisfied by $M_{n,m}(x)$ which has the following form:

$$RE : r_2(n,m,x)M_{n,m+2}(x) + r_1(n,m,x)M_{n,m+1}(x) + r_0(n,m,x)M_{n,m}(x) = 0, \quad (3.7)$$

then we substitute m by $m-1$ we get

$$r_2(n,m-1,x)M_{n,m+1}(x) + r_1(n,m-1,x)M_{n,m}(x) + r_0(n,m-1,x)M_{n,m-1}(x) = 0. \quad (3.8)$$

Hence, we can express $M_{n,m-1}(x)$ as a linear combination of $M_{n,m+1}(x)$ and $M_{n,m}(x)$ as follows

$$\begin{aligned} M_{n,m-1}(x) &= -\frac{r_2(n,m-1,x)}{r_0(n,m-1,x)}M_{n,m+1}(x) - \frac{r_1(n,m-1,x)}{r_0(n,m-1,x)}M_{n,m}(x) \\ &= A(n,m,x)M_{n,m+1}(x) + B(n,m,x)M_{n,m}(x), \quad (3.9) \end{aligned}$$

where $A(n,m,x)$ and $B(n,m,x)$ are elements of $\mathbb{Q}(n,m,x)$. Substituting (3.9) in (3.6) leads to

$$\begin{aligned} M'_{n,m}(x) &= \big(R_2(n,m-1) + R_0(n,m-1)A(n,m,x)\big)M_{n,m+1}(x) \\ &+ \big(R_1(n,m-1) + R_0(n,m-1)B(n,m,x)\big)M_{n,m}, \quad (3.10) \end{aligned}$$

which is the derivative rule $DR2$ that has coefficients depending on x.

Maple Session 3.2 (Gröbner elimination: Jacobi orthogonal polynomials $P_n^{(a,b)}(x)$)

We load the packages we need

```
> with(Ore_algebra):
> with(Groebner):
```

and define the Algebra where the Gröbner elimination will be performed.

```
> A:=skew_algebra(comm=[a,b],shift=[Sn,n],diff=[Dx,x],
> polynom={x,n});
```

$$A := Ore\_algebra$$

The Jacobi orthogonal polynomials are annihilated by the following operator system:

```
> DE:= (1-x^2)*Dx^2+(b-a-(a+b+2)*x)*Dx+n*(n+a+b+1);
```

$$DE := (1 - x^2)\,Dx^2 + (b - a - (a + b + 2)\,x)\,Dx + n\,(n + a + b + 1)$$

```
> RE:=2*(n+2)*(n+a+2+b)*(a+2+b+2*n)*Sn^2-(2*n+3+a+b)*(x*a^2+a^2+
> 4*x*a*n+2*x*a*b+6*x*a+4*x*n*b+4*x*n^2+6*b*x-b^2+8*x+b^2*x+12*x*n)
> *Sn+2*(2*n+a+4+b)*(n+b+1)*(n+a+1);
```

$$\begin{aligned}RE := {}& 2\,(n+2)\,(n+a+2+b)\,(a+2+b+2\,n)\,Sn^2 - (2\,n+3+a+b)(\\ & x\,a^2 + a^2 + 4\,x\,a\,n + 2\,x\,a\,b + 6\,x\,a + 4\,x\,n\,b + 4\,x\,n^2 + 6\,b\,x - b^2\\ & + 8\,x + b^2\,x + 12\,x\,n)Sn\\ & + 2\,(2\,n+a+4+b)\,(n+b+1)\,(n+a+1)\end{aligned}$$

For the Gröbner basis computation we choose the degree lexicographical term order $\prec$.

```
> T1:= MonomialOrder(A,lexdeg([Dx,Sn],[x,n]));
```

$$T1 := monomial\_order$$

We compute a Gröbner basis w. r. t. $\prec$.

```
> GB:= Basis([DE,RE],T1):
```

The first generator of the Gröbner basis GB is the derivative rule we are looking for

```
> DR:=collect(GB[1],[Dx,Sn],factor);
```

$$\begin{aligned}DR := {}& (x-1)\,(x+1)\,(n+b+1)\,(n+a+1)\,(2\,n+a+4+b)\\ & (a+2+b+2\,n)\,Dx-\\ & 2\,(n+1)\,(n+b+1)\,(n+a+1)\,(2\,n+a+4+b)\,(n+a+b+1)\,Sn\\ & + (n+b+1)\,(n+a+1)\,(2\,n+a+4+b)\,(n+a+b+1)\\ & (x\,a + a + b\,x - b + 2\,x + 2\,x\,n)\end{aligned}$$

up to a multiplicative factor

```
> ct:= collect(content(GB[1],[Sn,Dx]),[Sn,Dx],factor);
```

$$ct := (2\,n+a+4+b)\,(n+b+1)\,(n+a+1).$$

Dividing by the common factor ct we get:

```
> DR:=collect(DR/ct,[Dx,Sn],factor);
```

$$\begin{aligned}DR := {}& (x-1)\,(x+1)\,(a+2+b+2\,n)\,Dx - 2\,(n+1)\,(n+a+b+1)\,Sn\\ & + (n+a+b+1)\,(x\,a + a + b\,x - b + 2\,x + 2\,x\,n).\end{aligned}$$

Chapter 4

∂-Finite and Holonomic Functions

The solutions of linear recurrence (resp. differential) equations with polynomial or rational coefficients are particularly interesting since they can be uniquely determined by a finite amount of information, namely, the coefficients of the equations and a finite number of initial conditions. These functions were a focus subject of Doron Zeilberger in [Zei90b] , where he generalized the notion of D-finite functions in one variable introduced by Richard P. Stanley in [Sta80] to the case of multivariate functions and sequences as well. A function $f(x_1,\dots,x_n)$ (resp. a sequence $\mathcal{U}(k_1,\dots k_n)$) is $\mathcal{D}$-finite if the vector space generated by its derivatives (resp. its shifts) is of finite dimension over $\mathbb{C}(x_1,\dots,x_n)$ (resp. $\mathbb{C}(k_1,\dots,k_n)$). The notion of $\mathcal{D}$-finiteness may be naturally translated in terms of ideals to the more general setting of Ore algebras where it is called ∂-finiteness. In the first part of this chapter, we will first define and characterize *∂-finite functions* and then we will specially focus on their closure properties and how they can be algorithmically computed. Unfortunately, the ∂-finite property is not sufficient to guarantee the termination of the algorithms of summation and integration of the next chapter. For this reason, we need to introduce another class of functions which are called *holonomic functions* and they will be presented in the second part. At the end, we see that we are rather interested in functions lying in the intersection of the two classes because we need certain nice properties of each.

4.1 ∂-Finite Functions

4.1.1 Definition and Characterization

Definition 4.1 (∂-Finiteness)
Let $\mathbb{O} = \mathbb{K}[\boldsymbol{\partial};\boldsymbol{\sigma},\boldsymbol{\delta}] := \mathbb{K}[\partial_1;\sigma_1,\delta_1]\cdots[\partial_r;\sigma_r,\delta_r]$ be an Ore algebra over a field $\mathbb{K}$.

(i) A left ideal $I \subseteq \mathbb{O}$ is called ∂-**finite** with respect to $\mathbb{O}$ if $\dim_{\mathbb{K}}(\mathbb{O}/I) < \infty$, that is, the $\mathbb{K}$-vector space $\mathbb{O}/I$ is of finite dimension.

(ii) A function f is called ∂-**finite** if an annihilating left ideal $\operatorname{Ann}(f)$ of f is ∂-finite.

Remark 4.2

- The "∂" in the definition is just a symbol and does not refer to any of the ∂_i's that effectively generate the Ore algebra $\mathbb{O}$.
- We know that $\mathbb{O}/\mathrm{Ann}(f)$ is isomorphic to the $\mathbb{O}$-module $\mathbb{O}f$ (see Remark 2.15) generated by the "derivatives" $\boldsymbol{\partial}^\alpha f = \partial_1^{\alpha_1} \cdots \partial_r^{\alpha_r} f$, thus f is ∂-finite if its "derivatives" generate a finite dimensional $\mathbb{K}$-vector space.

Example 4.3
Consider the function $f(x,y) = \exp(\frac{x+y}{x-y})$ in two continuous variables and denote by D_x and D_y the partial derivatives with respect to x and y, respectively. It is easy to show that the successive derivatives $D_x^\alpha D_y^\beta$ are of the form

$$r(x,y)\exp(\frac{x+y}{x-y}),$$

where $r(x,y)$ is a rational function in $\mathbb{Q}(x,y)$. This means that the derivatives of f generate a one-dimensional $\mathbb{Q}(x,y)$-vector space. Hence, f is a ∂-finite function with respect to the Ore algebra $\mathbb{O} = \mathbb{Q}(x,y)[D_x; 1, D_x][D_y; 1, D_y]$.

Definition 4.4 (Rectangular systems)
A finite set of polynomials $\{P_1, \dots, P_r\}$ in the Ore algebra $\mathbb{O} = \mathbb{K}[\partial_1; \sigma_1, \delta_1] \cdots [\partial_r; \sigma_r, \delta_r]$ is called a **rectangular system**, if for each $1 \le i \le r$ the polynomial P_i is different from zero and depends only on the operator ∂_i and contains none of the others.

Proposition 4.5
A left ideal $I \subseteq \mathbb{O} = \mathbb{K}[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$ is ∂-finite if and only if it contains a rectangular system.

Proof:

- ($\Rightarrow$): let $I \subseteq \mathbb{O}$ be a ∂-finite left ideal of dimension n. For each $1 \le i \le r$ we consider the sequence $\{1, \partial_i, \partial_i^2, \dots\}$ which generates a finite $\mathbb{K}$-vector subspace in $\mathbb{O}/I$ of dimension $l < n$. Hence, the set $\{1, \partial_i, \partial_i^2, \dots, \partial_i^{l+1}\}$ is $\mathbb{K}$-linear dependent, that is, there exists a polynomial $P_i(\partial_i) = \sum\limits_{j=1}^{l+1} a_j \partial_i^j$ with coefficients in $\mathbb{K}$ which reduces to zero modulo I and follows $P_i \in I$.
- ($\Leftarrow$): let $I \subseteq \mathbb{O}$ be a left ideal containing a rectangular system $\{P_i(\partial_i);\ 1 \le i \le r\}$. If we denote by d_i the degree of P_i in ∂_i, then $\mathbb{O}/I$ is generated as a $\mathbb{K}$-vector space by the set $\{\partial_1^{\alpha_1} \cdots \partial_r^{\alpha_r};\ 0 \le \alpha_i \le d_i\}$. Hence, $\dim_{\mathbb{K}}(\mathbb{O}/I) \le \prod\limits_{i=1}^{r} d_i < \infty$.

□

We note that a rectangular system is not always a Gröbner basis. We illustrate this fact by the following counter-example from [Chy98].

Example 4.6
Consider the Ore algebra $\mathbb{O} = \mathbb{Q}(x,y)[D_x; 1, D_x][D_y; 1, D_y]$ given by the commutator relations

$$D_x f(x,y) = f(x,y) D_x + \frac{\partial f}{\partial x}(x,y), \qquad D_y f(x,y) = f(x,y) D_x + \frac{\partial f}{\partial y}(x,y).$$

and the rectangular system

$$P = D_x + a(x,y), \qquad Q = D_y + b(x,y),$$

where a and b are two rational functions in $\mathbb{Q}(x,y)$. We denote the left S-polynomial of the pair (P,Q) by

$$\begin{aligned} R &= D_yP - D_xQ \\ &= D_yD_x + D_ya(x,y) - D_xD_y - D_xb(x,y) \\ &= a(x,y)D_y - b(x,y)D_x + \frac{\partial a}{\partial y}(x,y) - \frac{\partial b}{\partial x}(x,y). \end{aligned}$$

Let $\prec$ be an admissible term order such that $D_x \prec D_y$, then the left reduction of R with respect to $G = \{P,Q\}$ gives

$$\begin{aligned} R \longrightarrow_g r_1 &= -b(x,y)D_x + \frac{\partial a}{\partial y}(x,y) - \frac{\partial b}{\partial x}(x,y) - a(x,y)b(x,y), \\ r_1 \longrightarrow_f r &= \mathrm{LNF}(R \mid G) = \frac{\partial a}{\partial y}(x,y) - \frac{\partial b}{\partial x}(x,y). \end{aligned}$$

If $\frac{\partial a}{\partial y}(x,y) - \frac{\partial b}{\partial x}(x,y) \neq 0$, it follows that $\mathrm{LNF}(R \mid G) \neq 0$ and by Theorem 3.37-4, the set G is not a Gröbner. We note that an admissible term order such that $D_y \prec D_x$ yields the same left normal form.

4.1.2 Closure Properties

Given two ∂-functions f and g, in this subsection we will show that $f+g$, fg and Pf (P Ore operator) are (under some conditions for the product) also ∂-finite. Moreover, we will present algorithms to compute a Gröbner basis for this annihilating ideal resulting from the above operations. These algorithms rely on the principles of the **FGLM** algorithm which was initially designed for zero dimensional ideals. The extension of this algorithm to the Ore algebra context in order to algorithmically execute the closure properties was done by F. Chyzak (see [Chy98]).

Proposition 4.7 (Closure under Sum)
If f and g are ∂-finite functions with respect to some Ore algebra $\mathbb{O} = \mathbb{K}[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$, then $f+g$ is ∂-finite with respect to $\mathbb{O}$ as well.

Proof: Let P be an operator in $\mathbb{O}$, then

$$P{\cdot}(f+g) = P{\cdot}f + P{\cdot}g$$

which yields

$$\mathbb{O}{\cdot}(f+g) \simeq \mathbb{O}{\cdot}f \oplus \mathbb{O}{\cdot}g$$

from which it follows

$$\mathbb{O}/\mathrm{Ann}_{\mathbb{O}}(f+g) \simeq \mathbb{O}/\mathrm{Ann}_{\mathbb{O}}(f) \oplus \mathbb{O}/\mathrm{Ann}_{\mathbb{O}}(g).$$

Finally, from the ∂-finiteness of f and g it follows that $\mathrm{Ann}_{\mathbb{O}}(f+g)$ is a finite dimensional $\mathbb{K}$-vector space, since it is the direct sum of two finite dimensional $\mathbb{K}$-vector spaces. Hence, by Definition 4.1 the sum $f+g$ is ∂-finite. □

In the previous proposition, we have shown that summation preserves the property of ∂-finiteness. Therefore, it is natural to be interested in the computation of the annihilating ideal $\mathrm{Ann}_{\mathbb{O}}(f+g)$ of the sum of two ∂-finite functions f and g. In what follows, we present two different types of algorithms which compute a generating set not necessarily of $\mathrm{Ann}_{\mathbb{O}}(f+g)$ but of a subideal of it. In this context, we will consider two cases:

- **case 1**: The functions f and g are given by rectangular systems of $\mathrm{Ann}_{\mathbb{O}}(f)$ and $\mathrm{Ann}_{\mathbb{O}}(g)$.
- **case 2**: The functions f and g are given by Gröbner bases of $\mathrm{Ann}_{\mathbb{O}}(f)$ and $\mathrm{Ann}_{\mathbb{O}}(g)$.

In the first case, we will apply Gaussian elimination to get a rectangular system of $\mathrm{Ann}_{\mathbb{O}}(f+g)$. In the second case, we will apply Gröbner bases techniques in two different ways to compute a Gröbner basis of a subideal of $\mathrm{Ann}_{\mathbb{O}}(f+g)$.

First case

Let $\mathbb{O}=\mathbb{K}[\boldsymbol{\partial};\boldsymbol{\sigma},\boldsymbol{\delta}]$ be an Ore algebra and let f and g be two ∂-finite functions with respect to $\mathbb{O}$ given by rectangular systems P and Q of $\mathrm{Ann}_{\mathbb{O}}(f)$ and $\mathrm{Ann}_{\mathbb{O}}(g)$. For **each variable** ∂ of $\mathbb{O}$ let $P(\partial)$ and $Q(\partial)$ be two polynomials of degrees m and n, respectively. Since $P(\partial)\bullet f\equiv 0$ and $Q(\partial)\bullet g\equiv 0$, the sets $\{f,\partial\bullet f,\ldots,\partial^m\bullet f\}$ and $\{g,\partial\bullet g,\ldots,\partial^n\bullet g\}$ are $\mathbb{K}$-linearly dependent and it follows

$$\partial^l\bullet f=\sum_{j=0}^{m-1}p_j^l\,(\partial^j\bullet f)\quad(l\geq m)\qquad\text{and}\qquad\partial^l\bullet g=\sum_{j=0}^{n-1}q_j^l\,(\partial^j\bullet g)\quad(l\geq n),\tag{4.1}$$

where p_j^l and q_j^l are elements of $\mathbb{K}$.
Let x_j, y_j and t_j denote $\partial^j\bullet f$, $\partial^j\bullet g$ and $\partial^j\bullet(f+g)$, respectively. From the linearity of ∂ the linear system of equations

$$\begin{aligned}t_0&=x_0+y_0\\t_1&=x_1+y_1\\&\vdots\\t_l&=x_l+y_l=\sum_{j=0}^{m-1}p_j^lx_j+\sum_{j=0}^{n-1}q_j^ly_j\quad(\max(m,n)\leq l\leq m+n).\end{aligned}$$

follows. We begin with $l=\max(m,n)$ and solve the linear system of equations with respect to the t_j's by Gaussian elimination of the x_j's and y_j's in order to get a vanishing $\mathbb{K}$-linear combination of the form

$$r_0t_0+r_1t_1+\cdots+r_lt_l=0,\quad\text{where}\quad r_j\in\mathbb{K}.\tag{4.2}$$

If this fails, we increase l by one and repeat the elimination which must terminate successfully at most for $l = m + n$. According to the previous notations, (4.2) is rewritten as follows

$$R(\partial)\cdot(f+g) = (r_0 + r_1\partial + \cdots + r_l\partial^l)\cdot(f+g) \equiv 0. \tag{4.3}$$

Hence, $R(\partial)$ is an element of a rectangular system of $\mathrm{Ann}_{\mathbb{O}}(f+g)$.
We note that by this method the returned ideal does not take into consideration possible mixed relations between the "derivatives". These mixed relations describe $\mathbb{K}$-linear dependencies between elements of a $\mathbb{K}$-basis of $\mathrm{Ann}_{\mathbb{O}}(f+g)$ which allow a better estimation of its dimension as a $\mathbb{K}$-vector space. The procedures presented in the second case produce such mixed relations.

Second case

- **Procedure 1**: computes a Gröbner basis of a subideal of $\mathrm{Ann}_{\mathbb{O}}(f+g)$, relying on the observation that $\mathrm{Ann}_{\mathbb{O}}(f) \cap \mathrm{Ann}_{\mathbb{O}}(g) \subseteq \mathrm{Ann}_{\mathbb{O}}(f+g)$ and using the method presented in Section 3.2.3 which computes a Gröbner basis of intersection of ideals.

- **Procedure 2**: follows a similar principle as the **FGLM** algorithm [FGDM93] designed for zero-dimensional ideals in commutative polynomial rings. This procedure will be applied to perform ∂-finite closure properties not only for summation but also for other operations like product and action of Ore operators. In the following subsection, we will shortly present the FGLM algorithm and the general idea behind it. Afterwards, we will show how it may be extended and used to compute the ∂-finite annihilating ideal corresponding to each of these algebraic operations.

FGLM Algorithm

It is known that Gröbner basis computation depends on the choice of the term order. From a complexity point of view the degree-lexicographical ordering is the best one, whereas the pure lexicographical ordering leads to computations which are much longer than degree orderings (see [CGH89]). However, from a practical point of view the lexicographical ordering is better suited for computing the solution of systems of polynomial equations (see [Laz92]). Between these two extreme cases, there are many other orderings, for instance the elimination block-orderings. The FGLM algorithm is designed for zero-dimensional ideals $I \subset R = \mathbb{K}[x_1, \dots, x_r]$, which are ideals such that R/I is finite-dimensional as $\mathbb{K}$-vector space. This algorithm transforms a Gröbner basis G_1 w. r. t. a term order $\prec_1$ into a Gröbner basis G_2 w. r. t. a term order $\prec_2$. Before we present the algorithm, we first characterize the elements of a $\mathbb{K}$-basis of R/I in the following proposition.

Proposition 4.8
Let R be the polynomial ring $\mathbb{K}[x_1, \dots, x_r]$, T the set of terms of R and I an ideal of R. We denote

by G a reduced Gröbner basis of I w. r. t. an admissible term order $\prec$. A $\mathbb{K}$-basis of R/I is the set

$$\begin{aligned} B(I) &= \{t \in T;\ t \text{ is not divisible by any term of } \mathrm{lt}(I)\} \\ &= \{t \in T;\ t \text{ is not divisible by any term of } \mathrm{lt}(G)\} \\ &= \{t \in T;\ \mathrm{NF}(t \mid G) \neq 0\} \\ &= B(G), \end{aligned}$$

where NF$(t \mid G)$ denotes the reduced normal form of t w. r. t. G.

Proof: See [BKW93, Proposition 6.52]. □

The FGLM algorithm goes systematically through the terms of T, which are elements of $B(G_2)$

Algorithm 4.1: **FGLM**

```
Input  : $G_1 \subset \mathbb{K}[x_1, \dots, x_r]$ a Gröbner basis w.r.t. $\prec_1$;
Output: $G_2$ a Gröbner basis of the ideal $\langle G_1 \rangle$ w.r.t. $\prec_2$;
begin
    Initialization: $T := \{1\}$; $G_2 := \{\}$; $j := 1$;
    while $T \neq \{\}$ do
        $T \leftarrow T \setminus \{t \in T$ ; it exists $g \in G_2$ such that $t$ divides lt$(g)\}$;
        $t_j :=$ min$_{\prec_2} T$;
        $T \leftarrow T \setminus \{t_j\}$;
        define $NF_i$ to be a reduced normal form of $t_i$ w.r.t. $G_1$ ;
        if $\{NF_i;\ 1 \leq i \leq j\}$ are linearly dependent then
            $G_2 \leftarrow G_2 \cup \{c_1 t_1 + c_2 t_2 + \cdots + c_j t_j\}$;
            with $c_i \in \mathbb{K}$ (not all zero) such that $c_1 NF_1 + c_2 NF_2 + \cdots + c_j NF_j = 0$.
        else
            $T \leftarrow T \cup \{x_i t_j;\ 1 \leq i \leq r\}$;
            $j \leftarrow j + 1$;
        end
    end
    return $G_2$;
end
```

and minimal with respect to $\prec_2$ starting by the set $T = \{1\}$. In each step it checks if the considered term $t \in T$ is the leading term of an element of G_2, if it is not the case it enlarges the set T by terms which are multiples of t and repeats the loop until F becomes empty. The j'th while loop of the algorithm is described as follows: while $T_{j-1} \neq \emptyset$ we define the subset T_j of $B(G_2)$ to be

$$T_j := T_{j-1} \setminus \{t \in T_{j-1}\ ;\ \exists g \in G_2 \text{ such that } t \mid \mathrm{lt}(g)\}.$$

We choose $t_j \in T_j$ minimal w.r.t. $\prec_2$ and we compute its normal form NF$_j$ w. r. t. G_1. Then, by Gaussian elimination we check the linear dependence between NF$_j$ and all normal forms NF$_i$ for $1 \leq i \leq j - 1$ from the previous steps. Two cases are possible

- The normal forms are $\mathbb{K}$-linearly dependent, that is, there exist $c_1, \ldots, c_j \in \mathbb{K}$ such that
$$c_1\mathrm{NF}_1 + \cdots + c_j\mathrm{NF}_j = 0.$$
This means that the leading term of the polynomial $g_j = c_1t_1 + \cdots + c_jt_j$ is not in $B(I)$ and hence, in $\mathrm{lt}(G_2)$ (see Proposition 4.8). Furthermore, since the terms t_i $(1 \leq i \leq j$) are chosen such that they are not divisible by any leading term of the partially constructed G_2, it follows that g_j is a new element of G_2.

- The normal forms are $\mathbb{K}$-linearly independent, in this case we redefine T_j as
$$T_j := (T_j \setminus \{t_j\}) \cup \{x_it_j,\ 1 \leq i \leq r\}$$
and we execute a further step. The algorithm terminates when G_2 is complete and this must happen in a finite number of steps since R/I is finite dimensional as a $\mathbb{K}$-vector space.

After presenting the FGLM Algorithm, our aim now is to extend it in such a way to compute a Gröbner basis of an annihilating ideal of a finite sum of ∂-finite functions. Let $f_1, \ldots, f_s$ be functions given by annihilating ideals $I_1, \ldots, I_s \subseteq \mathbb{O} = \mathbb{K}[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$. Let $G_1, \ldots, G_s$ be Gröbner bases of $I_1, \ldots, I_s$ with respect to admissible term orders $\prec_1, \ldots, \prec_s$, respectively. The extended form for ∂-finite summations of the FGLM algorithm computes in fact a Gröbner basis with respct to an admissible term order $\prec$ of the ideal $\mathrm{Ann}(f_1 \oplus f_2 \oplus \cdots \oplus f_s) \subseteq \mathrm{Ann}(f_1 + f_2 + \cdots + f_s)$. This extension is based on a appropriate definition of a normal form $NF(t)$ of a term $t = \boldsymbol{\partial}^\alpha \in \mathbb{O}$ as follows:
$$NF(t) := \sum_{i=1}^{s} \mathrm{RedLNF}(t \mid G_i) \ \in \mathbb{O}/I_1 \oplus \cdots \oplus \mathbb{O}/I_s. \tag{4.4}$$
Then, we execute the steps of the while loop as in the original FGLM algorithm. We note that, by definition, $NF(t)$ is still an element of a finite dimensional $\mathbb{K}$-vector space which ensures the termination of the computation of G.

Proposition 4.9 (Closure under Product)
Let $\mathbb{O} = \mathbb{K}[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$ be an Ore algebra and f and g two ∂-finite functions with respect to $\mathbb{O}$ and given by annihilating ideals I and J. Moreover, we suppose that for each $1 \leq i \leq r$, there exist two polynomials $A_i(u)$ and $B_i(u)$ in $\mathbb{K}[u]$ such that
$$\sigma_i = A_i(\partial_i) \qquad \text{and} \qquad \delta_i = B_i(\partial_i), \tag{4.5}$$
then the product fg is ∂-finite.

Proof: We note that a natural framework to deal with products of the form $(P{\cdot}f)(Q{\cdot}g)$, where P and Q are two operators in $\mathbb{O}$, is the tensor product over $\mathbb{K}$ defined by
$$\mathbb{T} := \mathbb{O}{\cdot}f \otimes \mathbb{O}{\cdot}g \simeq \mathbb{O}/I \otimes \mathbb{O}/J,$$
where $\mathbb{O}, \mathbb{O}{\cdot}f$ and $\mathbb{O}{\cdot}g$ are considered as $\mathbb{K}$-vector spaces. Furthermore, it is required that $\mathbb{T}$ is closed under the action of the operators ∂_i, that is,
$$\partial_i{\cdot}(P{\cdot}f)(Q{\cdot}g) \in \mathbb{T}$$

Algorithm 4.2: **Extended FGLM for ∂-finiteness under sum**

```
Input  : - A Gröbner basis $G_i$ w. r. t. $\prec_i$ for an annihilating ideal of the $\partial$-finite
           function $f_i$, where $1 \leq i \leq s$ ;
         - An admissible term order $\prec$ on $\mathbb{O}$;
Output: A Gröbner basis $G$ of Ann$(f_1 \oplus f_2 \oplus \cdots \oplus f_s)$ w. r. t. $\prec$;
begin
    1. For each term $t = \partial_1^{\alpha_1} \cdots \partial_r^{\alpha_r} \in \mathbb{O}$ define
       $NF(t) := \sum_{i=1}^{s} \text{RedLNF}(t \mid G_i)$;
    2. Execute FGLM Algorithm;
    return $G$;
end
```

for all $1 \leq i \leq r$. In this way $\mathbb{T}$ is, besides its $\mathbb{K}$-vector space structure, endowed by an $\mathbb{O}$-module structure which is ensured by the restrictions on the σ_i's and δ_i's given in (4.5) as follows: we first remember that for two functions f and g we have the relation

$$\partial{\cdot}(fg) = \sigma(f)(\partial{\cdot}g) + \delta(f)g.$$

It follows that

$$\partial_i{\cdot}(P{\cdot}f)(Q{\cdot}g) = \sigma_i(P{\cdot}f)[\partial_i{\cdot}(Q{\cdot}g)] + \delta_i(P{\cdot}f)(Q{\cdot}g) \tag{4.6}$$

which means in terms of operators

$$\begin{aligned}\partial_i(P \otimes Q) &= (A_i(\partial_i)P) \otimes (\partial_i Q) \quad + \quad (B_i(\partial_i)P) \otimes Q && (4.7)\\ &\in \mathbb{O}{\cdot}f \otimes \mathbb{O}{\cdot}g \quad + \quad \mathbb{O}{\cdot}f \otimes \mathbb{O}{\cdot}g && (4.8)\\ &\in \mathbb{T}. && (4.9)\end{aligned}$$

Finally, since f and g are ∂-finite, then $\mathbb{T}$ is a finite-dimensional $\mathbb{K}$-vector space as a product of two finite dimensional vector spaces $\mathbb{O}/I$ and $\mathbb{O}/J$. Hence, the product fg is ∂-finite. □

Remark 4.10

- We note that the Ore operators considered in Table 2.1 satisfy the condition (4.5) of Proposition 4.9.

- The construction of a rectangular system of a ∂-finite product fg may be analogously done as for ∂-finite sums. The only difference is to express, for **each variable** ∂ of $\mathbb{O}$, the derivatives $\partial^l{\cdot}(fg)$ by means of the Leibniz rule in a finite $\mathbb{K}$-basis whose elements are of the form

  $$(\partial^i{\cdot}f)(\partial^j{\cdot}g).$$

  Then, we proceed by Gaussian elimination to get an annihilating polynomial $R(\partial)$ for each variable ∂ of $\mathbb{O}$.

If the ideals $I = \text{Ann}(f)$ and $J = \text{Ann}(g)$ are given by Gröbner bases G_1 and G_2 with respect to the term orders $\prec_1$ and $\prec_2$, then the FGLM algorithm may be extended to compute a Gröbner basis G with respect to a term order $\prec$ of the ideal $\text{Ann}(f \otimes g) \subseteq \text{Ann}(fg)$. This extension is based on an appropriate definition of the normal form as follows

$$NF(1) = \text{RedLNF}(1 \mid G_1) \otimes \text{RedLNF}(1 \mid G_2),$$

and

$$\begin{aligned} NF(\partial_i(P \otimes Q)) &= \text{RedLNF}(A_i(\partial_i)P \mid G_1) \otimes \text{RedLNF}(\partial_i Q \mid G_2) \\ &\quad + \text{RedLNF}(B_i(\partial_i)P \mid G_1) \otimes \text{RedLNF}(Q \mid G_2). \end{aligned}$$

Algorithm 4.3: **Extended FGLM for ∂-finiteness under product**

```
Input  : - A Gröbner bases $G_1$ and $G_2$ w. r. t. $\prec_1$ and $\prec_2$ for annihilating ideals of
           $f_1$ and $f_2$ ;
         - An admissible term order $\prec$ on $\mathbb{O}$;
Output: A Gröbner basis $G$ of $\text{Ann}(f_1 \otimes f_2)$ w. r. t. $\prec$;
begin
    1. Define $NF$ by
           $NF(1) = \text{RedLNF}(1 \mid G_1) \otimes \text{RedLNF}(1 \mid G_2)$,
       and
           $NF(\partial_i(P \otimes Q)) = \text{RedLNF}(A_i(\partial_i)P \mid G_1) \otimes \text{RedLNF}(\partial_i Q \mid G_2)$
               $+\text{RedLNF}(B_i(\partial_i)P \mid G_1) \otimes \text{RedLNF}(Q \mid G_2)$.
    2. Execute FGLM Algorithm;
    return $G$;
end
```

Proposition 4.11 (Closure under the action of Ore operators)
Let $\mathbb{O} = \mathbb{K}[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$ be an Ore algebra and f a ∂-finite function with respect to $\mathbb{O}$. Then, for any operator $P \in \mathbb{O}$ the function $P{\cdot}f$ is also ∂-finite.

Proof: The proof is an immediate consequence of the following inclusion

$$\mathbb{O}{\cdot}(P{\cdot}f) \subseteq \mathbb{O}{\cdot}f,$$

and the ∂-finiteness of f. □

In the case of the action of an Ore operator $P \in \mathbb{O}$ on a ∂-finite function f, we may also extend the FGLM algorithm in order to compute a Gröbner basis of $\text{Ann}(P{\cdot}f)$. If G is a Gröbner basis of an annihilating ideal of f with respect to an admissible term order $\prec$, then we define the normal form of a term $t = \boldsymbol{\partial}^{\boldsymbol{\alpha}} \in \mathbb{O}$ to be:

$$NF(t) := \text{RedLNF}(tP \mid G).$$

Algorithm 4.4: **Extended FGLM for ∂-finiteness under the action of an Ore operator**

```
Input : An operator $P \in \mathbb{O}$ and Gröbner basis $G$ w. r. t. $\prec$ for an annihilating ideal
        of $f$;
Output: A Gröbner basis $G'$ of $\text{Ann}_{\mathbb{O}}(P \cdot f)$ w. r. t. $\prec$;
begin
    1. For each term $t = \partial_1^{\alpha_1} \cdots \partial_r^{\alpha_r} \in \mathbb{O}$ define
       $NF(t) := \text{RedLNF}(P \cdot t \mid G)$;
    2. Execute FGLM Algorithm;
    return $G'$;
end
```

Remark 4.12
We note that the algorithms 4.2 and 4.3 are implemented by F.Chyzak in the Maple package `Holonomy` under the commands `dfinite_add` and `dfinite_mul`.

I introduce the following explanatory example to show stepwise how the FGLM algorithm concepts are applied to compute ∂-finite closure.

Example 4.13 (**∂-finiteness under the action of an Ore operator**)
We consider the Legendre polynomials $P_n(x)$ and define the Ore algebra

$$\mathbb{O} := \mathbb{Q}(n, x)[S_n; S_n, 0][D_x; 1, D_x].$$

Let $\prec$ to be a lexicographical total term order with $D_x \prec S_n$. A Gröbner basis of the annihilating ideal of $P_n(x)$ w.r.t. $\prec$ is given by

$$G_1 = \{(n+1)S_n - (x^2-1)D_x - x(n+1), (x^2-1)D_x^2 + 2xD_x - n(n+1)\}. \tag{4.10}$$

We want to compute a Gröbner basis $G2$ of the annihilating ideal of $Q \cdot P_n(x)$, where

$$Q = Sn + 1.$$

In the *first iteration* we have $T_1 = \{1\}$ and $t_1 = 1$ and

$$NF_1 = \text{RedLNF}((S_n + 1) \cdot 1 \mid G) \tag{4.11}$$
$$= (\frac{x^2-1}{n+1})D_x + (x+1). \tag{4.12}$$

In the *second iteration* we have $T_2 = \{S_n, D_x\}$, $t_2 = \min_\prec T_2 = D_x$ and

$$\begin{aligned} NF_2 &= \text{RedLNF}((S_n+1)\cdot D_x \mid G) && (4.13)\\ &= (x+1)D_x + (n+1). && (4.14)\end{aligned}$$

The resolution of the equation

$$c_1 NF_1 + c_2 NF_2 = 0 \tag{4.15}$$

leads to $\{c_1 = 0, c_2 = 0\}$ which means that the normal forms NF_1 and NF_2 are linearly independent. We continue with the *third iteration*, where $T_3 = \{S_n\} \cup \{S_n D_x, D_x^2\}$, $t_3 = \min_\prec T_3 = S_n$ and

$$\begin{aligned} NF3 &= \text{RedLNF}((S_n+1)\cdot S_n \mid G)\\ &= \frac{(-2-3x+2x^2-2xn-n+3x^3+x^2n+2x^3n)}{(n^2+3n+2)}Dx && (4.16)\\ &+ \frac{(2x^2n+xn-n-1+2x+3x^2)}{(n+2)}.\end{aligned}$$

The resolution of the equation

$$c_1 NF_1 + c_2 NF_2 + c_3 NF_3 = 0 \tag{4.17}$$

leads to the following solution

$$\begin{aligned} c_1 &= (n^2+3n+2)(3x+2nx+1)\\ c_2 &= (n+2)(2x^2-2n+3x^2-3)\\ c_3 &= 2(n+2)(n^2+3n+2)\end{aligned}$$

and thus, $G_2^{(1)} = c_3 S_n + c_2 D_x + c_1$ is an element of G_2. Since $S_n D_x$ is a multiple of the leading monomial S_n of $G_2^{(1)}$ then $T_4 = D_x^2$, $t_4 = D_x^2$ and

$$\begin{aligned} NF4 &= \text{RedLNF}((S_n+1)\cdot D_x^2 \mid G)\\ &= (\frac{xn-n-2}{x-1})D_x + \frac{n^2+n}{x-1}. && (4.18)\end{aligned}$$

The resolution of the equation

$$c_1 NF_1 + c_2 NF_2 + c_3 NF_3 + c_4 NF_4 = 0 \tag{4.19}$$

yields the following solution

$$\begin{aligned} c_1 &= (n+2)(n+1)(2n^2+2x^2n+2n+3x^3+2x)\\ c_2 &= 2(n+2)(n+1)(x+1)\\ c_3 &= 2(n+2)(n+1)(x+1)\\ c_4 &= (x^2-1)(2x^2n+3x^2+2n^2+1+4n).\end{aligned}$$

Hence, the remaining second element of G_2 is

$$G_2^{(2)} = c_4 D_x^2 + c_3 S_n + c_2 D_x + c_1. \tag{4.20}$$

4.2 Holonomic functions

In the last section, we studied ∂-finite functions which are generally solutions of systems of linear operators with rational function coefficients. We have also seen that these functions satisfy some nice closure properties which may be executed algorithmically. However, to understand the operations of summation and integration in the next chapter and in particular to justify the termination of their related algorithms, we need another class of functions called *holonomic*. The theory of holonomy was introduced and studied by Joseph Bernstein [Ber72] to give an elementary proof of a famous conjecture of Gelfand concerning the existence of a meromorphic extension of the distribution complex valued function $z \mapsto P^z$, where P is a polynomial in several variables in $\mathbb{R}^n$. The notion of holonomy which is defined in the context of $\mathcal{D}$-modules is a rather technical notation which is made in terms of graduations and filtrations of algebras and modules. In fact, it translates into a notion of module dimension. We begin by giving a basic and general overview on the theory of graded and filtered modules which forms the landscape in which holonomic functions will be defined

Definition 4.14 (Graded algebra)
Let R be a $\mathbb{K}$-algebra. We say that R is graded if there are $\mathbb{K}$-vector spaces R_i, $i \in \mathbb{N}$ such that

i) $R = \bigoplus_{i\geq 0} R_i$,

ii) $R_iR_j \subseteq R_{i+j}$.

The R_i are called **homogeneous components** of R and the elements of R_i are called homogeneous elements of degree i.

Example 4.15

- The property ii) of a graded algebra means that the product of two homogeneous elements of degree i and j is still homogeneous of degree $i + j$.
- The most important example of a graded algebra is the polynomial ring $\mathbb{K}[x_1, \ldots, x_n]$. The monomials $x_1^{k_1} \cdots x_n^{k_n}$ with $k_1 + \cdots + k_n = m$ form a basis of the homogeneous component of degree m.

A graded algebra admits a special kind of module.

Definition 4.16 (Graded module)
Let $R = \bigoplus_{i\in\mathbb{N}} R_i$ be a graded $\mathbb{K}$-algebra. A left R-module M is a **graded module** if there exist $\mathbb{K}$-vector spaces M_i, for $i \geq 0$, such that

i) $M = \bigoplus_{i\geq 0} M_i$,

ii) $R_iM_j \subseteq M_{i+j}$.

The M_i are the homogeneous components of degree i of M.

Remark 4.17

- Note that the definition of graded module depends on the graded structure chosen for the algebra R.

If we consider the Weyl Algebra $\mathcal{A}_r(\mathbb{K}) = \langle x_1, \dots, x_r, D_{x_1}, \dots, D_{x_r} \rangle$ (see Example 2.9), the degree of an operator (see Remark 2.6) cannot be used to make $\mathcal{A}_r$ into a graded algebra. The problem is that an element like $D_{x_1}x_1$ ought to be homogeneous of degree 2, but it is equal to $x_1D_{x_1} + 1$, which is not homogeneous. To use this degree effectively we must generalize graded algebras to get filtered algebras.

Definition 4.18 (Filtered algebra)
Let R be a $\mathbb{K}$-algebra. A family $\mathcal{F} = \{F_i\}_{i\geq 0}$ of a $\mathbb{K}$-vector spaces is a **filtration** of R if

i) $F_0 \subseteq F_1 \cdots \subseteq R$,

ii) $R = \bigcup_{i\geq 0} F_i$,

iii) $F_iF_j \subseteq F_{i+j}$.

If an algebra has a filtration it is called a **filtered algebra**. It is convenient to use the convention that $F_j = \{0\}$ if $j < 0$.

Remark 4.19
Every graded algebra is filtered. In fact, suppose that $R = \bigoplus_{i\geq 0} R_i$ is a graded algebra and consider the vector spaces $F_k = \bigoplus_{i=0}^{k} R_i$. Clearly $F_k \subseteq F_{k+1}$ and their union is the whole of R. Moreover, since $R_iR_j \subseteq R_{i+j}$ we have that $F_kF_l \subseteq F_{k+l}$. Hence, $\{F_k\}_{k\geq 0}$ is a filtration of R.

There are filtered algebras which do not have natural grading. This is the case for the Weyl algebras, which however, have many different filtrations. In the following example, we focus on the filtration we deal with in this section, the so called **Bernstein filtration**.

Example 4.20 (Bernstein filtration)
The Bernstein filtration of a Weyl algebra $\mathcal{A}_r$ is defined using the degree of operators in $\mathcal{A}_r$. Denote by B_k the set of all operators of $\mathcal{A}_r$ of degree $\leq k$. These are vector subspaces of $\mathcal{A}_r$. Conditions i) and ii) of a filtration are clearly satisfied by the B_k and condition iii) is a consequence of the properties of the degree function defined on $\mathcal{A}_r$ (see Remark 2.6). The Bernstein filtration $\mathcal{B} = \{B_k\}_{k\geq 0}$ has a very special feature that each B_k is a finite dimensional vector space. A basis of B_k is determined by the monomials $\boldsymbol{x}^\alpha\boldsymbol{D}^\beta$ whose total weight $|\alpha| + |\beta| \leq k$. In particular, $B_0 = \mathbb{K}$ and $\{1, x_1, \dots, x_r, D_{x_1}, \dots, D_{x_r}\}$ is a basis of B_1.

We may use a filtration of an algebra to construct a graded algebra. This is very useful because many properties of this graded algebra pass on to its parent filtered one.

Definition 4.21 (Associated graded algebra)
Let R be a $\mathbb{K}$-algebra and $\mathcal{F} = \{F_i\}_{i\geq 0}$ a filtration of R. As a first step in the construction of the graded algebra, we introduce the symbol map of order k, which is a canonical projection of vector spaces

$$\sigma_i : F_i \longrightarrow F_i/F_{i-1}$$

Thus for an element $d \in F_i$, the symbol $\sigma_i(d)$ is nonzero if and only if $d \notin F_{i-1}$.
Consider now the $\mathbb{K}$-vector space

$$gr^{\mathcal{F}} R = \bigoplus_{i \geq 0} F_i / F_{i-1}.$$

We want to make it into a graded algebra. For that it is enough to define the multiplication of two homogeneous elements and extend it by linearity. Let $\sigma_k(a)$ for $a \in F_k$ and $\sigma_l(b)$ for $b \in F_l$ be two homogeneous elements. We define their product by

$$\sigma_k(a)\sigma_l(b) = \sigma_{k+l}(ab).$$

A straightforward verification shows that $gr^{\mathcal{F}} R$ with this multiplication is a graded algebra with homogeneous components F_i/F_{i-1}. This is called the **graded algebra** of R **associated** with the filtration $\mathcal{F}$.

Remark 4.22
We denote the graded algebra of $\mathcal{A}_r$ associated with the Bernstein filtration $\mathcal{B}$ by $\mathcal{S}_r = gr^{\mathcal{B}} \mathcal{A}_r$.

The graded Algebra $\mathcal{S}_r$ hides a surprise which we illustrate in the following theorem.

Theorem 4.23
The graded Algebra $\mathcal{S}_r$ is isomorphic to the commutative polynomial ring over $\mathbb{K}$ in $2r$ variables.

Proof: For a proof of the theorem we refer to [Cou95, Theorem 3.1] $\square$

One of the algebraic objects we are interested in throughout this dissertation are left modules of the Weyl algebra $\mathcal{A}_r$. In the following, we focus on the main properties that these modules may inherit from the filtered structure of $\mathcal{A}_r$ with respect to the Bernstein filtration.

Definition 4.24 (Filtered $\mathcal{A}_r$-module)
Let M be a left A_r-module. A family $\Gamma = \{\Gamma_i\}_{i \geq 0}$ of **finite dimensional** $\mathbb{K}$-vector spaces of M is a filtration of M with respect to the Bernstein filtration $\mathcal{B}$ (see Example 4.20) if it satisfies

i) $\Gamma_0 \subseteq \Gamma_1 \subseteq \cdots \subseteq M$,

ii) $\bigcup_{i \geq 0} \Gamma_i = M$,

iii) $B_i \Gamma_j \subseteq \Gamma_{i+j}$.

The convention that $\Gamma_j = \{0\}$ if $j < 0$ remains in force.

Remark 4.25

- It is clear that $\mathcal{B}$ is a filtration of $\mathcal{A}_r$ as an $\mathcal{A}_r$-module.

- Analogously to Definition 4.21, we may define the graded module associated with a filtered module. Let M be a left $\mathcal{A}_r$-module and Γ be a filtration of M with respect to $\mathcal{B}$. Define the symbol map of order k of the filtration Γ to be the canonical projection
$$\mu_i : \Gamma_i \longrightarrow \Gamma_i/\Gamma_{i-1}.$$
Consider the $\mathbb{K}$-vector space
$$gr^{\Gamma} M = \bigoplus_{i\geq 0} \Gamma_i/\Gamma_{i-1}.$$
We define an action of $\mathcal{S}_r = gr^{\mathcal{B}}\mathcal{A}_r$ on this vector space as follows
$$\sigma_k(a)\mu_l(u) = \mu_{l+k}(au), \quad \text{with } a \in B_k \text{ and } u \in \Gamma_l.$$
The graded $\mathcal{S}_r$-module $gr^{\Gamma} M$ is called the **graded module associated** to the filtration Γ.

- It is not always true that if M is finitely generated over $\mathcal{A}_r$ then $gr^{\Gamma} M$ is finitely generated over $\mathcal{S}_r$. When $gr^{\Gamma} M$ is finitely generated we say that Γ is a **good filtration**. However, it is true that every finitely generated $\mathcal{A}_r$-module admits a good filtration. Indeed, if M is generated by $u_1, \dots, u_s$ then the filtration Γ defined by $\Gamma_i = \sum_{k=1}^{s} B_i u_k$ is a good filtration. The graded module $gr^{\Gamma} M$ is generated over $\mathcal{S}_r$ by the symbols of $u_1, \dots, u_s$.

Definition 4.26 (Induced filtration)
Let M be a left $\mathcal{A}_r$-module with a filtration Γ with respect to $\mathcal{B}$. Suppose that N is a submodule of M. We may use Γ to construct filtrations for both N and M/N as follows

i) A filtration of N is $\Gamma' = \{\Gamma_i \cap N\}_{i\geq 0}$.

ii) A filtration of M/N is $\Gamma'' = \{\Gamma_i/(\Gamma_i \cap N)\}$.

Example 4.27
We have seen in Remark 4.25 that $\mathcal{B}$ is a filtration of $\mathcal{A}_r$ as an $\mathcal{A}_r$-module. Moreover, a left ideal I of $\mathcal{A}_r$ may be seen as a left submodule of $\mathcal{A}_r$. Hence, a filtration $\mathcal{A}_r/I$ is
$$\Gamma = \{\Gamma_i = B_i/(B_i \cap I)\}_{i\geq 0}. \tag{4.21}$$
We note that this filtration is a good filtration which will be used later to define a certain dimension of $\mathcal{A}_r/I$ and will lead to the definition of holonomic functions.

Using graded and filtered modules introduced previously, we define a dimension for $\mathcal{A}_r$-modules. This dimension is independent of the chosen filtration, and in this sense it is a combinatorical invariant that characterizes the considered $\mathcal{A}_r$-module.
In the following we state a result from commutative algebra that is the key for the definition of this dimension.

Theorem 4.28 (Hilbert polynomial)
Let $M = \bigoplus_{i \geq 0} M_i$ be a finitely generated graded module over the polynomial ring $\mathbb{K}[x_1, \ldots, x_n]$. There exists a polynomial $HP_M(t) \in \mathbb{Q}[t]$ and a positive integer N such that

$$\sum_{i=0}^{s} \dim_{\mathbb{K}}(M_i) = HP_M(s)$$

for every $s \geq N$.
The polynomial HP_M is known as the **Hilbert polynomial** of the module M.

Now, we are ready to define the dimension of an $\mathcal{A}_r$-module.

Definition 4.29 (Bernstein dimension)
Let M be a finitely generated left $\mathcal{A}_r$-module. Suppose that Γ is a good filtration of M with respect to the Bernstein filtration $\mathcal{B}$, that is, the graded module $gr^{\Gamma} M$ is finitely generated over $\mathcal{S}_r = gr^{\mathcal{B}} \mathcal{A}_r$ (see Remark 4.25). Denote by $HP(t, \Gamma, M)$ the Hilbert polynomial of the graded module $gr^{\Gamma} M$ over the polynomial ring $\mathcal{S}_r$ (see Theorem 4.23). Hence, by Theorem 4.28, there exists an integer N such that for $s \geq N$

$$HP(s, \Gamma, M) = \sum_{i=0}^{s} \dim_{\mathbb{K}}(\Gamma_i / \Gamma_{i-1}) = \dim_{\mathbb{K}}(\Gamma_s). \tag{4.22}$$

The last equation in (4.22) follows from the fact that $\dim_{\mathbb{K}}$ is additive over exact sequences of vector spaces. The degree of $HP(t, \Gamma, M)$ is called the **Bernstein dimension** of M and denoted by $d(M)$.

Remark 4.30
It is shown in [Cou95, Section 9.2] that the Bernstein dimension is independent of the chosen good filtration, and thus one is able to talk about the dimension of an $\mathcal{A}_r$-module M without reference to any particular filtration.

Example 4.31 (Bernstein dimension of $\mathcal{A}_r$)
Let M be the $\mathcal{A}_r$-module $\mathcal{A}_r$. The Bernstein filtration $\mathcal{B}$ is a good filtration of M and it is possible to compute explicitly $HP(s, \mathcal{B}, M)$ in this case. By (4.22), we should determine the dimension of B_s. But the set of monomials $\{x^{\alpha} D_x^{\beta}\}$ with $|\alpha| + |\beta| \leq s$ form a basis of B_s as a $\mathbb{K}$-vector space. So it is enough to compute the elements of this basis. To do this we should compute the non-negative solutions of the inequality

$$\alpha_1 + \cdots + \alpha_r + \beta_1 + \cdots + \beta_r \leq s. \tag{4.23}$$

It is known from combinatorics that (4.23) admits $\binom{s+2r}{2r}$ such solutions. Hence,

$$HP(s, \mathcal{B}, M) = \binom{s+2r}{2r}$$

which is a polynomial of degree $2r$ in s. Thus $d(\mathcal{A}_r)$ is $2r$.

Example 4.32 (Computation of Bernstein dimension)
Let $I = \langle xD_x - 3, D_x^4 \rangle$ be a left ideal of $\mathcal{A}_1$. We want to compute the Bernstein dimension of the left $\mathcal{A}_1$-module $M = \mathcal{A}_1/I$. We know from Example 4.27 that

$$\Gamma = \{\Gamma_i = B_i/(B_i \cap I)\}_{i \geq 0}$$

is a good filtration of M with respect to the Bernstein filtration $\mathcal{B}$. Moreover, it is not difficult to show that

$$\dim_{\mathbb{K}}(\Gamma_s) = \dim_{\mathbb{K}}(B_s) - \dim_{\mathbb{K}}(B_s \cap I) \quad \text{for } s \geq 0.$$

On the other hand, we know that B_s is generated as a $\mathbb{K}$-vector space by the set of monomials

$$\{x^\alpha D_x^\beta \,;\, 0 \leq \alpha, \beta \leq s\}.$$

A $\mathbb{K}$-basis of $B_s \cap I$ is the set of all monomials which are contained in B_s and not in I, that is,

$$\{x^\alpha D_x^\beta \,;\, 1 \leq \alpha, \beta \leq s \text{ und } \alpha + \beta \leq s\}, \quad \text{for } 0 \leq s \leq 3.$$

and

$$\{x^\alpha D_x^\beta \,;\, 1 \leq \alpha, \beta \leq s \text{ und } \alpha + \beta \leq s\} \cup \{D_x^s\}, \quad \text{for } s \geq 4.$$

We summarize the computations in the following table.

s	0	1	2	3	4	5	$\cdots$
$\dim_{\mathbb{K}}(\Gamma_s)$	1	3	5	7	8	9	$\cdots$

An observation of the table shows that for $s \geq 3$ the $\mathbb{K}$-dimension of Γ_s is $s + 4 = HP(s, \Gamma, M)$, the Hilbert polynomial of M which is of degree one in s. Hence, the Bernstein dimension of $M = \mathcal{A}_1/I$ is one.

Now, we are ready to present an important result due to Bernstein [Ber72, Theorem 1.3] , in which he established a lower bound on the Bernstein dimension of a finitely generated $\mathcal{A}_r$-module.

Theorem 4.33 (Bernstein's inequality)
If M is a finitely generated nonzero left $\mathcal{A}_r$-module, then the Bernstein dimension of M is greater or equal to r.

Proof: For a proof of the theorem, the reader may refer to the original paper [Ber72, Theorem 1.3] or to the book [Cou95, Theorem 9.4.2]. □

Remark 4.34
In the previous theorem, a lower bound was given on the Bernstein dimension of a left $\mathcal{A}_r$-module because next, we will be more interested in those with minimal dimension r. However, we have to mention that there exists also an upper bound which is $2r$ (see [Cou95, Corollary 9.3.4]).

The non-commutative structure is so strong that the ring $\mathcal{A}_r$ has no two-sided ideals, except the zero-ideal and the whole ring $\mathcal{A}_r$. But of course there are one-sided left and right ideals. The introduction on the theory of filtered modules presented previously is essentially aimed to characterize, in a $\mathcal{D}$-module context, a class of left-ideals in $\mathcal{A}_r$, which we present in the following definition.

Definition 4.35 (Holonomic ideal)
A left ideal I in $\mathcal{A}_r$ is called **holonomic** if it is zero, or the Bernstein dimension of the left $\mathcal{A}_r$-module $M = \mathcal{A}_r/I$ is minimal, that is, by Bernstein's inequality (Theorem 4.33), $d(M) = r$.

Definition 4.36 (Holonomic function)
An object f on which the Weyl algebra $\mathcal{A}_r$ may act by left multiplication, is called a **holonomic function**, if there exists a holonomic left ideal in $\mathcal{A}_r$ that annihilates f.

A wonderful source of examples of holonomic ideals is given by the following result.

Corollary 4.37
Every left ideal $I \neq 0$ of the Weyl algebra $\mathcal{A}_1$ is holonomic.

Proof: By Theorem 4.33 and Remark 4.34 the Bernstein dimension of the module $M = \mathcal{A}_1/I$ satisfies $1 \leq d(M) \leq 2$. Since the Bernstein dimension 2 is reached only by the module $\mathcal{A}_1$ itself, that is, for $I = 0$ then $d(M)$ must be one and therefore, the ideal I is holonomic. □

In the following proposition, we present a nice property of holonomic ideals that is crucial for the termination of many algorithms of the next chapter.

Proposition 4.38 (Elimination property)
Let I be a left holonomic ideal of $\mathcal{A}_r$. For every $r+1$ generators out of the $2r$ generators $\{x_1, \dots, x_r, D_1, \dots D_r\}$ of $\mathcal{A}_r$ there exists a nonzero operator of I that only depends on these $r+1$ generators.

Proof: We follow the proof given by Zeilberger in [Zei90b] to whom it is was shown by Bernstein himself. Without loss of generality, let us take the $r+1$ generators to be the set $\{x_1, \dots, x_r, D_1\}$ and denote by S the subalgebra of $\mathcal{A}_r$ generated by this set. Consider the mapping

$$\phi : S \longrightarrow \mathcal{A}_r/I, \quad P \mapsto P \text{ mod } I.$$

Since I is holonomic, by Definition 4.35 the Bernstein dimension of $\mathcal{A}_r/I$ is r. Moreover, we have seen in Example 4.31 that the Bernstein dimension of S is $r+1$. By Theorem 4.28 and with respect to the Bernstein filtration this leads to, for $s \gg 0$

$$\begin{aligned} \dim_{\mathbb{K}} B_s &= \mathcal{O}(s^{r+1}), \\ \dim_{\mathbb{K}} B_s/(B_s \cap I) &= \mathcal{O}(s^r). \end{aligned}$$

Hence, for $s \gg 0$ the restriction of the linear map ϕ to the finite dimensional vector space B_s is a linear transformation from a higher-dimensional vector space to a lower-dimensional vector space, and its kernel must therefore be nonzero. But this kernel is precisely $I \cap S$. □

Remark 4.39 (Holonomic canonical representation)

- A holonomic function f may be given by any generating set of its annihilating ideal with appropriate initial conditions. However, in general it is not clear how many initial conditions are required to uniquely specify the function. The above proposition guarantees, in particular, the existence of r operators

$$P_i(D_{x_i}, x_1, \dots, x_r), \quad i = 1, \dots, r. \tag{4.24}$$

  of order n_i in D_{x_i} that annihilate f. These operators with r initial conditions

$$D_{x_1}^{i_1} D_{x_2}^{i_2} \cdots D_{x_r}^{i_r} f(x_0), \quad 0 \le i_1 \le n_1, \dots, 0 \le i_r \le n_r \tag{4.25}$$

  specify uniquely f and it is called **a canonical holonomic representation system** of f. It is also important to note that the point x_0 should not be a common zero of the leading coefficients of the operators P_i.

- The canonical representation (4.24), with its accompanying initial conditions is **not unique**. Given such a representation, we can left-multiply by any operator in $(D_{x_i}, x_1, \dots, x_r)$ to get higher-order equations and add the appropriate number of initial conditions. Therefore, it is desirable that a holonomic function may be given by a canonical representation with minimal order in D_{x_i} and coefficients in $(x_1, \dots, x_r)$ without common factors.

- Proposition 4.38 is a proof of the **existence** of a canonical holonomic representation of f. This representation can be computed from a generating set of the annihilating ideal of f, by aimed elimination of the variables D_{x_i} using Gröbner bases techniques (see Chapter 3).

In Definition 4.36 of a holonomic function, we considered all objects on which the Weyl algebra may act. One of the objects we are concerned with in this thesis are sequences. But how the Weyl algebra may act on them?
We first consider the case of sequences in one variable $(u_k)_{k\in\mathbb{N}}$. Moreover, we denote the *generating function* of $(u_k)_{k\in\mathbb{N}}$ by

$$f(x) = \sum_{k=0}^{\infty} u_k x^k. \tag{4.26}$$

Multiplying $f(x)$ by x we get

$$xf(x) = \sum_{k=0}^{\infty} u_k x^{k+1} = \sum_{k=1}^{\infty} u_{k-1} x^k \tag{4.27}$$

which means that this action of x on $f(x)$ induces a backward shift action on the sequence $(u_k)_{k\in\mathbb{N}}$. Thus,

$$x \leftrightarrow S_k^{-1}. \tag{4.28}$$

Similarly, multiplying $f(x)$ by D_x yields

$$D_x f(x) = \sum_{k=0}^{\infty} k u_k x^{k-1} = \sum_{k=0}^{\infty} (k+1) u_{k+1} x^k. \tag{4.29}$$

Hence, we get the relation

$$D_x \leftrightarrow (k+1)S_k. \tag{4.30}$$

The relations (4.28) and (4.30) leads also to

$$\begin{aligned} k &\leftrightarrow xD_x, \\ S_k &\leftrightarrow x^{-1}. \end{aligned} \tag{4.31}$$

Definition 4.40 (Holonomic sequence)
A multivariate sequence $(\mathcal{U}_k)_{k\in\mathbb{N}^r}$ is called holonomic if its multivariate generating function

$$f(\boldsymbol{x}) = \sum \mathcal{U}_{\boldsymbol{k}}\, \boldsymbol{x}^{\boldsymbol{k}}, \quad \boldsymbol{k} = (k_1,\ldots,k_r), \quad \boldsymbol{x} = (x_1,\ldots,x_r) \tag{4.32}$$

is holonomic. Moreover the action of the Weyl algebra $\mathcal{A}_r$ on $(\mathcal{U}_k)_{k\in\mathbb{N}^r}$ is defined by the following relations

$$x_i \leftrightarrow S_{k_i}^{-1}, \quad D_{x_i} \leftrightarrow (k_i+1)S_{k_i}, \quad S_{k_i} \leftrightarrow x_i^{-1} \quad \text{and} \quad k_i \leftrightarrow x_i D_{x_i}. \tag{4.33}$$

Hence, by means of the relations (4.33) the elimination property may be directly carried into the shift context. In fact, the holonomy of a sequence $(\mathcal{U}_k)_{k\in\mathbb{N}^r}$ guarantees by Proposition 4.38 the existence of an operator $P(x_1,\ldots,x_r,D_r)$ that annihilates it, which is, using the relations (4.33), the operator $P(S_{k_1}^{-1},\ldots,S_{k_r}^{-1},(k_r+1)S_{k_r})$. Getting rid of the dominators in the S_{k_i}'s we get a nontrivial shift operator $Q(k_r, S_{k_1}\ldots,S_{k_r})$.

Example 4.41
The sequence $u_k = k!$ satisfies the following recurrence equation

$$k\,u_{k-1} - u_k = 0, \tag{4.34}$$

which in operator notation yields

$$kS_k^{-1} - 1. \tag{4.35}$$

Substituting the relations (4.31) we get a differential operator

$$(xD_x)x - 1 = x(D_x x) - 1 = x(xD_x + 1) - 1 = x^2 D_x + x + 1 \tag{4.36}$$

annihilating the generating function $f(x)$ of u_k and generates a left ideal in $\mathcal{A}_1(\mathbb{C})$. Hence $f(x)$ is holonomic by Corollary 4.37 and consequently u_k too.

Closure properties

Similarly to ∂-finite functions, holonomic functions share some nice closure properties which may be proved using the theory of filtered modules presented previously. For instance, if f and g are holonomic, then the sum $f+g$ and the product fg are also holonomic. Furthermore, if $f(x_1,\ldots,x_r)$ is holonomic then $f(x_1,\ldots,x_{r-1},c)$, $c \in \mathbb{K}$, and $\int f(x_1,\ldots,x_r)\,dx_r$ are holonomic in $(x_1,\ldots,x_r)$. Holonomic sequences fulfil analogeous statements, where the integral is

replaced by the summation quantifier. For detailed proofs of these properties, the reader may refer to [Zei90b] or [Cou95].
Unfortunately, this similarity disappears when we eventually want to execute algorithmically these closure properties. Indeed, the computation of these closures should rely on holonomic ideal representations of the given functions. However, computing such annihilating ideals in the corresponding Weyl algebra is a hot topic of $\mathcal{D}$-module theory and still a nontrivial task, both on the theoretical or algorithmic level. Recently, algorithms for determining the complete holonomic annihilating ideal of f^s in the Weyl algebra $\mathcal{A}_r$, where f is a polynomial in $\mathbb{K}[x_1, \dots, x_r]$, have been designed in [Oak97, SST00] and implemented in [MML08].

4.3 Conclusion

In the last sections, we introduced two classes of functions, namely holonomic and ∂-finite functions. This section gives a comparison between the two classes and arguments the need of these two notions.
One obvious difference is that the definition of ∂-finite functions concerns all kinds of Ore operators, whereas holonomy is essentially defined for the differential setting and may be extended to the shift setting by means of the generating function (see Equation (4.32)).
The reason of not restricting ourselves to just one class is that we need certain properties of either classes. On one hand, the description of ∂-finite functions by rectangular systems is easier to obtain and allows the algorithmic execution of their closure properties (see Subsection 4.1.2). On the other hand, the elimination property (see Proposition 4.38), which is crucial for the termination of the algorithms of the next chapter, requires holonomy. Thereby, we are more concerned with the functions lying in the intersection of the two classes. The following theorem which is due to Masaki Kashiwara, confirms that in the differential setting the two classes in fact coincide.

Theorem 4.42
Let $\mathcal{A}_r(\mathbb{K})$ be the Weyl algebra in $\boldsymbol{x} = x_1, \dots, x_r$ and let $\mathbb{O}_{\text{rat}}$ be the rational differential Ore algebra $\mathbb{K}(\boldsymbol{x})[\boldsymbol{D_x}; \boldsymbol{1}, \boldsymbol{D_x}]$. A left ideal I of $\mathbb{O}_{\text{rat}}$ is ∂-finite if and only if $I \cap \mathcal{A}_r$, which is also a left ideal of $\mathcal{A}_r$, is holonomic.

Proof: One direction is obvious and is in fact a consequence of the elimination property of holonomic ideals. If I is a holonomic left ideal of $\mathcal{A}_r$ then by Proposition 4.38, there exists for each $1 \leq i \leq r$ a nonzero operator that involves only D_{x_i} and none of the remaining differential operators D_{x_j} with $j \neq i$. These operators form a rectangular system for I and thus, the left ideal I is ∂-finite.
The converse direction is less obvious and we will not give a proof here. Besides the original paper of Kashiwara [Kas78], we may refer the reader to [Tak92], where a more elementary proof is given. □

Whereas the notions of ∂-finiteness and holonomy coincide in the differential setting, unfortunately this is not in general the case in the multivariate shift setting. The following example given by Herbert S. Wilf and Doron Zeilberger in [WZ91] illustrates this fact.

Example 4.43 (Non-holonomic Multivariate Sequence)
The hypergeometric sequence $\mathcal{U}_{n,k} = \frac{1}{n^2+k^2}$ is ∂-finite and it is given by the rectangular system

$$((n+1)^2 + k^2)S_n - (n^2 + k^2)$$
$$((k+1)^2 + n^2)S_k - (n^2 + k^2),$$

which are the generators of the annihilating ideal of $\mathcal{U}_{n,k}$ in the rational Ore algebra

$$\mathbb{O} = \mathbb{Q}(n,k)[S_n; S_n, 0][S_k; S_k, 0].$$

Suppose now that $\mathcal{U}_{n,k}$ is holonomic, then by the elimination property (Proposition 4.38) there exists a nonzero k-free operator $Q(n, S_n, S_k)$ annihilating $\mathcal{U}_{n,k}$. Hence, we get a recurrence equation of the form

$$\sum_{i,j\geq 0} \frac{p_{ij}(n)}{(n+i)^2 + (k+j)^2} = 0, \tag{4.37}$$

where p_{ij} are polynomials which are not all zeros. Moreover, each dominator $(n+i)^2 + (k+j)^2$ considered as a polynomial in k involves two poles which are pairwise distinct. Hence, the sum cannot cancel unless all p_{ij} are zeros. Therefore, no such a recurrence exists and $\mathcal{U}_{n,k}$ is not holonomic.

In this thesis, we are essentially concerned with functions that are both ∂-finite and holonomic. These functions are in general given by a ∂-finite ideal I_r in the multivariate rational Ore algebra

$$\mathbb{O}_r = \mathbb{K}(\boldsymbol{x})[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}].$$

However, in the algorithms of the next chapter, we will need to eliminate one of the variables x_i using Gröbner bases techniques. To perform this elimination, we would like to determine from the ∂-finite ideal I_r the holonomic ideal in the polynomial Ore algebra

$$\mathbb{O}_p = \mathbb{K}[\boldsymbol{x}][\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$$

corresponding to the given function f. This ideal is

$$I_p = I_r \cap \mathbb{O}_p.$$

In the pure differential case, the ideal I_p is named Weyl closure and has been solved by Harrison Tsai in the univariate [Tsa00b] and multivariate case [Tsa00a]. The computation of I_p when shift operators are involved is still an open problem. In practice, if $\{p_1, \dots, p_s\}$ is a generating set of I_r in $\mathbb{O}_r$, then after getting rid of the denominators in the p_i's by multiplying through, we get a set $\{p'_1, \dots, p'_s\}$ that generates an ideal $I' \subseteq \mathbb{O}_p$. The ideal I' is in general only a subideal of I_p and may not give a complete holonomic description of the function f. Therefore, using this ideal to perform the elimination of one of the x_i's to get an x_i-free operator may fail, even if we know by the elimination property that such an operator must exist. This phenomenon is called **extension /contraction problem**. Let's illustrate this fact by the following examples

Example 4.44
Consider the sequence $\mathcal{U}_{n,k} = \binom{n}{k}$ which is annihilated by the ideal I_r generated by the operators

$$\begin{aligned} P &= (n+1-k)S_n - (n+1) \\ Q &= (k+1)S_k - (n-k) \end{aligned}$$

in the rational Ore algebra $\mathbb{O}_r = \mathbb{K}(n,k)[S_n;S_n,0][S_k;S_k,0]$. Pascal's Triangle rule represented by the operator

$$R = S_nS_k - S_k - 1$$

is an element of I_r, since

$$R = \frac{1}{n-k}(S_kP + Q).$$

However, R is not contained in the ideal I_p of the polynomial Ore algebra $\mathbb{O}_p = \mathbb{K}[n,k][S_n;S_n,0][S_k;S_k,0]$ generated by $\{P,Q\}$.

Example 4.45
The function $f(x) = x^3$ is ∂-finite and its annihilating ideal I_r is generated by the operator

$$P = xD_x - 3$$

in the rational Ore algebra $\mathbb{O}_r = \mathbb{K}(x)[D_x;1,D_x]$. By Corollary 4.37 $f(x)$ is also holonomic. The operator D_x^4 is contained in I_r, since

$$D_x^4 = (\frac{1}{x}D_x^3)P.$$

but it is not an element of the ideal $I_p \subseteq \mathbb{O}_p = \mathbb{K}[x][D_x;1,D_x]$ generated by P.

Chapter 5

Algorithms for Summation and Integration

5.1 Creative Telescoping

In his paper *A holonomic systems approach to special functions identities* [Zei90b] and relying on the works of Celine Fasenmyer, [Fas47, Fas49] Doron Zeilberger presented a method to compute recurrence or differential equations satisfied by definite sums or integrals of holonomic functions. This method is called *creative telescoping*. To describe the idea of this method we consider a holonomic function $f(t, \boldsymbol{x})$ in the variables t and $\boldsymbol{x} = x_1, \dots, x_s$. The main ingredient of creative telescoping is an annihilating operator of f of the form

$$T = P(\boldsymbol{x}, \boldsymbol{\partial_x}) + \partial_t \cdot Q(t, \boldsymbol{x}, \partial_t, \boldsymbol{\partial_x}) \tag{5.1}$$

which lies in the Ore algebra

$$\mathbb{O}_p = \mathbb{K}(\boldsymbol{x})[t][\boldsymbol{\partial_x}; \boldsymbol{\sigma_x}, \boldsymbol{\delta_x}][\partial_t; \sigma_t, \delta_t]$$

and where ∂_t and $\boldsymbol{\partial_x} = \partial_{x_1}, \dots, \partial_{x_s}$ refer to Ore operators acting on the variables t and $\boldsymbol{x}$, respectively.
In the differential setting, t is a continuous variable and the operator ∂_t stands for the differential operator $D_t = \frac{dt}{t}$. In this case, applying

$$T = P(\boldsymbol{x}, \boldsymbol{\partial_x}) + D_t \cdot Q(t, \boldsymbol{x}, D_t, \boldsymbol{\partial_x}) \tag{5.2}$$

to the definite integral

$$F(\boldsymbol{x}) = \int_a^b f(t, \boldsymbol{x}) dt$$

we get an equation of the form

$$\begin{aligned} 0 &= P(\boldsymbol{x}, \boldsymbol{\partial_x}) \cdot F(\boldsymbol{x}) + [Q(t, \boldsymbol{x}, D_t, \boldsymbol{\partial_x}) f(t, \boldsymbol{x})]_{t=a}^{t=b} \\ &= P(\boldsymbol{x}, \boldsymbol{\partial_x}) \cdot F(\boldsymbol{x}) + R(\boldsymbol{x}, \boldsymbol{\partial_x}). \end{aligned} \tag{5.3}$$

In the difference setting, t denotes a discrete variable and the operator ∂_t stands for the difference operator $\triangle_t = S_t - 1$. Hence an annihilating operator of $f(t, \boldsymbol{x})$ is

$$T = P(\boldsymbol{x}, \boldsymbol{\partial_x}) + \triangle_t \cdot Q(t, \boldsymbol{x}, \triangle_t, \boldsymbol{\partial_x}) \tag{5.4}$$

which we apply to the definite sum

$$S(\boldsymbol{x}) = \sum_{t=a}^{b} f(t, \boldsymbol{x})$$

to get an equation of the form

$$\begin{aligned} 0 &= P(\boldsymbol{x}, \boldsymbol{\partial_x}) \cdot S(\boldsymbol{x}) + [Q(t, \boldsymbol{x}, \triangle_t, \boldsymbol{\partial_x}) f(t, \boldsymbol{x})]_{t=a}^{t=b+1} \\ &= P(\boldsymbol{x}, \boldsymbol{\partial_x}) \cdot S(\boldsymbol{x}) + R(\boldsymbol{x}, \boldsymbol{\partial_x}). \end{aligned} \tag{5.5}$$

We note that the equations (5.3) and (5.5) are inhomogeneous because of R. An interesting case of integrals and sums for which R vanishes is described in the following definition

Definition 5.1
We say that the integral $\int_a^b f(t, \boldsymbol{x}) dt$ and the sum $\sum\limits_{t=a}^{b} f(t, \boldsymbol{x})$ have **natural boundaries**, if for any operator $Q \in \mathbb{O}$ we have $[Q \cdot f]_{t=a}^{t=b} = 0$ and $[Q \cdot f]_{t=a}^{t=b+1} = 0$, respectively.

Typical examples of natural boundaries are sums with finite support, integrals over the whole real line such that $\lim\limits_{t \to \pm\infty} t^{\alpha} \boldsymbol{x}^{\beta} D_t^{\gamma} \boldsymbol{D_x}^{\delta} f(t, \boldsymbol{x}) = 0$, or contour integrals over closed paths.
By the method of creative telescoping, it is preferably looked for a homogeneous equation satisfied by the expression in question (integral or sum). In the case of natural boundaries, the inhomogeneous part R in the equations (5.3) and (5.5) evaluates to zero and $P(\boldsymbol{x}, \boldsymbol{\partial_x})$ is the searched annihilating operator of the expression. Otherwise, we may get rid of the inhomogeneity by computing a left annihilating operator C of $Q \cdot f$ using Proposition 4.11. Hence, the operator $C \cdot P$ annihilates the sum or the integral in question.
The method of creative telescoping may be applied to compute annihilating operators for multiple sums and integrals. For this purpose, we should determine, for given holonomic function $f(\boldsymbol{t}, \boldsymbol{x})$, an annihilating operator of the form

$$T = P(\boldsymbol{x}, \boldsymbol{\partial_x}) + \partial_{t_1} Q_1(\boldsymbol{t}, \boldsymbol{x}, \boldsymbol{\partial_t}, \boldsymbol{\partial_x}) + \cdots + \partial_{t_l} Q_l(\boldsymbol{t}, \boldsymbol{x}, \boldsymbol{\partial_t}, \boldsymbol{\partial_x}), \tag{5.6}$$

where $\boldsymbol{\partial_t} = (\partial_{t_1}, \ldots, \partial_{t_l})$ and $\boldsymbol{\partial_x} = (\partial_{x_{l+1}}, \ldots, \partial_{x_s})$ are differential or difference operators acting on the variables $\boldsymbol{t} = (t_1, \ldots, t_l)$ and $\boldsymbol{x} = (x_{l+1}, \ldots, x_s)$, respectively.
Until now, we explained the principle of creative telescoping without mentioning how to determine the operator T in equations (5.3) and (5.5). For the determination of T we present two algorithms. The first one is due to Zeilberger [Zei90b] and he named it “slow algorithm”, in opposite to his algorithm which treats the special case of single sums and integrals of proper hypergeometric terms, which he called “fast algorithm” [Zei90a]. The second algorithm is due Nobuki Takayama [Tak90a, Tak90b] in the context of pure Weyl algebras and was optimized and extended to more general ore algebras by Frédéric Chyzak and Bruno Salvy [CS96].

5.1.1 Zeilberger's Slow Algorithm

In this algorithm a more restrictive form of the operator T is required

$$T = P(\boldsymbol{x}, \boldsymbol{\partial_x}) + \partial_t \cdot Q(\boldsymbol{x}, \partial_t, \boldsymbol{\partial_x}), \tag{5.7}$$

where the coefficients of the operators P and Q are free of the variable t. To determine the operator T in (5.7) we should have as input an annihilating ideal I for a given holonomic function $f(t, \boldsymbol{x})$. The main step of this algorithm is to use the generators of I to get a new operator $T \in I$ not containing the variable t. In this case, the operator ∂_t will commute with all the variables of T and we may proceed by left Skew Euclidean division (see Section 2.3) to write T in the form (5.7), where the operator P is simply the remainder of this division. To eliminate the variable t, Zeilberger in [Zei90b] used a noncommutative version of "Sylvester's dialytic elimination". However, he also mentioned that the application of noncommutative Gröbner bases to perform this elimination may lead to better results. The use of Gröbner bases requires first to fix an appropriate elimination ordering for the variable t, that is, a total ordering $\prec$ where t is lexicographically greater than the variables $\boldsymbol{x}$ and the acting operators ∂_t and $\boldsymbol{\partial_x}$

$$t \succ \boldsymbol{x} \succ \partial_t \succ \boldsymbol{\partial_x}.$$

Afterwards, we compute a Gröbner basis G of the ideal I with respect to $\prec$ in the polynomial Ore algebra

$$\mathbb{O}_p = \mathbb{K}(\boldsymbol{x})[t][\boldsymbol{\partial_x}; \boldsymbol{\sigma_x}, \boldsymbol{\delta_x}][\partial_t; \sigma_t, \delta_t] \tag{5.8}$$

to get, if it exists, the operator T among the generators of G. Thus, an annihilating operator $P(\boldsymbol{x}, \boldsymbol{\partial_x})$ for $\int_a^b f(t, \boldsymbol{x})dt$ and $\sum_{t=a}^b f(t, \boldsymbol{x})$ by simply substituting ∂_t to zero (see equations (5.2) and (5.4)).
We note that the success of this algorithm drastically depends on the existence of the operator $T \in I$, which is guaranteed by the elimination property (Proposition 4.38) only if I is holonomic. Unfortunately, there are no known algorithmic or even theoretic methods to compute a complete annihilating holonomic ideal for a given function f, except (as far as I know) for the special case where $\mathbb{O}_p$ is the Weyl algebra $\mathcal{A}_r$ and $f = g^n$ with g a polynomial in $\mathbb{K}[x_1, \ldots, x_r]$ (see [MML08]). In practice, the ideal I is a ∂-finite description of the function f in the rational Ore algebra

$$\mathbb{O}_r = \mathbb{K}(\boldsymbol{x}, t)[\boldsymbol{\partial_x}; \boldsymbol{\sigma_x}, \boldsymbol{\delta_x}][\partial_t; \sigma_t, \delta_t]$$

and we have seen in the last chapter how we can algorithmically compute this ideal for algebraic operations preserving ∂-finiteness. However, what we effectively need to perform Gröbner basis computation is the ideal $I_p = I \cap \mathbb{O}_p$. The determination of I_p is partially solved in the context of pure differential operators (Weyl algebra) and still an open problem when other Ore operators are involved, in particular shift operators. What we effectively do in practice to bypass this problem is getting rid of the denominators of the coefficients of the generators of I to generate an annihilating ideal $\mathfrak{I} \subset \mathbb{O}_p$ of the function f. The ideal $\mathfrak{I}$ is in general only a subideal of I_p and hence we may be confronted with the extension/extraction problem described in Section 4.3 and the Gröbner basis computation may fail to return the operator T.

Example 5.2
In this example, we want to prove the following identity

$$\sum_{k=0}^{n} \binom{n}{k}^3 = \sum_{k=0}^{n} \binom{n}{k}^2 \binom{2k}{n} \tag{5.9}$$

(see [Str93]) using Gröbner basis elimination techniques. The summand $\binom{n}{k}^3$ is annihilated by the following operator system

$$REk1 = (1+k)^3 S_k + (k-n)^3 \tag{5.10}$$
$$REn1 = (1-k+n)^3 Sn - (1+n)^3, \tag{5.11}$$

and the summand $\binom{n}{k}^2\binom{2k}{n}$ is annihilated by the following one

$$REk2 = (2+2k-n)(1+2k-n)(1+k)S_k - 2(k-n)^2(1+2k), \tag{5.12}$$
$$REn2 = (-1+k-n)^2 S_n + (1+n)(2k-n). \tag{5.13}$$

The polynomials of each system generate a left ideal in the Ore algebra

$$\mathbb{O}_p = \mathbb{Q}(n)[k][Sn; Sn, 0][Sk; Sk, 0]. \tag{5.14}$$

Maple Session 5.1 (Proof of the identity: $\sum_{k=0}^{n} \binom{n}{k}^3 = \sum_{k=0}^{n} \binom{n}{k}^2 \binom{2k}{n}$)

We begin by loading the packages we need

```
> read "hsum15.mpl";
```

$$\textit{Package “Hypergeometric Summation“, Maple V – Maple 15}$$
$$\textit{Copyright 1998 – 2012, Wolfram Koepf, University of Kassel}$$

```
> with(Ore_algebra):
> with(Groebner):
```

and define the Ore algebra where the Gröbner basis computation will be performed.

```
> A:=skew_algebra(shift=[Sn,n],shift=[Sk,k],polynom=k);
```

$$A := \mathit{Ore\_algebra}$$

Using the following procedure

```
> HolonomicRE:=proc(term,sk)
> local s,k,r;
> s:=op(0,sk):  k:=op(1,sk):
> r:=ratio(term,k);
> denom(r)*s(k+1)-numer(r)*s(k)=0;
> end proc:
```

we determine annihilating operator systems of the summands $\binom{n}{k}^3$ and $\binom{n}{k}^2\binom{2k}{n}$, respectively.

```
> rek1:=HolonomicRE(binomial(n,k)^3,S(k));
```

$$\mathit{rek1} := (k+1)^3\,\mathrm{S}(k+1) + (-n+k)^3\,\mathrm{S}(k) = 0$$

```
> ren1:=HolonomicRE(binomial(n,k)^3,S(n));
```

$$\mathit{ren1} := (-n-1+k)^3\,\mathrm{S}(n+1) + (n+1)^3\,\mathrm{S}(n) = 0$$

```
>  rek2:=HolonomicRE(binomial(n,k)^2*binomial(2*k,n),S(k));
```

$$\mathit{rek2} := (2\,k+2-n)\,(-n+2\,k+1)\,(k+1)\,\mathrm{S}(k+1)$$
$$-\,2\,(-n+k)^2\,(2\,k+1)\,\mathrm{S}(k) = 0$$

```
>  ren2:=HolonomicRE(binomial(n,k)^2*binomial(2*k,n),S(n));
```

$$\mathit{ren2} := (-n-1+k)^2\,\mathrm{S}(n+1) - (n+1)\,(-n+2\,k)\,\mathrm{S}(n) = 0$$

Written in operator notation:

```
>  REk1:=subs({S(k+1)=Sk,S(k)=1},op(1,rek1));
```

$$\mathit{REk1} := (k+1)^3\,\mathit{Sk} + (-n+k)^3$$

```
>  REn1:=subs({S(n+1)=Sn,S(n)=1},op(1,ren1));
```

$$\mathit{REn1} := (-n-1+k)^3\,\mathit{Sn} + (n+1)^3$$

```
>  REk2:=subs({S(k+1)=Sk,S(k)=1},op(1,rek2));
```

$$\mathit{REk2} := (2\,k+2-n)\,(-n+2\,k+1)\,(k+1)\,\mathit{Sk} - 2\,(-n+k)^2\,(2\,k+1)$$

```
>  REn2:=subs({S(n+1)=Sn,S(n)=1},op(1,ren2));
```

$$\mathit{REn2} := (-n-1+k)^2\,\mathit{Sn} - (n+1)\,(-n+2\,k)$$

What we are looking for are k-free operators contained in the left ideals generated by the operator systems $\{REk1, REn1\}$ and $\{REk2, REn2\}$. For this purpose we choose a term order eliminating the variable k, namely the lexicographical term order $\prec$ with $Sk \prec Sn \prec k$.

```
>  T:=MonomialOrder(A,plex(k,Sn,Sk));
```

$$T := \mathit{monomial\_order}$$

We compute a Gröbner basis w.r.t. $\prec$ of the ideal generated by $REk1$ and $REn1$

```
>  GB1:=Basis([REn1,REk1],T):
```

and we extract all k-free oprators

```
>  L1:=remove(has,GB1,k):
```

For the summand $\binom{n}{k}^3$ we get a single k-free operator.

```
>  nops(L1);
```

$$1$$

Substituting Sk by 1 we get a third order shift operator.

```
>  RE1:=collect(subs(Sk=1,L1[1]),Sn,factor);
```

$$\mathit{RE1} := (3\,n+4)\,(n+3)^2\,\mathit{Sn}^3 + (-148-18\,n^3-232\,n-114\,n^2)\,\mathit{Sn}^2$$
$$-\,(3\,n+5)\,(15\,n^2+55\,n+48)\,\mathit{Sn} - 8\,(7+3\,n)\,(n+1)^2$$

We apply the same steps to the summand $\binom{n}{k}^2\binom{2k}{n}$.

```
>  GB2:=Basis([REn2,REk2],T):
>  L2:=remove(has,GB2,k):
>  nops(L2);
```

$$2$$

```
>  RE2:=collect(subs(Sk=1,L2[1]),Sn,factor);
```

$$RE2 := -3\,(n+1)\,(n+3)^2\,Sn^3 + 3\,(n+1)\,(44+7\,n^2+35\,n)\,Sn^2 + 24\,(n+2)^2\,(n+1)\,Sn$$

```
>  ct2:=collect(content(RE2,Sn),Sn,factor);
```

$$ct2 := 3+3\,n$$

```
>  RE2:=collect(RE2/ct2,Sn,factor);
```

$$RE2 := -(n+3)^2\,Sn^3 + (44+7\,n^2+35\,n)\,Sn^2 + 8\,Sn\,(n+2)^2$$

```
>  RE3:=collect(subs(Sk=1,L2[2]),Sn,factor);
```

$$RE3 := (4\,n+9)\,(n+4)^2\,Sn^4 + (-725\,n-231\,n^2-738-24\,n^3)\,Sn^3 - 4\,(n+2)\,(15\,n^2+78\,n+103)\,Sn^2 - 32\,(n+2)^2\,(n+1)\,Sn$$

We shift back the two shift operators to get

```
>  RE2:=sort(subs({n=n-1,Sn^3=Sn^2,Sn^2=Sn,Sn=1},RE2));
```

$$RE2 := -(n+2)^2\,Sn^2 + (35\,n+9+7\,(n-1)^2)\,Sn + 8\,(n+1)^2$$

```
>  RE3:=subs({n=n-1,Sn^4=Sn^3,Sn^3=Sn^2,Sn^2=Sn,Sn=1},RE3);
```

$$RE3 := (5+4\,n)\,(n+3)^2\,Sn^3 + (-725\,n-13-231\,(n-1)^2-24\,(n-1)^3)\,Sn^2 - 4\,(n+1)\,(15\,(n-1)^2+78\,n+25)\,Sn - 32\,(n+1)^2\,n$$

a second order and a third order shift operators $RE2$ and $RE3$ annihilating the summand $\binom{n}{k}^2\binom{2k}{n}$. The shift operator $RE3$ is of order three, but it is not equal to the operator $RE1$ annihilating $\binom{n}{k}^3$. However, $RE1$ and $RE3$ may have a common right factor which can be determined (if it exists) by performing $\gcd$ computation. In fact, this can be done by using the command `skew_gcd(p,q,x,A)` which returns a list $\{g, a, b, u, v\}$ such that $up + vq = 0$ and $ap + bq = g$, where g is a right gcd of p and q.

```
>  gcdL1:=collect(skew_gcdex(RE1,RE3,Sn,A),Sn,factor);
```

$$gcdL1 := [-(27\,n+35)\,(n+2)^2\,Sn^2 + (27\,n+35)\,(7\,n^2+21\,n+16)\,Sn + 8\,(27\,n+35)\,(n+1)^2,\ -5-4\,n,\ 3\,n+4,\ -(4\,n+9)\,(27\,n+35)\,Sn - 4\,n\,(27\,n+62),\ (7+3\,n)\,(27\,n+35)\,Sn + (7+3\,n)\,(27\,n+62)]$$

The first element of the list $gcdL1$ is the common right factor we are looking for which leads, after removing the content, to the following second order annihilating shift operator for both summands:

```
>  RE:=collect(gcdL1[1]/content(gcdL1[1],Sn),Sn,factor);
```

$$RE := -(n+2)^2\,Sn^2 + (7\,n^2+21\,n+16)\,Sn + 8\,(n+1)^2$$

Since the operator RE is equal to $RE2$, then the computation of a gcd of $RE1$ and $RE2$

```
>  gcdL2:=collect(skew_gcdex(RE1,RE2,Sn,A),Sn,factor);
```

$$gcdL2 := [-(n+2)^2\,Sn^2 + (7\,n^2+21\,n+16)\,Sn + 8\,(n+1)^2,\ 0,\ 1,\ 1,\ (3\,n+4)\,Sn + 7 + 3\,n]$$

leads indirectly to the following factorization of $RE1$:

```
>   F:=gcdL2[5]*gcdL2[1];
```

$$F := ((3\,n+4)\,Sn+7+3\,n)$$
$$(-(n+2)^2\,Sn^2+(7\,n^2+21\,n+16)\,Sn+8\,(n+1)^2)$$

```
>   expand(skew_product(gcdL2[5],gcdL2[1],A)+RE1);
```

$$0$$

Hence, the sums $\sum\limits_{k=0}^{n}\binom{n}{k}^3$ and $\sum\limits_{k=0}^{n}\binom{n}{k}^2\binom{2k}{n}$ are annihilated by a common second order shift operator RE and satisfy the same initial values $s(0)=1$ and $s(1)=2$ which proves the identity.
We note that we can use the Maple package `hsum15`, which contains an implementation of Zeilberger's fast algorithm relying essentially on linear algebra techniques, to compute recurrence equations for both sums.

```
>   linkssummand:=binomial(n,k)^3;
```

$$\mathit{linkssummand} := \text{binomial}(n,\,k)^3$$

```
>   sumrecursion(linkssummand,k,S(n));
```

$$-(n+2)^2\,\mathrm{S}(n+2)+(7\,n^2+21\,n+16)\,\mathrm{S}(n+1)+8\,(n+1)^2\,\mathrm{S}(n)=0$$

```
>   rechtssummand:=binomial(n,k)^2*binomial(2*k,n);
```

$$\mathit{rechtssummand} := \text{binomial}(n,\,k)^2\,\text{binomial}(2\,k,\,n)$$

```
>   sumrecursion(rechtssummand,k,S(n));
```

$$-(n+2)^2\,\mathrm{S}(n+2)+(7\,n^2+21\,n+16)\,\mathrm{S}(n+1)+8\,(n+1)^2\,\mathrm{S}(n)=0$$

Remark 5.3

- Using other term orders, such as *degree lexicographical* order and *weighted degree* order, the Gröbner basis computations did not succeed to yield to a lower order shift operator $RE1$ annihilating the sum $\sum\limits_{k=0}^{n}\binom{n}{k}^3$.
- Using the command `nc_factorize` of the **REDUCE** package NCPOLY, we can factorize (if such a factorization exists!) operators like $RE1$ in order to get lower order *right factor*. The factorization of $RE1$ leads to

$$-\big((3n+4)S_n+(7+3n)\big)\big((n+2)^2S_n^2+(-7n^2-21n-16)S_n-8(1+n)^2\big), \quad (5.15)$$

  which confirms the factorization F obtained in the previous Maple session.

Example 5.4
We consider the sum

$$s(n,x)=\sum_{k=0}^{n}\binom{n}{k}L_k^a(x), \quad (5.16)$$

where $L_k^a(x)$ are the Laguerre orthogonal polynomials. The summand $\binom{n}{k}L_n^a(x)$ is annihilated by the following operator system

$$\begin{aligned} REk &= (1+k)(2+k)^2 S_k^2 + (1+k)(1+k-n)(3+a+2k-x)S_k \qquad (5.17)\\ &+ (1+a+k)(k-n)(1+k-n),\\ REn &= (n-k+1)S_n - (1+n), \qquad (5.18)\end{aligned}$$

which generates a left ideal in

$$\mathbb{O}_p = \mathbb{Q}(a,n,x)[S_k; S_k, 0][S_n; S_n, 0]. \qquad (5.19)$$

In the following Maple session, we will show how to get a recurrence equation of the sum $s(n,x)$ of lowest order (if possible!).

Maple Session 5.2 (recurrence equation for $s(n,x) = \sum_{k=0}^{n} \binom{n}{k} L_k^a(x)$)

We begin by loading the packages we need

```
>  read "hsum15.mpl";
```

$$Package\ \text{“}Hypergeometric\ Summation\text{“},\ Maple\ V\ -\ Maple\ 15$$
$$Copyright\ 1998-2012,\ Wolfram\ Koepf,\ University\ of\ Kassel$$

```
>  with(Ore_algebra):
>  with(Groebner):
```

and define the Ore algebra where the Gröbner basis computation will be performed

```
>  A:=skew_algebra(shift=[Sn,n],shift=[Sk,k],comm=[a,x],polynom=k);
```

$$A := Ore\_algebra$$

We determine a recurrence equation w.r.t. k of the summand $\binom{n}{k} L_k^a(x)$. We note that the Laguerre polynomials $L_k^a(x)$ can be expressed as a sum

$$L_k^a(x) = \sum_{j=0}^{k} \binom{k+a}{k-j} \frac{(-x)^j}{j!}. \qquad (5.20)$$

Hence, the summand $\binom{n}{k} L_k^a(x)$ is equal to

$$\sum_{j=0}^{k} \binom{n}{k} \binom{k+a}{k-j} \frac{(-x)^j}{j!}. \qquad (5.21)$$

We apply to the summand of (5.21) the command `sumrecursion` of the package `hsum15` to get a recurrence equation rek w.r.t k satisfied by $\binom{n}{k} L_k^a(x)$.

```
>  laguerresummand:=proc(n,alpha,x,k) binomial(n+alpha,n-k)/k!*(-x)^k
>  end proc:
>  summand:=binomial(n,k)*laguerresummand(k,a,x,j);
```

$$summand := \frac{\text{binomial}(n,\ k)\,\text{binomial}(k+a,\ k-j)\,(-x)^j}{j!}$$

```
>  rek:=sumrecursion(summand,j,S(k));
```

$$\begin{aligned} rek := & (k+1)\,(k+2)^2\,\mathrm{S}(k+2) \\ & + (k+1)\,(-n+k+1)\,(-x+2\,k+a+3)\,\mathrm{S}(k+1) \\ & + (-n+k+1)\,(k+a+1)\,(-n+k)\,\mathrm{S}(k) = 0 \end{aligned}$$

To get a recurrence equation ren w.r.t. n satisfied by $\binom{n}{k}L_k^a(x)$, we use the following procedure:

```
> HolonomicRE:=proc(term,sk)
> local s,k,r;
> s:=op(0,sk):  k:=op(1,sk):
> r:=ratio(term,k);
> denom(r)*s(k+1)-numer(r)*s(k)=0;
> end proc:
```

```
> ren:=HolonomicRE(summand,S(n));
```

$$ren := (-n-1+k)\,\mathrm{S}(n+1) - (-n-1)\,\mathrm{S}(n) = 0$$

We express rek and ren in operator notation.

```
> REk:=subs({S(k+2)=Sk^2,S(k+1)=Sk,S(k)=1},op(1,rek));
```

$$\begin{aligned} REk := & (k+1)\,(k+2)^2\,Sk^2 + (k+1)\,(-n+k+1)\,(-x+2\,k+a+3)\,Sk \\ & + (-n+k+1)\,(k+a+1)\,(-n+k) \end{aligned}$$

```
> REn:=subs({S(n+1)=Sn,S(n)=1},op(1,ren));
```

$$REn := (-n-1+k)\,Sn + n + 1$$

To get a k-free operator we choose the lexicographical term order $\prec$ with $Sk \prec Sn \prec k$.

```
> T:= MonomialOrder(A,plex(k,Sn,Sk));
```

$$T := monomial\_order$$

We compute a Gröbner basis w.r.t. $\prec$ of the ideal generated by REk and REn.

```
> GB:=Basis([REk,REn],T):
```

Extracting the k-free operator and substituting S_k by 1 we get

```
> RE:=collect(subs(Sk=1,op(remove(has,GB,k))),Sn,factor);
```

$$\begin{aligned} RE := & (3+n)\,Sn^3 + (-12-5\,n-a+x)\,Sn^2 + (2\,a+14-x+8\,n)\,Sn - 4 \\ & - 4\,n \end{aligned}$$

which is a third order recurrence equation satisfied by $s(n,x)$.

Using the **REDUCE** package NCPOLY we do not get lower order right factors of RE.

Remark 5.5
The recurrence equation satisfied by $\sum\limits_{k=0}^{n}\binom{n}{k}L_k^a(x)$ can be obtained in a completely different way using the Maple package `multsum11` which was implemented by my colleague *Torsten Sprenger* in relation with his master thesis [Spr04]. This implementation uses a variant of Fasenmyer algorithm [Fas49] which essentially relies on linear algebra techniques to compute recurrence equations for multiple sums.

Example 5.6
In this example, we will apply Zeilberger's slow algorithm to an integral (see [AS64], 11.4.28).

We consider the integral

$$I(n) = \int_0^\infty e^{-a^2x^2} x^m J_n(bx)dx, \tag{5.22}$$

$J_n(x)$ denoting the Bessel function of first kind. The integrand $e^{-a^2x^2}x^m J_n(bx)$ is annihilated by the following operator system

$$\begin{aligned} DE &= x^2D_x^2 + x(3-2m+4a^2x^2)D_x \qquad (5.23)\\ &+ (1-2m+m^2-n^2+8a^2x^2+b^2x^2-4a^2mx^2+4a^4x^4), \\ RE &= bxS_n^2 - 2(1+n)S_n + bx, \qquad (5.24) \end{aligned}$$

which generates a left ideal in

$$\mathbb{O}_p = \mathbb{Q}(a,b,m,n)[x][D_x;1,Dx][S_n;S_n,0]. \tag{5.25}$$

Maple Session 5.3 (Recurrence equation for $I(n) = \int_0^\infty e^{-a^2x^2}x^m J_n(bx)dx$)

We load the needed packages

```
> with(Ore_algebra):
> with(Groebner):
```

and define the Ore algebra where the Gröbner basis computation will be performed

```
> A:=skew_algebra(shift=[Sn,n],diff=[Dx,x],comm=[m,a,b],polynom=x);
```

$$A := Ore\_algebra$$

The integrand $e^{-a^2x^2}x^m J_n(bx)dx$ is annihilated by the following operators:

```
> DE:=x^2*Dx^2+x*(3-2*m+4*a^2*x^2)*Dx
> +(1-2*m+m^2-n^2+8*a^2*x^2+b^2*x^2-4*a^2*m*x^2+4*a^4*x^4);
```

$$DE := x^2\,Dx^2 + x\,(3-2\,m+4\,a^2\,x^2)\,Dx + 1 - 2\,m + m^2 - n^2 + 8\,a^2\,x^2$$
$$+ b^2\,x^2 - 4\,a^2\,m\,x^2 + 4\,a^4\,x^4$$

```
> RE:=b*x*Sn^2-2*(n+1)*Sn+b*x;
```

$$RE := b\,x\,Sn^2 - 2\,(n+1)\,Sn + b\,x$$

```
> T:=MonomialOrder(A,wdeg([3,1,0],[x,Sn,Dx]));
```

$$T := monomial\_order$$

To get an x-free operator we choose a weighted lexicographical term order $\prec_w$ with $w = (3,1,0)$ for (x, S_n, D_x).

```
> T:=MonomialOrder(A,wdeg([3,1,0],[x,Sn,Dx]));
```

$$T := monomial\_order$$

We compute a Gröbner basis of the ideal generated by DE and RE.

```
> GB:=Basis([DE,RE],T):
```

Extracting the x-free operator and substituting D_x by 0 we get

```
> REn:=subs(Dx=0,collect(op(remove(has,GB,x)),Sn,factor));
```

$$REn := b^2\,(n+1)\,(-4-n+m)\,Sn^4$$
$$-\,2\,(n+2)\,(4\,a^2\,n^2 + 16\,a^2\,n - b^2\,m + b^2 + 12\,a^2)\,Sn^2$$
$$+\,b^2\,(3+n)\,(n+m)$$

which is a recurrence equation of order 4 satisfied by $I(n)$.

Using NCPOLY we do not find a lower order right factor of REn.

5.1.2 Takayama's Algorithm

Gert Almkvist and Doron Zeilberger [AZ90] were the first who observed that in the case of natural boundaries (see Definition 5.1), where the inhomogeneous part R in equations (5.3) and (5.5) vanishes, the complete elimination of the variable t to perform creative telescoping is more restrictive than necessary. In this context only the operator P must be free of t and hence commutes with the operator ∂_t. Mainly, it doesn't matter if the operator Q contains t for the reason that only the operator P must be computed to perform creative telescoping.
Relying on the fact that it is useless to compute the operator Q, the strategy followed by Takayama's algorithm is to reduce modulo the right ideal $\partial_t\mathbb{O}_r$ before performing the elimination of the variable t. This means that all computations will take place in

$$I_t = I/\partial_t\mathbb{O}_r$$

where I is the annihilating ideal of a given holonomic function $f(t, \boldsymbol{x})$. In I_t the action of the operator ∂_t is to multiply by 0. In practice, to perform the reduction modulo $\partial_t\mathbb{O}_r$ we should simply apply the following rewrite rules:

- In the differential setting:

$$\begin{aligned} t^k D_t^p &= D_t^p t^k + (-1)^p (t^k)^{(p)} \\ &\equiv (-1)^p (t^k)^{(p)} \quad \text{modulo} \quad D_t\mathbb{O}_r. \end{aligned} \tag{5.26}$$

- In the difference setting:

$$\begin{aligned} n^k S_n^p &= (S_n^p - 1)(n-p)^k + (n-p)^k \\ &= (S_n - 1)\sum_{i=0}^{p} \binom{p}{i} S_n^i \,(n-p)^k + (n-p)^k \\ &\equiv (n-p)^k \quad \text{modulo} \quad \Delta_n\mathbb{O}_r. \end{aligned} \tag{5.27}$$

These rewrite rules can be recursively proved relying on the commutator rules defining the Ore algebra $\mathbb{O}_r$, namely

$$\begin{aligned} tD_t &= D_t t - 1, \\ nS_n &= S_n n - S_n. \end{aligned}$$

Performing reduction before elimination has at least two consequences:

- Decrease the size of the data.

- Get a bigger annihilating ideal, i.e. shorter operators of lower orders.

However, following this strategy may lead to a technical complication which is explained by the following example:

Example 5.7
Consider an operator in $\mathbb{O}_r$ which is written in the form

$$L + D_t Q,$$

where D_t denotes the operator of partial derivation w.r.t. to t.
Multiplying by t we get

$$tL + tD_t Q.$$

Since $tD_t = D_t t - 1$ this leads to

$$\begin{aligned} & tL + (D_t t - 1)Q \\ = \; & tL + D_t(tQ) - Q \\ = \; & tL - Q \quad \text{modulo} \quad D_t\mathbb{O}_r. \end{aligned}$$

Hence, we get an additional term $-Q$ which we lose if we first reduce modulo $D_t\mathbb{O}_r$ and then multiply by the variable t. We need thereby an algorithm which ensures that the multiplication by t or powers of it takes place before the reduction modulo $D_t\mathbb{O}_r$.

An algorithm to solve this problem in the context of Weyl algebras was given by Nobuki Takayama [Tak90a, Tak90b]. This algorithm relies on a generalization to the noncommutative case of procedures for module-Gröbner basis computation over a polynomial ring. F. Chyzak and B. Salvy [CS96] extended this algorithm to the more general case of Ore algebras. This extension is supposed to treat with modules and submodules and for this reason let's first focus on where and how we may consider module structures in our context.
The operator (5.1) we are looking for lies in

$$M = I + \partial_t \mathbb{O}_r \subseteq \mathbb{O}_r \tag{5.28}$$

which has no ideal structure since it is the sum of a left ideal I and a right ideal $\partial_t\mathbb{O}_r$. However, M may be considered as an $\mathbb{O}_m$-module, where

$$\mathbb{O}_m = \mathbb{K}(\boldsymbol{x})[\boldsymbol{\partial_x}; \boldsymbol{\sigma_x}, \boldsymbol{\delta_x}][\partial_t; \sigma_t, \delta_t]. \tag{5.29}$$

The module M contains all products of the form:

$$t^i G_j, \quad i = 0, 1, \dots,$$

where G_j are generators of I. Hence, it is clear that M is of infinite type since the cardinality of the powers t^i is infinite.

The ideal $I_t = I/\partial_t \mathbb{O}_r$ inherits from M, by reduction by $\partial_t \mathbb{O}_r$, an $\mathbb{O}$-module structure, where $\mathbb{O}$ denotes

$$\mathbb{O} = \mathbb{K}(\boldsymbol{x})[\boldsymbol{\partial_x}; \boldsymbol{\sigma_x}, \boldsymbol{\delta_x}]. \tag{5.30}$$

In I_t considered as $\mathbb{O}$-module, every element G_j may be written as

$$G_i = \sum_{i=0}^{s} t^i \, G_{ij}, \qquad G_{ij} \in \mathbb{O}, \tag{5.31}$$

which we denote as an infinite vector

$$G_j = \begin{pmatrix} G_{0j} \\ G_{1j} \\ \vdots \\ G_{sj} \\ 0 \\ \vdots \end{pmatrix}.$$

What we are looking for is a linear combination of the form

$$\sum_{j=1}^{m} P_j(\boldsymbol{x}, \boldsymbol{\partial_x}) \, G_j = \begin{pmatrix} P(\boldsymbol{x}, \boldsymbol{\partial_x}) \\ 0 \\ \vdots \end{pmatrix}, \tag{5.32}$$

where $P_j(\boldsymbol{x}, \boldsymbol{\partial_x})$ are operators in $\mathbb{O}$ and $P(\boldsymbol{x}, \boldsymbol{\partial_x})$ is the sought annihilating operator in (5.7).

Example 5.8
We reconsider the sequence $\mathcal{U}_{n,k} = \binom{n}{k}$ which is annihilated by the ideal I generated by the operators:

$$\begin{aligned} P &= (n+1-k)S_n - (n+1) \\ &= ((n+1)S_n - (n+1))k^0 - S_n k \\ &= \begin{pmatrix} (n+1)S_n - (n+1) \\ -S_n \\ 0 \\ \vdots \end{pmatrix}, \end{aligned}$$

and

$$\begin{aligned} Q &= (k+1)S_k - (n-k) \\ &= (S_k - 1)k + 2k - n \\ &= -nk^0 + 2k \quad \text{modulo} \quad \triangle_k = (S_k - 1) \\ &= \begin{pmatrix} -n \\ 2 \\ 0 \\ \vdots \end{pmatrix}. \end{aligned}$$

The following linear combination

$$2 \begin{pmatrix} (n+1)S_n - (n+1) \\ -S_n \\ 0 \\ \vdots \end{pmatrix} + S_n \begin{pmatrix} -n \\ 2 \\ 0 \\ \vdots \end{pmatrix} = \begin{pmatrix} (n+1)S_n - 2(n+1) \\ 0 \\ \vdots \end{pmatrix}$$

gives the k-free annihilating operator $P(n, S_n) = (n+1)S_n - 2(n+1)$ of the sequence $\mathcal{U}_{n,k}$.

To get the operator $P(\boldsymbol{x}, \boldsymbol{\partial_x})$, we should perform elimination by computing a Gröbner basis in M with respect to an order $\prec_m$ that gives priority to the position in the vector than the term order on $\mathbb{O}_m$ (POT order: "position over term") see Remark 5.9. The question now: how to perform this elimination in the infinitely generated module M? To overcome this technical problem, the idea of Takayama is to proceed by *module-truncation* with respect to the degree of t, that means in each loop only operators in M up to a fixed maximal total degree d in t are considered. These operators generate an M-submodule of finite type.
The extension of Takayama's idea to our context consists in each loop of three steps:

- Multiply the generators of I by powers of t to create as much operators as possible with maximal total degree d in t, which ensures that multiplication by t or powers of it takes place before reduction (see Example (5.7)).
- Perform reduction using the rules (5.26) and (5.27).
- Perform a module Gröbner basis computation with respect to a POT term order $\prec_m$ elimination t.

If no operator P is found, more multiples by powers of t should be included and the bound d must be increased. If the ideal I is holonomic then the termination of the algorithm is ensured by the elimination property see Proposition 4.38.

Remark 5.9 (Module term order)
Let R be a ring, $\boldsymbol{e_1} = (1, 0, \ldots, 0), \ldots, \boldsymbol{e_m} = (0, 0, \ldots, 1)$ a standard basis of R^m and $\prec$ a monomial ordering on R. For monomials $\boldsymbol{X} = X\boldsymbol{e_i}$ and $\boldsymbol{Y} = Y\boldsymbol{e_j}$ of R^m, We say that $\prec_m$ is a **POT** (position over term) term order on R^m if

$$\boldsymbol{X} \prec_m \boldsymbol{Y} \Leftrightarrow i < j \quad \text{or} \quad i = j \text{ and } X \prec Y,$$

and we say that $\prec_m$ is **TOP** (term over position) term order on R^m if

$$\boldsymbol{X} \prec_m \boldsymbol{Y} \Leftrightarrow X < Y \quad \text{or} \quad X = Y \text{ and } i < j.$$

Algorithm 5.1: **Extended Takayama's algorithm**

```
Input  : ${G_1, \dots, G_s}$ a set of polynomials in
         $\mathbb{O}_p = \mathbb{K}(\boldsymbol{x})[t][\boldsymbol{\partial_x}; \boldsymbol{\sigma_x}, \boldsymbol{\delta_x}][\partial_t; \sigma_t, \delta_t]$ ;
Output: A telescoping operator $P(\boldsymbol{x}, \boldsymbol{\partial_x})$ satisfying (5.7);
begin
    Initialization: $G := \{\}, d := \max_{1 \le i \le s} \deg_t G_i$;
    while $G := \{\}$ do
        $H := \{t^\alpha G_i \mid \ |\alpha| \le d - \deg_t G_i\}$;
        $Hred := \{H_i \mod \partial_t \mid \ H_i \in H\}$;
        $G :=$ module Gröbner basis of $Hred$ w. r. t. $\prec_m$ eliminating $t$;
        $G := \{P_i \mid \ \deg_t P_i = 0\}$;
        $d = d + 1$;
    end
    return $G$;
end
```

Example 5.10
In this example, we consider the sum

$$A_n := \sum_{k=0}^{n} \binom{n}{k}^2 \binom{n+k}{k}^2 \tag{5.33}$$

describing the *Apéry Numbers*. Our aim is to determine a recurrence equation satisfied by A_n using the extended Takayama's algorithm. This recurrence equation played an essential role in Apéry proof [vdP78] of the irrationality of

$$\zeta(3) = \sum_{k=1}^{\infty} \frac{1}{k^3}. \tag{5.34}$$

We note that the sum A_n is of natural boundaries since it is of finite support and thus Takayama's algorithm can be applied.

Maple Session 5.4 (Takayama's algorithm: recurrence equation for $A_n := \sum_{k=0}^{n} \binom{n}{k}^2 \binom{n+k}{k}^2$)

We load the needed packages

```
>  with(Ore_algebra):
>  with(Groebner):
>  read "hsum15.mpl";
```

Package "Hypergeometric Summation", Maple V − Maple 15
Copyright 1998 − 2012, Wolfram Koepf, University of Kassel

and define the following Ore algebra

```
>  A:=shift_algebra([Sn,n],[Sk,k]);
```

$$A := Ore\_algebra$$

Using the following procedure

```
>  HolonomicRE:=proc(term,sk)
>  local s,k,r;
>  s:=op(0,sk):  k:=op(1,sk):
>  r:=ratio(term,k);
>  denom(r)*s(k+1)-numer(r)*s(k)=0;
>  end proc:
```

we determine an annihilating operator system of the summand $\binom{n}{k}^2\binom{n+k}{k}^2$

```
>  p:=HolonomicRE(binomial(n,k)^2*binomial(n+k,k)^2,S(n));
```

$$p := (-n-1+k)^2\,\mathrm{S}(n+1) - (n+1+k)^2\,\mathrm{S}(n) = 0$$

```
>  q:=HolonomicRE(binomial(n,k)^2*binomial(n+k,k)^2,S(k));
```

$$q := (k+1)^4\,\mathrm{S}(k+1) - (n+1+k)^2\,(-n+k)^2\,\mathrm{S}(k) = 0$$

Written in operator notation:

```
>  P:=subs({S(n+1)=Sn,S(n)=1},op(1,p));
```

$$P := (-n-1+k)^2\,Sn - (n+1+k)^2$$

```
>  Q:=subs({S(k+1)=Sk,S(k)=1},op(1,q));
```

$$Q := (k+1)^4\,Sk - (n+1+k)^2\,(-n+k)^2$$

The operators P and Q considered as polynomials in k leads to the following data:

```
>  degP_k:=degree(P,k);
```

$$degP\_k := 2$$

```
>  degQ_k:=degree(Q,k);
```

$$degQ\_k := 4$$

```
>  d:=max(degP_k,degQ_k);
```

$$d := 4$$

After one execution of the **while**-loop of Takayama's algorithm, we do not get a module Gröbner basis containing a k-free polynomial. Therefore, we increase d by 1.

```
>  d:=d+N;
```

$$d := 4 + N$$

```
>  N:=1:
```

$$N := 1$$

We left multiply the operators P and Q by powers of k to get the sets:

```
>  H_P:=[seq(skew_product(k^i,P,A),i=0..d-degP_k)];
```

$$H\_P := [$$
$$-2\,n\,k - 1 - n^2 - 2\,n - 2\,k - k^2 + (n^2 + 2\,n - 2\,n\,k + 1 - 2\,k + k^2)\,Sn$$
$$, -2\,n\,k^2 - k - k\,n^2 - 2\,n\,k - 2\,k^2 - k^3$$
$$+ (k\,n^2 + 2\,n\,k - 2\,k^2 + k^3 - 2\,n\,k^2 + k)\,Sn, -2\,k^3\,n - k^2 - k^2\,n^2$$
$$- 2\,n\,k^2 - 2\,k^3 - k^4 + (k^2\,n^2 + 2\,n\,k^2 - 2\,k^3 + k^4 - 2\,k^3\,n + k^2)\,Sn,$$
$$-2\,k^4\,n - k^3 - k^3\,n^2 - 2\,k^3\,n - 2\,k^4 - k^5$$
$$+ (k^3\,n^2 + 2\,k^3\,n - 2\,k^4 + k^5 - 2\,k^4\,n + k^3)\,Sn]$$

```
>   H_Q:=[seq(reverse_polynomial(skew_product(k^i,Q,A),A,{Sk
>   }),i=0..d-degQ_k)];
```

$$H\_Q := [-n^4 - n^2 - 2\,n^3 + (2\,n^2 + 2\,n)\,k + (-1 + 2\,n + 2\,n^2)\,k^2 - 2\,k^3$$
$$+ (-1 + Sk)\,k^4, (-n^4 - n^2 - 2\,n^3)\,k + (2\,n^2 + 2\,n)\,k^2$$
$$+ (-1 + 2\,n + 2\,n^2)\,k^3 + (-Sk - 2)\,k^4 + (-1 + Sk)\,k^5]$$

We note that we use the command `reverse_polynomial` to change the representation of the polynomials $k^i Q$ by moving the shift operator S_k to the left of monomials and thus to prepare them for reduction (see the rewriting rule (5.27)). Since the operators of `H_P` do not contain the shift operator S_k, we need only to reduce the elements of `H_Q` by $(Sk - 1)$ i.e. substitute S_k by 1.

```
>   H_Qred:=subs(Sk=1,H_Q);
```

$$H\_Qred := [-n^4 - n^2 - 2\,n^3 + (2\,n^2 + 2\,n)\,k + (-1 + 2\,n + 2\,n^2)\,k^2 - 2\,k^3,$$
$$(-n^4 - n^2 - 2\,n^3)\,k + (2\,n^2 + 2\,n)\,k^2 + (-1 + 2\,n + 2\,n^2)\,k^3 - 3\,k^4]$$

Finally the set $Hred$ is

```
>   Hred:=map(op,[H_P,H_Qred]);
```

$$Hred := [$$
$$-2\,n\,k - 1 - n^2 - 2\,n - 2\,k - k^2 + (n^2 + 2\,n - 2\,n\,k + 1 - 2\,k + k^2)\,Sn$$
$$, -2\,n\,k^2 - k - k\,n^2 - 2\,n\,k - 2\,k^2 - k^3$$
$$+ (k\,n^2 + 2\,n\,k - 2\,k^2 + k^3 - 2\,n\,k^2 + k)\,Sn, -2\,k^3\,n - k^2 - k^2\,n^2$$
$$- 2\,n\,k^2 - 2\,k^3 - k^4 + (k^2\,n^2 + 2\,n\,k^2 - 2\,k^3 + k^4 - 2\,k^3\,n + k^2)\,Sn,$$
$$-2\,k^4\,n - k^3 - k^3\,n^2 - 2\,k^3\,n - 2\,k^4 - k^5$$
$$+ (k^3\,n^2 + 2\,k^3\,n - 2\,k^4 + k^5 - 2\,k^4\,n + k^3)\,Sn,$$
$$-n^4 - n^2 - 2\,n^3 + (2\,n^2 + 2\,n)\,k + (-1 + 2\,n + 2\,n^2)\,k^2 - 2\,k^3,$$
$$(-n^4 - n^2 - 2\,n^3)\,k + (2\,n^2 + 2\,n)\,k^2 + (-1 + 2\,n + 2\,n^2)\,k^3 - 3\,k^4]$$

The module Göbner basis computation takes place in the following Ore algebra:

```
>   Ap:=shift_algebra([Sn,n],comm=k,polynom=k);
```

$$Ap := Ore\_algebra$$

and it is performed w.r.t. the following term order:

```
>   T:=MonomialOrder(Ap,lexdeg([k],[Sn]),[k]);
```

$$T := monomial\_order$$

We note that the third argument in the `MonomialOrder` call is very important to refrain from any multiplication by the variable k in the S-polynomials computations. The module Göbner basis computation gives

```
>   G:=Basis(Hred,T);
```

$$G := [-34\,Sn\,n^3 + Sn^2\,n^3 + n^3 - 153\,Sn\,n^2 + 6\,Sn^2\,n^2 + 3\,n^2 - 231\,Sn\,n + 12\,Sn^2\,n + 3\,n - 117\,Sn + 1 + 8\,Sn^2, -2\,Sn\,n - 2\,n - 2 - 3\,k + 3\,Sn\,k - 2\,Sn, -17\,n^3 - 39\,n^2 - 27\,n - 5 - 24\,k\,n^2 + Sn\,n^3 + 3\,Sn\,n + 3\,Sn\,n^2 + Sn + 12\,k^2 - 24\,n\,k, -100\,n^4 - 13\,n^2 - 91\,n^3 - 48\,k\,n^2 + 17\,n - Sn + 5\,Sn\,n^2 - Sn\,n - 96\,n^3\,k - 48\,k\,n^4 + 8\,Sn\,n^4 + 11\,Sn\,n^3 + 2\,n^5\,Sn - 34\,n^5 + 5 + 24\,k^3, -27\,n + Sn - 71\,n^2 + 7\,Sn\,n^2 + 3\,Sn\,n - 24\,k\,n^2 - 144\,n^3\,k + 72\,k^4 - 332\,n^4 - 165\,n^3 - 312\,k\,n^4 + 40\,Sn\,n^4 + 21\,Sn\,n^3 + 40\,n^5\,Sn + 20\,n^6\,Sn - 288\,n^5\,k + 4\,n^7\,Sn - 96\,n^6\,k - 416\,n^5 - 268\,n^6 - 68\,n^7 - 5, -197 - 288\,k - 1286\,n + 25\,Sn - 3692\,n^2 - 1584\,n\,k + 412\,Sn\,n^2 + 154\,Sn\,n - 3768\,k\,n^2 + 18\,k^5\,Sn - 5064\,n^3\,k - 18\,k^5 - 6437\,n^4 - 6116\,n^3 - 4200\,k\,n^4 + 601\,Sn\,n^4 + 628\,Sn\,n^3 + 374\,n^5\,Sn + 150\,n^6\,Sn - 2184\,n^5\,k + 36\,n^7\,Sn - 672\,n^6\,k - 4450\,n^5 - 1998\,n^6 - 540\,n^7 + 4\,n^8\,Sn - 96\,n^7\,k - 68\,n^8]$$

and leads to the following second order recurrence equation:

```
>   RE:=op(collect(remove(has,G,k),Sn,factor));
```

$$RE := (n+2)^3\,Sn^2 - (2\,n+3)\,(17\,n^2 + 51\,n + 39)\,Sn + (n+1)^3.$$

Example 5.11
We reconsider the sum

$$s(n) := \sum_{n=k}^{n} \binom{n}{k}^3. \tag{5.35}$$

We have seen in Example 5.2 that the application of "brutal" Gröbner basis elimination to the operator system annihilating the summand $\binom{n}{k}$ leads to a third order recurrence equation satisfied by $s(n)$. Even if we choose different term orders, we do not succeed to get a recurrence equation of lower order. In the following Maple session, we will show that Takayama's algorithm applied to3 $s(n)$ gives rise to a second order recurrence equation due to its reduction preprocessing step (Algorithm 5.1: lines 4 and 5).

Maple Session 5.5 (Takayama's algorithm: recurrence equation for $s(n) := \sum_{n=k}^{n} \binom{n}{k}^3$)

```
>   with(Ore_algebra):
>   with(Groebner):
>   A:=shift_algebra([Sn,n],[Sk,k]);
```

$$A := Ore\_algebra$$

The summand $\binom{n}{k}^3$ is annihilated by the following operator system:

```
>  P:= (-n-1+k)^3*Sn+(n+1)^3;
```

$$P := (-n-1+k)^3\,Sn + (n+1)^3,$$

```
>  Q:= (k+1)^3*Sk+(-n+k)^3;
```

$$Q := (k+1)^3\,Sk + (-n+k)^3.$$

P and Q considered as polynomials in k leads to the following data:

```
>  degP_k:=degree(P,k);
```

$$degP\_k := 3$$

```
>  degQ_k:=degree(Q,k);
```

$$degQ\_k := 3$$

```
>  d:=max(degP_k,degQ_k);
```

$$d := 3$$

```
>  d:=d+N:
```

After two iterations of the **while**-loop of Takayama's algorithm, we do not get a module Gröbner basis that contains a k-free polynomial. Therefore, we increase d by 2.

```
>  N:=2:
```

We multiply the operators P and Q by powers of k

```
>  H_P:=[seq(skew_product(k^i,P,A),i=0..d-degP_k)];
```

$$\begin{aligned}&H\_P := [n^3+3\,n^2+3\,n+1+\\&(-n^3-3\,n^2+3\,n^2\,k-3\,n+6\,n\,k-3\,n\,k^2-1+3\,k-3\,k^2+k^3)\,Sn, k\\&+k\,n^3+3\,n^2\,k+3\,n\,k+(-3\,n^2\,k-3\,n\,k^3+3\,n^2\,k^2-3\,n\,k+6\,n\,k^2\\&+k^4-k+3\,k^2-3\,k^3-k\,n^3)Sn, k^2+k^2\,n^3+3\,n^2\,k^2+3\,n\,k^2+(\\&-3\,n^2\,k^2-3\,k^4\,n+3\,k^3\,n^2-3\,n\,k^2+6\,n\,k^3+k^5-k^2+3\,k^3-3\,k^4\\&-k^2\,n^3)Sn]\end{aligned}$$

```
>  H_Q:=[seq(reverse_polynomial(skew_product(k^i,Q,A),A,{Sk
>  }),i=0..d-degQ_k)];
```

$$\begin{aligned}&H\_Q := [-n^3+3\,n^2\,k-3\,n\,k^2+(1+Sk)\,k^3,\\&-k\,n^3+3\,n^2\,k^2+(-3\,n-Sk)\,k^3+(1+Sk)\,k^4,\\&-k^2\,n^3+(3\,n^2+Sk)\,k^3+(-3\,n-2\,Sk)\,k^4+(1+Sk)\,k^5]\end{aligned}$$

and reduce by (S_k-1)

```
>  H_Qred:=subs(Sk=1,H_Q);
```

$$\begin{aligned}&H\_Qred := [-n^3+3\,n^2\,k-3\,n\,k^2+2\,k^3,\\&-k\,n^3+3\,n^2\,k^2+(-3\,n-1)\,k^3+2\,k^4,\\&-k^2\,n^3+(3\,n^2+1)\,k^3+(-3\,n-2)\,k^4+2\,k^5]\end{aligned}$$

to finally get

```
>  Hred:=map(op,[H_P,H_Qred]);
```

$$Hred := [n^3 + 3\,n^2 + 3\,n + 1 + (-n^3 - 3\,n^2 + 3\,n^2\,k - 3\,n + 6\,n\,k - 3\,n\,k^2 - 1 + 3\,k - 3\,k^2 + k^3)\,Sn, k + k\,n^3 + 3\,n^2\,k + 3\,n\,k + (-3\,n^2\,k - 3\,n\,k^3 + 3\,n^2\,k^2 - 3\,n\,k + 6\,n\,k^2 + k^4 - k + 3\,k^2 - 3\,k^3 - k\,n^3)Sn, k^2 + k^2\,n^3 + 3\,n^2\,k^2 + 3\,n\,k^2 + (-3\,n^2\,k^2 - 3\,k^4\,n + 3\,k^3\,n^2 - 3\,n\,k^2 + 6\,n\,k^3 + k^5 - k^2 + 3\,k^3 - 3\,k^4 - k^2\,n^3)Sn,\ -n^3 + 3\,n^2\,k - 3\,n\,k^2 + 2\,k^3, -k\,n^3 + 3\,n^2\,k^2 + (-3\,n - 1)\,k^3 + 2\,k^4, -k^2\,n^3 + (3\,n^2 + 1)\,k^3 + (-3\,n - 2)\,k^4 + 2\,k^5]$$

The module Gröbner basis computation takes place in the following Ore algebra

```
>  Ap:=shift_algebra([Sn,n],comm=k,polynom=k);
```

$$Ap := Ore\_algebra$$

and it is performed w.r.t. the following term order

```
>  T:=MonomialOrder(Ap,lexdeg([k],[Sn]),[k]);
```

$$T := monomial\_order$$

gives

```
>  G:=Basis(Hred,T);
```

$$G := [-7\,Sn\,n^2 - 8\,n^2 - 16\,n - 21\,Sn\,n - 16\,Sn - 8 + Sn^2\,n^2 + 4\,Sn^2\,n + 4\,Sn^2,\ -Sn\,n - n + 2\,k - Sn + 2\,Sn\,k, 11\,n^2 - Sn\,n^2 - 2\,Sn\,n - 12\,n\,k + 7\,n + 2 - Sn + 12\,k^2, 7\,n^3 - Sn\,n^3 - 2\,Sn\,n^2 + 7\,n^2 + 2\,n - Sn\,n + 8\,k^3, -n^4 + 14\,n^3 + 16\,k\,n^3 - Sn\,n^4 - 3\,Sn\,n^3 + 9\,n^2 - 3\,Sn\,n^2 + 2\,n - Sn\,n + 16\,k^4, -91\,n^5 + 22\,n^4 + 95\,n^3 + 30\,n^2 + 5\,Sn\,n^5 - 5\,Sn\,n^4 - 25\,Sn\,n^3 - 15\,Sn\,n^2 + 96\,n^4\,k + 96\,k\,n^3 + 96\,k^5]$$

and leads to the following second order recurrence equation

```
>  RE:=op(collect(remove(has,G,k),Sn,factor));
```

$$RE := (n + 2)^2\,Sn^2 + (-7\,n^2 - 21\,n - 16)\,Sn - 8\,(n + 1)^2$$

5.2 Chyzak's Algorithm

F. Chyzak in his paper [Chy00] presented algorithms which extend Gosper's algorithm and Zeilberger fast algorithm for *indefinite* and *definite* summations of hypergeometric terms to the more general case of ∂-finite functions (see Definition 4.1). In this part we distinguish two cases:

5.2.1 Indefinite case

For a better understanding of the idea behind Chyzak's algorithm in the indefinite case, we first briefly recall Gosper's algorithm. We consider the sum

$$s(n) = \sum_{k=0}^{n-1} f(k), \tag{5.36}$$

where $f(k)$ is a *hypergeometric* term that does not depend on n, that is the consecutive term ratio

$$r(k) = \frac{f(k+1)}{f(k)} \tag{5.37}$$

is a *rational function* of k. What is looked for is a function $g(k)$ satisfying

$$g(k+1) - g(k) = f(k) \tag{5.38}$$

The function $g(k)$ is called *antidifference* of $f(k)$. In this case, the sum $s(n)$ may be expressed in a simple form as follows

$$\begin{aligned} s(n) &= \sum_{k=0}^{n-1} f(k) \\ &= \sum_{k=0}^{n-1} \Big((g(k+1) - g(k) \Big) \\ &= g(n) - g(0) \\ &= g(n) + c. \end{aligned} \tag{5.39}$$

We say that a hypergeometric term $f(k)$ is *Gosper summable* if it has a hypergeometric term antidifference. Let $g(k)$ be a hypergeometric term, then the ratio

$$\frac{g(k)}{f(k)} = \frac{g(k)}{g(k+1) - g(k)} = \frac{1}{\frac{g(k+1)}{g(k)} - 1} \tag{5.40}$$

is clearly a rational function of k. So we get

$$g(k) = q(k)f(k), \tag{5.41}$$

where $q(k)$ is a rational function of k. Substituting $q(k)f(k)$ for $g(k)$ in the equation (5.38) reveals that $q(k)$ satisfies the equation

$$r(k)q(k+1) - q(k) = 1 \tag{5.42}$$

which is an inhomogeneous first order linear recurrence equation with rational coefficients. Thus the problem of finding a hypergeometric antidifference $g(k)$ of $f(k)$ is reduced to the problem of finding rational solutions of a first order linear recurrence equation. Gosper reduced the problem

further to that finding polynomial solutions of yet another recurrence. In this way, the existence and the computability of $g(k)$ depends on the solvability of such recurrence equations. For more details about Gosper's algorithm we refer to the books [PWZ96] and [Koe98, Koe06].
We will see now, how the equations (5.38) and (5.41) are the key points of Chyzak's algorithm that extends Gosper's to the more general case of ∂-finite functions. In operator notation the equation (5.38) is written as

$$\triangle_k \cdot g(k) = f(k), \tag{5.43}$$

where $\triangle_k = (S_k - 1)$. Applying $\triangle_k$ to (5.41) we get

$$\triangle_k \cdot g(k) = \triangle_k \cdot (q(k) \cdot f(k)) = (\triangle_k \cdot q(k)) \cdot f(k). \tag{5.44}$$

(5.43) together with (5.44) leads to

$$(\triangle_k \cdot q(k) - 1) \cdot f(k) = 0 \tag{5.45}$$

which means that the operator $(\triangle_k\, q(k) - 1)$ lies in $\mathrm{Ann}_{\mathbb{O}}(f)$, where $\mathbb{O} = \mathbb{K}(k)[\triangle_k; S_k, \triangle_k]$. Hence, the problem is to find the rational function $q(k) \in \mathbb{K}(k)$. This formulation of the problem by means of operators and ideals of operators opens the door to the use of Gröbner bases techniques and allows the generalization of Gosper's algorithm to the more general context of ∂-finite functions.
Let $f(\boldsymbol{x}, t)$ be a ∂-finite function with respect to the Ore algebra

$$\mathbb{O}_r = \mathbb{K}(\boldsymbol{x}, t)[\boldsymbol{\partial_x}; \boldsymbol{\sigma_x}, \boldsymbol{\delta_x}][\partial_t; \sigma_t, \delta_t],$$

where ∂_t denotes the difference operator $\triangle_t$ in the case of a discrete variable t and the differential operator D_t if t is continuous. The function f is described by its annihilating ideal $I = \mathrm{Ann}_{\mathbb{O}_r}(f)$ given by a Gröbner basis G. Now we want to find an operator $g = q \cdot f$ which lies in the left submodule $\mathbb{O}_r \cdot f$ of $\mathbb{O}_r$. Furthermore, we have seen in Section 2.2 that $\mathbb{O}_r \cdot f$ is isomorphic to the finite dimensional vector space $\mathbb{O}_r/I$ w. r. t. the homomorphism $q \mapsto$ reduced LNF$(q \mid G)$. As a consequence, the operator q can be written as

$$q = q_1 m_1 + q_2 m_2 + \cdots + q_s m_s, \tag{5.46}$$

where $\{m_i \mid 1 \leq i \leq s\}$ is a set of terms of $\mathbb{O}_r$ satisfying

$$m_i \notin \mathrm{L}(G), \tag{5.47}$$

and $\mathrm{L}(G)$ is the ideal generated by the leading terms of G. In this manner, the set of terms

$$\mathcal{M} := \{m_1, \ldots, m_s\}$$

is a *standard basis* of the $\mathbb{K}(\boldsymbol{x}, t)$-vector space $\mathbb{O}_r/I$. Hence, the problem for ∂-finite functions is to find

$$q_1, q_2, \ldots, q_s \in \mathbb{K}(\boldsymbol{x}, t) \tag{5.48}$$

such that

$$\partial_t \cdot (q_1 m_1 + q_2 m_2 + \cdots + q_s m_s) - 1 \in \langle G \rangle_{\mathbb{O}_r} = \text{Ann}_{\mathbb{O}}(f). \tag{5.49}$$

In fact, the condition (5.49) means that the left normal form of the operator

$$\mathcal{Q} := \partial_t \cdot (q_1 m_1 + q_2 m_2 + \cdots + q_s m_s) - 1 \tag{5.50}$$

(with unknown functions $q_i \in \mathbb{K}(\boldsymbol{x}, t)$) is equal to zero modulo G. Equating the coefficients of LNF$(\mathcal{Q} \mid G)$ to zero gives rise to a coupled system of first order linear difference (resp. differential) equations. If such a system admits rational solutions q_i then an antidifference (resp. antiderivative) is found, otherwise it does not exist.

Remark 5.12
For practical purposes and before equating the coefficients of LNF$(\mathcal{Q} \mid G)$ to zero, we should first write LNF$(\mathcal{Q} \mid G)$ in a standard representation form, that is with all Ore operators on the right hand side of the coefficients. This can be obtained using the following commutator rules:

$$\begin{aligned} \triangle_t \cdot q_i(\boldsymbol{x}, t) &= q_i(\boldsymbol{x}, t+1)\triangle_t + q_i(\boldsymbol{x}, t+1) - q_i(\boldsymbol{x}, t) \\ D_t \cdot q_i(\boldsymbol{x}, t) &= q_i(\boldsymbol{x}, t) D_t + \frac{\partial}{\partial t} q_i(\boldsymbol{x}, t). \end{aligned} \tag{5.51}$$

The question is now how to find, if they exist, the rational solutions q_i of the obtained system of first order linear difference (resp. differential) equations? This may be achieved using *direct methods* as proposed by Abramov and Barakatou [AB98, Bar99] or *indirect methods* by uncoupling the system using Gaussian elimination or special uncoupling algorithms like the algorithm of Abramov and Zima [AZ96] and then iteratively solve the difference (resp. differential) equations with Abramov's algorithm [Abr89, Abr95, ABP95].

Example 5.13
In this example, we use Chyzak's algorithm to compute an antiderivative of

$$\int H_n(x) dx, \tag{5.52}$$

where $H_n(x)$ denotes the Hermite orthogonal polynomials. We consider the Ore algebra

$$\mathbb{O}_r := \mathbb{Q}(n, x)[S_n; S_n, 0][D_x; 1, D_x].$$

An annihilating ideal I for $H_n(x)$ is generated by the following polynomials:

$$\begin{aligned} DE &= D_x^2 - 2x D_x + 2n, \\ RE &= S_n^2 - 2x S_n + 2(n+1). \end{aligned} \tag{5.53}$$

A Gröbner basis of I is given by

$$G = \{D_x + S_n - 2x, S_n^2 - 2x S_n + 2(n+1)\}. \tag{5.54}$$

Algorithm 5.2: **Chyzak's algorithm**

```
Input :
    • $\mathbb{O}_r = \mathbb{K}(\boldsymbol{x}, t)[\boldsymbol{\partial_x}; \boldsymbol{\sigma_x}, \boldsymbol{\delta_x}][\partial_t; \sigma_t, \delta_t]$;
    • $G$ a Gröbner basis of $I = \text{Ann}_{\mathbb{O}_r}(f)$ for a $\partial$-finite function $f$;
Output: Rational functions $q_i \in \mathbb{K}(\boldsymbol{x}, t)$ satisfying the condition (5.49);
begin
    1. $\mathcal{Q} := \partial_t(q_1 m_1 + q_2 m_2 + \cdots + q_s m_s) - 1$;
    2. compute LNF($\mathcal{Q} \mid G$);
    3. Write LNF($\mathcal{Q} \mid G$) in a standard representation form using (5.51);
    4. Equate the coefficients of LNF($\mathcal{Q} \mid G$) to zero. ;
    5. Find rational solutions of the corresponding system $\mathcal{E}$;
    if solving $\mathcal{E}$ is successful then
        return $q_1, \ldots, q_i$ ;
    else
        return $\{\}$;
    end
end
```

From G we deduce that the set

$$\mathcal{M} = \{1, S_n\}$$

is a standard basis of $\mathbb{O}_r/I$ as $\mathbb{Q}(n,x)$-vector space. We introduce the polynomial

$$\mathcal{Q} = D_x(q_1(n,x) + q_2(n,x)S_n) - 1.$$

If $\mathcal{Q} \in \langle G \rangle_{\mathbb{O}_r}$, then

$$(D_x(q_1(n,x) + q_2(n,x)S_n) - 1)H_n = 0. \tag{5.55}$$

Applying the integral sign we get

$$\int (D_x(q_1(n,x) + q_2(n,x)S_n) - 1)H_n dx = 0. \tag{5.56}$$

Finally, an antiderivative of H_n is given by

$$\int H_n dx = (q_1(n,x) + q_2(n,x)S_n)H_n. \tag{5.57}$$

In the following Maple Session, we show how to determine (if they exist) the rational functions q_1, q_2.

Maple Session 5.6 (Determination of the rational functions q_1 and q_2)

```
>  with(Ore_algebra):
>  with(Groebner):
```

We define the Ore algebra

```
>  A:=skew_algebra(shift=[Sn,n],diff=[Dx,x],func=[q1,q2,b]);
```

$$A := \mathit{Ore\_algebra}$$

and compute G.

```
>  DE:= Dx^2-2*x*Dx+2*n;
```

$$\mathit{DE} := \mathit{Dx}^2 - 2\,x\,\mathit{Dx} + 2\,n,$$

```
>  RE:=Sn^2-2*x*Sn+2*n+2;
```

$$\mathit{RE} := \mathit{Sn}^2 - 2\,x\,\mathit{Sn} + 2\,n + 2.$$

```
>  T:=MonomialOrder(A,tdeg(Dx,Sn));
```

$$T := \mathit{monomial\_order}$$

```
>  G:=Basis([DE,RE],T);
```

$$G := [\mathit{Dx} + \mathit{Sn} - 2\,x,\ \mathit{Sn}^2 - 2\,x\,\mathit{Sn} + 2\,n + 2].$$

We introduce the polynomial Q

```
>  Q:=sort(skew_product(Dx,q1(x)+skew_product(q2(x),Sn,A),A)-1,
   {Dx,Sn});
```

$$Q := \mathrm{q2}(x)\,\mathit{Dx}\,\mathit{Sn} + \mathrm{q1}(x)\,\mathit{Dx} + (\tfrac{d}{dx}\,\mathrm{q2}(x))\,\mathit{Sn} + (\tfrac{d}{dx}\,\mathrm{q1}(x)) - 1.$$

In order to compute a left normal form of Q w.r.t. G we should make the following substitutions

```
>  Q:=subs({q1(x)=q1,q2(x)=q2,diff(q1(x),x)=q11,
>  diff(q2(x),x)=q21},Q);
```

$$Q := \mathit{q2}\ \mathit{Dx}\ \mathit{Sn} + \mathit{q1}\ \mathit{Dx} + \mathit{q21}\ \mathit{Sn} + \mathit{q11} - 1$$

and define the corresponding Ore algebra and term order

```
>  A_NF:=skew_algebra(shift=[Sn,n],diff=[Dx,x],
>  comm=[q1,q2,q11,q21]);
```

$$\mathit{A\_NF} := \mathit{Ore\_algebra},$$

```
>  T_NF:=MonomialOrder(A_NF,tdeg(Dx,Sn));
```

$$\mathit{T\_NF} := \mathit{monomial\_order}.$$

A left normal form of Q w.r.t. G is

```
>  LNF:=NormalForm(Q,G,T_NF);
```

$$\mathit{LNF} := \mathit{q11} - 1 + 2\,\mathit{q2}\,n + 2\,\mathit{q2} + 2\,\mathit{q1}\,x + (\mathit{q21} - \mathit{q1})\,\mathit{Sn}$$

Equating the coefficients of LNF to zero

```
>  sys_op:=[coeffs(LNF,Sn)];
```

$$\mathit{sys\_op} := [\mathit{q11} - 1 + 2\,\mathit{q2}\,n + 2\,\mathit{q2} + 2\,\mathit{q1}\,x,\ \mathit{q21} - \mathit{q1}]$$

we get the following linear system of differential equations

```
>  sys_fnc:=subs({q1=q1(x),q2=q2(x),q11=diff(q1(x),x),
>  q21=diff(q2(x),x)},sys_op);
```

$$\mathit{sys\_fnc} := [(\tfrac{d}{dx}\,\mathrm{q1}(x)) - 1 + 2\,\mathrm{q2}(x)\,n + 2\,\mathrm{q2}(x) + 2\,\mathrm{q1}(x)\,x,\ (\tfrac{d}{dx}\,\mathrm{q2}(x)) - \mathrm{q1}(x)]$$

in the variables

```
>  vars:=[q1(x),q2(x)];
```

$$vars := [\mathrm{q1}(x),\ \mathrm{q2}(x).]$$

To solve such a system we use the package

```
>  with(LinearFunctionalSystems):
```

which gives the following solution

```
>  Sol:=RationalSolution(sys_fnc,vars);
```

$$Sol := [0,\ \_c_1].$$

It is important to note that the constant c_1 is an element of $\mathbb{Q}(n)$. To determine c_1, we should only substitute Sol in one equation of sys_fnc and solve it to get

```
>  solve(eval(subs(q1(x)=0,sys_fnc[1]))=0,q2(x));
```

$$\frac{1}{2\,(n+1)}.$$

Hence, an antiderivative of the Hermite polynomials is given by

$$\int H_n dx = \frac{1}{2(n+1)} H_{n+1} \tag{5.58}$$

5.2.2 Definite Case

Chyzak's indefinite algorithm can be used for definite summation (resp. integration) in the same way as Zeilberger's famous fast algorithm makes use of Gosper's algorithm. We first briefly recall Zeilberger's fast algorithm for definite summation. We consider the sum

$$F(n) := \sum_{k\in\mathbb{Z}} f(n,k), \tag{5.59}$$

where $f(n,k)$ is hypergeometric in both variables. In general, we can not always expect that for a fixed n the summand $f(n,k)$ is Gosper summable. However, if $f(n,k)$ is *proper hypergeometric* (see [Koe98]) there always exists a linear combination of f and its n-shifts which is Gosper summable, that is

$$P(n,S_n) = p_0 + p_1(n)S_n + \cdots + p_l(n)S_n^l = \sum_{j=0}^{l} p_j S_n^j \tag{5.60}$$

satisfying

$$P(n,S_n)f(n,k) = g(n,k+1) - g(n,k), \tag{5.61}$$

where $p_i \in \mathbb{K}(n)$ and such that

$$g(n,k) = q(n,k)f(n,k) \tag{5.62}$$

for some rational function q. In operator notation this means that

$$(P(n,S_n) - \triangle_k \cdot q)\cdot f(n,k) = 0 \tag{5.63}$$

which leads to

$$P(n, S_n) + \triangle_k \cdot q \in \mathrm{Ann}_{\mathbb{O}}(f), \tag{5.64}$$

where $\mathbb{O} = \mathbb{K}(n, k)[\triangle_n; S_n, \triangle_n][\triangle_k; S_k, \triangle_k]$. Summing both sides of (5.61) over all integer values of k gives

$$\sum_{k\in\mathbb{Z}} P(n, S_n)f(n, k) = \sum_{k\in\mathbb{Z}} (g(n, k + 1) - g(n, k)). \tag{5.65}$$

Moreover, if for each n the function $g(n, k)$ has a finite support in k then we get

$$\sum_{k\in\mathbb{Z}} P(n, S_n)f(n, k) = P(n, S_n) \sum_{k\in\mathbb{Z}} f(n, k) = 0. \tag{5.66}$$

Thus, $P(n, S_n)$ is a homogeneous recurrence equation for the sum $F(n)$. Zeilberger's fast algorithm computes this recurrence equation by executing Gosper's algorithm to check the indefinite summability of

$$P(n, S_n)f(n, k) = \sum_{j=0}^{l} p_j S_n^j\, f(n, k), \quad p_j \in \mathbb{K}(n). \tag{5.67}$$

The algorithm starts with $f(n, k)$ which corresponds to the expression (5.67) for $l = 0$ and $p_0 = 1$. In each loop, the maximal total degree l of the n-shifts is increased until a result is obtained. We note that Zeilberger's fast algorithm relies on a parameterized version of Gosper's algorithm that additionally solves for the parameters p_i.
As in the indefinite case, Zeilberger's fast algorithm may be translated to the ∂-finite setting. Similarly to the previous subsection, let $f(\boldsymbol{x}, t)$ be a ∂-finite function described by its annihilating ideal I which is given by a Gröbner basis G. The idea is to iteratively execute Algorithm 5.2 on the telescoping operator

$$\sum_{\boldsymbol{j}\in \boldsymbol{J}_L} p_{\boldsymbol{j}}(\boldsymbol{x})\boldsymbol{\partial}_{\boldsymbol{x}}^{\boldsymbol{j}} + \partial_t \cdot (q_1 m_1 + q_2 m_2 + \cdots + q_s m_s), \tag{5.68}$$

where $\boldsymbol{J}_L$ is a finite set of exponent vectors defined by

$$\boldsymbol{J}_L := \{\boldsymbol{j} := (j_1, \ldots, j_n);\ |\, \boldsymbol{j}\,| = \sum_{i=1}^{n} j_i \le L \in \mathbb{N}\}, \tag{5.69}$$

$p_{\boldsymbol{j}}(\boldsymbol{x}) \in \mathbb{K}(\boldsymbol{x})$, q_i and m_i as in (5.46). The algorithm starts with $\boldsymbol{J}_0 = \{(0, \ldots, 0)\}$ which corresponds to Chyzak's indefinite algorithm. If the function f is indefinite summable then the algorithm stops, otherwise it proceeds iteratively with expanded sets $\boldsymbol{J}_L$.

Example 5.14
In this example, we want to determine a recurrence equation for

$$s(n) := \sum_{n=k}^{n} \binom{n}{k}^3 \tag{5.70}$$

using Chyzak's definite algorithm.

Maple Session 5.7 (Chyzak's definite algorithm: recurrence equation for $s(n) := \sum_{n=k}^{n} \binom{n}{k}^3$)

```
>  with(Ore_algebra):
>  with(Groebner):
```

We define the Ore algebra

```
>  A:=shift_algebra(shift=[Sn,n],shift=[Sk,k],func=[Q,a,b,c]);
```

$$A := \mathit{Ore\_algebra}$$

The summand $\binom{n}{k}^3$ is annihilated by the following operator system:

```
>  G1:=(k+1)^3*Sk+(-n+k)^3;
```

$$\mathit{G1} := (k+1)^3\,\mathit{Sk} + (-n+k)^3,$$

```
>  G2:=(-n-1+k)^3*Sn+(n+1)^3;
```

$$\mathit{G2} := (-n-1+k)^3\,\mathit{Sn} + (n+1)^3.$$

The Gröbner basis computation is performed w.r.t. the following term order

```
>  T:=MonomialOrder(A,tdeg(Sk,Sn));
```

$$T := \mathit{monomial\_order}$$

which gives

```
>  G:=collect(Basis([G1,G2],T),{Sk,Sn},factor);
```

$$G := [(-n-1+k)^3\,\mathit{Sn} + (n+1)^3,\ (k+1)^3\,\mathit{Sk} + (-n+k)^3].$$

The first two iterations do not deliver a rational solution Q. Hence, for $L = 2$ (see (5.69)) we introduce the polynomial:

```
>  Ansatz:=sort(skew_product(c(n),Sn^2,A)+
>  skew_product(b(n),Sn,A)+a(n)+
>  skew_product(Sk-1,Q(k),A),{Sn,Sk});
```

$$\mathit{Ansatz} := \mathrm{c}(n)\,\mathit{Sn}^2 + \mathrm{Q}(k+1)\,\mathit{Sk} + \mathrm{b}(n)\,\mathit{Sn} + \mathrm{a}(n) - \mathrm{Q}(k).$$

In order to compute a normal form of *Ansatz* w.r.t. G we should make the following substitutions

```
>  Ansatz:=subs({Q(k)=q,Q(k+1)=q1,a(n)=a,b(n)=b,c(n)=c},Ansatz);
```

$$\mathit{Ansatz} := c\,\mathit{Sn}^2 + \mathit{q1}\,\mathit{Sk} + b\,\mathit{Sn} + a - q$$

and define the corresponding Ore algebra and term order

```
>  A_Nf:=shift_algebra(shift=[Sn,n],shift=[Sk,k],comm=[q,q1,a,b,c]);
```

$$\mathit{A\_Nf} := \mathit{Ore\_algebra},$$

```
>  T_Nf:=MonomialOrder(A_Nf,tdeg(Sk,Sn));
```

$$\mathit{T\_Nf} := \mathit{monomial\_order}.$$

A normal form of *Ansatz* w.r.t. G is:

```
>  LNf:=collect(NormalForm(Ansatz,G,T_Nf),{q,q1,a,b,c},factor);
```

$$\mathit{LNf} := a - \frac{(n+1)^3\,b}{(-n-1+k)^3} + \frac{(n+2)^3\,(n+1)^3\,c}{(-n-1+k)^3\,(k-n-2)^3} - q - \frac{(-n+k)^3\,\mathit{q1}}{(k+1)^3}.$$

Equating *LNf* to zero gives

```
>  LNf:=sort(collect(numer(normal(LNf)),{q,q1,a,b,c},factor),
>  {q,q1,a,b,c});
```

$$LNf := (-n-1+k)^3\,(k-n-2)^3\,(k+1)^3\,a - (n+1)^3\,(k-n-2)^3\,(k+1)^3\,b$$
$$+ (n+2)^3\,(n+1)^3\,(k+1)^3\,c - (-n-1+k)^3\,(k-n-2)^3\,(k+1)^3\,q$$
$$- (-n+k)^3\,(-n-1+k)^3\,(k-n-2)^3\,q1$$

and substituting q and $q1$ by $Q(k)$ and $Q(k{+}1)$ leads to an inhomogeneous first order linear recurrence equation in Q

```
>   LRE:=subs({q=Q(k),q1=Q(k+1)},LNf)=0;
```

$$LRE := (-n-1+k)^3\,(k-n-2)^3\,(k+1)^3\,a - (n+1)^3\,(k-n-2)^3\,(k+1)^3\,b$$
$$+ (n+2)^3\,(n+1)^3\,(k+1)^3\,c - (-n-1+k)^3\,(k-n-2)^3\,(k+1)^3\,\mathrm{Q}(k)$$
$$- (-n+k)^3\,(-n-1+k)^3\,(k-n-2)^3\,\mathrm{Q}(k+1) = 0.$$

The main objective is to determine the unknown parameters $a, b, c \in \mathbb{Q}(n)$ appearing in

$$\begin{aligned}(k-n)^3(k-n-1)^3(k-n-2)^3Q(n,k+1) + (k-n-1)^3(k-n-2)^3(k+1)^3\,Q(n,k)\\ = (k-n-1)^3(k-n-2)^3(k+1)^3\,a - (n+1)^3(k-n-2)^3(k+1)^3\,b\\ +(n+2)^3(n+1)^3(k+1)^3\,c \end{aligned}\tag{5.71}$$

which are the coefficients of a potential recurrence equation satisfied by the sum $s(n) = \sum\limits_{n=k}^{n} \binom{n}{k}^3$. The achievement of this task goes necessarily through the determination of rational solutions Q of the recurrence equation (5.71). We define

$$Q(n,k) := \frac{N(n,k)}{D(n,k)} = \frac{N_k}{D_k},\tag{5.72}$$

where $N_k, D_k \in \mathbb{Q}(n)[k]$. If we substitute (5.72) in (5.71) we get:

$$\begin{aligned}(k-n)^3(k-n-1)^3(k-n-2)^3\,\frac{N_{k+1}}{D_{k+1}} + (k-n-1)^3(k-n-2)^3(k+1)^3\,\frac{N_k}{D_k}\\ = (k-n-1)^3(k-n-2)^3(k+1)^3\,a - (n+1)^3(k-n-2)^3(k+1)^3\,b\\ +(n+2)^3(n+1)^3(k+1)^3\,c.\end{aligned}\tag{5.73}$$

Finding Q consists of two main steps:

- Determination of the denominator D_k.
- Determination of an upper bound of the degree of N_k in k.

The idea of the first step is to look for a D_k such that (5.73) is reduced to a recurrence equation in N_k with **polynomial** coefficients. In this case, if we observe (5.73) we should have

$$D_k \mid (k-n-1)^3(k-n-2)^3(k+1)^3,\tag{5.74}$$

and

$$D_{k+1} \mid (k-n)^3(k-n-1)^3(k-n-2)^3.\tag{5.75}$$

The last condition (5.75) means also, by executing a backward shift, that

$$D_k \mid (k-n-1)^3(k-n-2)^3(k-n-3)^3. \tag{5.76}$$

As a result of (5.74) and (5.76), we may set

$$D_k = (k-n-1)^3(k-n-2)^3. \tag{5.77}$$

Substituting (5.77) in (5.73) leads to

$$\begin{aligned}(k-n)^3\, N_{k+1} + (k+1)^3\, N_k &= (k-n-1)^3(k-n-2)^3(k+1)^3\, a \\ &\quad -(n+1)^3(k-n-2)^3(k+1)^3\, b \\ &\quad +(n+2)^3(n+1)^3(k+1)^3\, c\end{aligned} \tag{5.78}$$

which is a recurrence equation in N_k with polynomial coefficients having the form

$$A_k\, N_{k+1} + B_k\, N_k = C_k. \tag{5.79}$$

In order to determine an upper bound of the degree in k of a polynomial solution N_k of (5.79) we need the following proposition:

Proposition 5.15 (Degree upper bound)
For every polynomial solution g_k of

$$A_k\, g_{k+1} + B_k\, g_k = C_k, \tag{5.80}$$

where $A_k, B_k, C_k \in \mathbb{K}[k]$, an upper bound of the degree g_k in k is determined by the following algorithm.

1. If $\deg(A_k - B_k, k) \leq \deg(A_k + B_k, k)$, then

$$\deg(g_k, k) = \deg(C_k, k) - \deg(A_k + B_k, k).$$

2. If $m := \deg(A_k - B_k, k) > \deg(A_k + B_k, k)$, then let a denote the coefficient of k^d in $A_k - B_k$ and b the coefficient of k^{d-1} in $A_k + B_k$.

   a) If $\frac{-2b}{a} \notin \mathbb{N}$, then

   $$\deg(g_k, k) = \deg(C_k, k) - m + 1.$$

   b) If $\frac{-2b}{a} \in \mathbb{N}$, then

   $$\deg(g_k, k) \in \{\frac{-2b}{a}, \deg(C_k, k) - m + 1\}.$$

Proof: (See [Koe06], Proposition 11.9) □

In the following Maple Session, we show how to determine the parameters $a, b, c \in \mathbb{Q}(n)$.

Maple Session 5.8 (Chyzak's definite algorithm: recurrence equation for $s(n) := \sum_{n=k}^{n} \binom{n}{k}^3$ continued)

```
>  A_k:=(k-n)^3;
```

$$A\_k := (k-n)^3$$

```
>  B_k:=(k+1)^3;
```

$$B\_k := (k+1)^3$$

```
>  C_k:=(-n-1+k)^3*(k-n-2)^3*(k+1)^3*a
>  -(n+1)^3*(k-n-2)^3*(k+1)^3*b
>  +(n+2)^3*(n+1)^3*(k+1)^3*c;
```

$$C\_k := (-n-1+k)^3\,(k-n-2)^3\,(k+1)^3\,a - (n+1)^3\,(k-n-2)^3\,(k+1)^3\,b + (n+2)^3\,(n+1)^3\,(k+1)^3\,c$$

```
>  degree(expand(A_k-B_k),k);
```

$$2$$

```
>  degree(expand(A_k+B_k),k);
```

$$3$$

We see that Condition $a)$ of Proposition 5.15 is fulfilled. Hence, an upper bound of the degree of N_k in k is

```
>  degree(expand(C_k),k)-degree(expand(A_k+B_k),k);
```

$$6$$

and N_k has the following form

```
>  N:=(k)->sum(r[i]*k^i,i=0..6);
```

$$N := k \to \sum_{i=0}^{6} r_i\,k^i.$$

All we have to do now is to substitute N_k in (5.78), equate the coefficients in k and solve the resulting linear system of equations

```
>  Eq:=collect((k-n-2)^3*N(k+1)+(k+1)^3*N(k)-C_k,k):
>  COEFF:={coeffs(Eq,k)}:
>  vars:={a,b,c,seq(r[i],i=0..6)};
```

$$vars := \{a,\, b,\, c,\, r_0,\, r_1,\, r_2,\, r_3,\, r_4,\, r_5,\, r_6\}$$

to get the following solution

```
>  SOL:=solve(COEFF,vars);
```

$$SOL := \{a = 2\,r_6,\ b = \frac{1}{4}\,\frac{r_6\,(7\,n^2+21\,n+16)}{n^2+2\,n+1},\ c = -\frac{1}{4}\,\frac{(n^2+4\,n+4)\,r_6}{n^2+2\,n+1},\ r_0 = 0,\ r_1 = 0,\ r_2 = 0,$$
$$r_3 = -\frac{1}{2}\,(7\,n^3+37\,n^2+64\,n+36)\,r_6,\ r_4 = \frac{3}{4}\,(9\,n^2+31\,n+26)\,r_6,$$
$$r_5 = -\frac{3}{2}\,(5+3\,n)\,r_6,\ r_6 = r_6\}.$$

We recall that a potential recurrence equation satisfied by $s(n) = \sum_{n=k}^{n} \binom{n}{k}^3$ has the following form

```
>  REC:=c*Sn^2+b*Sn+a;
```

$$REC := c\,Sn^2 + b\,Sn + a.$$

Substituting the free parameter r_6 by -1 in SOL we get the recurrence we look for

```
>  REC:=collect(subs(r[6]=-1,
>  numer(normal(subs(SOL,REC)))),Sn,factor);
```

$$REC := (n+2)^2\,Sn^2 + (-16 - 21\,n - 7\,n^2)\,Sn - 8\,(n+1)^2.$$

Bibliography

[AB98] S. A. Abramov and M. A. Barkatou. Rational solutions of first order linear difference systems. In *Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation*, ISSAC '98, pages 124–131, New York, NY, USA, 1998. ACM. 85

[ABP95] S. A. Abramov, M. Bronstein, and M. Petkovsek. On polynomial solutions of linear operator equations. In *Proceedings of the 1995 International Symposium on Symbolic and Algebraic Computation*, ISSAC '95, pages 290–296, New York, NY, USA, 1995. ACM. 85

[Abr89] S. A. Abramov. Rational solutions of linear differential and difference equations with polynomial coefficients. *USSR Computational Mathematics and Mathematical Physics*, 29(6):7–12, 1989. 85

[Abr95] S. A. Abramov. Rational solutions of linear difference and q-difference equations with polynomial coefficients. In *Proceedings of the 1995 International Symposium on Symbolic and Algebraic Computation*, ISSAC '95, pages 285–289, New York, NY, USA, 1995. ACM. 85

[AS64] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions*. Dover Publ, New York, 1964. 71

[AZ90] G. Almkvist and D. Zeilberger. The method of differentiating under the integral sign. *J. Symbolic Comput.*, 10:571–591, 1990. 73

[AZ96] S. A. Abramov and E. V. Zima. A universal program to uncouple linear systems. In *Proceedings of the International Conference on Computational Modelling and Computing*, pages 16–26, Dubna, Russia, September 1996. 85

[Bar99] M. A. Barkatou. On rational solutions of systems of linear differential equations. *J. Symbolic Comput.*, 28(4-5):547–567, 1999. 85

[Ber72] I. N. Bernstein. The analytic continuation of generalized functions with respect to a parameter. *Functional Analysis and its Applications*, 6(4):273–285, 1972. 50, 55

[BG88] A.D. Bell and K.R. Goodearl. Uniform rank over differential operator rings and Poincaré-Birkoff-Witt extensions. *Pacific J. Math.*, 131:13–37, 1988. 22, 23

[BKW93] T. Becker, H. Kredel, and V. Weispfenning. *Gröbner Bases: A Computational Approach to Commutative Algebra*. Springer-Verlag, London, UK, 1 edition, 1993. 18, 44

[BP96] M. Bronstein and M. Petkovsek. An introduction to pseudo-linear algebra. *Theoretical Computer Science*, 157:3–33, 1996. 3

[Buc65] B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem multidimensianalen Polynomideal*. PhD thesis, Insbruck, 1965. 28

[Buc85] B. Buchberger. Gröbner bases: An algorithmic method in polynomial ideal theory. In *Recent Trends in Multidimensional System Theory*, pages 184–232. N. K. Bose, D. Reidel Publishing, 1985. 28

[CGH89] L. Caniglia, A. Galligo, and J. Heintz. Some new effectivity bounds in computational geometry. In *AAECC-6: Proceedings of the 6th International Conference on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, pages 131–151. Springer-Verlag, London, UK, 1989. 43

[Chy98] F. Chyzak. *Fonctions holonomes en calcul formel*. Thèse universitaire, École polytechnique, 1998. INRIA, TU 0531. 227 pages. 17, 40, 41

[Chy00] F. Chyzak. An extension of Zeilberger's fast algorithm to general holonomic functions. *Discrete Math.*, 217:115–134, 2000. 82

[Cou95] S.C. Coutinho. *A Primer of Algebraic D-modules*. Cambrige University, New York, NY, USA, 1995. 52, 54, 55, 59

[CS96] F. Chyzak and B. Salvy. Non-commutative elimination in Ore algebras proves multivariate identities. *J. Symbolic Comput.*, 26:187–227, 1996. 10, 64, 74

[Fas47] M. C. Fasenmyer. Some generalized hypergeometric polynomials. *Bull.Amer. Math. Soc.*, 53:806–813, 1947. 63

[Fas49] M. C. Fasenmyer. A note on pure recurrence relations. *Amer. Math. Monthly*, 56:14–17, 1949. 63, 71

[FGDM93] J. C. Faugère, P. Gianni, D.Lazard, and T. Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *J. Symbolic Comput.*, 16(4):329–344, 1993. 30, 43

[GMN$^+$91] A. Giovini, T. Mora, G. Niesi, L. Robbiano, and C. Traverso. ”One sugar cube, please” or Selection strategies in the Buchberger algorithm. In *Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation*, ISSAC '91, pages 5–4, New York, NY, USA, 1991. ACM. 31

[Jac62] N. Jacobson. *Lie algebras*. Interscience Publishers, Toronto, Ontario, Canada, 1962. 22

[Kas78] M. Kashiwara. On the holonomic systems of linear differential equations. *Inventiones Mathematicae*, 49(2):121–135, 1978. 59

[Koe95] W. Koepf. Identities for families of orthogonal polynomials and special functions. *Integral Transforms Spec. Funct*, 5(1–2):69–102, 1995. 34, 36

[Koe98] W. Koepf. *Hypergeometric Summation*. Vieweg, Braunschweig, Wiesbaden, 1998. 84, 88

[Koe06] W. Koepf. *Computeralgebra – Eine algorithmisch orientierte Einführung*. Springer Verlag, Berlin, Heidelberg, New York, 2006. 84, 93

[Kre93] H. Kredel. *Solvable Polynomial Rings*. Verlag Shaker, Aachen, 1993. 17, 18, 21, 22, 23, 24

[KRW90] A. Kandri-Rody and V. Weispfenning. Non-commutative Gröbner bases in algebras of solvable type. *J. Symbolic Comput.*, 9(1):1–26, 1990. 18, 22

[Laz92] D. Lazard. Solving zero-dimensional algebraic systems. *J. Symbolic Comput.*, 13(2):117–131, 1992. 30, 43

[Lev05] V. Levandovskyy. *Non-commutative Computer Algebra for Polynomial Algebras: Gröbner Bases, Applications and Implementation*. PhD thesis, Fachbereich Mathematik, Universität Kaiserslautern, 2005. 17, 21, 24, 29

[MML08] J. Martin-Morales and V. Levandovskyy. Computational D-module theory with singular, comparison with other systems and two new algorithms. In *Proceedings of the 2008 international symposium on Symbolic and algebraic computation*, ISSAC '08, pages 173–180, New York, NY, USA, 2008. ACM. 59, 65

[MR87] J.C. McConnell and J.C. Robson. *Non-commutaive Noetherian Rings*. Wiley, New York, USA, 1987. 23

[Oak97] T. Oaku. Algorithms for the b-function and D-modules associated with a polynomial. *J. Pure Appl. Algebra*, 9(117-118):495–518, 1997. 59

[Ore33] O. Ore. Theory of non-commutative polynomials. *Ann. of Math. (2)*, 34:480–508, 1933. 3, 10, 12

[PWZ96] M. Petkovsek, H. S. Wilf, and D. Zeilberger. *A=B*. A. K. Peters, Ltd. Wellesley MA, 1996. 84

[Spr04] T. Sprenger. Algorithmen für mehrfache Summen. Master's thesis, Universität Kassel, 2004. 71

[SST00] M. Saito, B. Sturmfels, and N. Takayama. *Gröbner Deformations of Hypergeometric Differential Equations*, volume 6 of *Algorithms Compt. Math.* Springer, Berlin, Heidelberg, New York, 2000. 59

[Sta80] R. P. Stanley. Differentiably finite power series. *European J. Combin.*, 1:175–188, 1980. 39

[Str93] V. Strehl. Binomial sums and identities. 10:37–49, 1993. 66

[Tak90a] N. Takayama. An algorithm of constructing the integral of a module–an infinite dimensional analog of Gröbner basis. In *Proceedings of the 1990 international symposium on Symbolic and algebraic computation*, ISSAC '90, pages 206–211, New York, NY, USA, 1990. ACM. 64, 74

[Tak90b] N. Takayama. Gröbner basis, integration and transcendental functions. In *Proceedings of the 1990 international symposium on Symbolic and algebraic computation*, ISSAC '90, pages 152–156, New York, NY, USA, 1990. ACM. 64, 74

[Tak92] N. Takayama. An approach to zero recognition of binomial sums and its complexity. *J. Symbolic Comput.*, 14:265–282, 1992. 59

[Tra89] C. Traverso. Gröbner trace algorithms. In *Proceedings of the 1988 International Symposium on Symbolic and Algebraic Computation*, ISSAC '88, pages 125–138. Springer-Verlag, London, UK, 1989. 32

[Tsa00a] H. Tsai. *Algorithms for algebraic analysis*. PhD thesis, University of California, Berkeley, 2000. 60

[Tsa00b] H. Tsai. Weyl closure of a linear differential operator. *J. Symbolic Comput.*, 9(4-5):747–775, 2000. 60

[vdP78] A. van der Poorten. A proof that Euler missed . . . Apéry's proof of the irrationality of $\zeta(3)$. *Math. Intelligencer*, 1:195–203, 1978. 77

[WZ91] H. S. Wilf and D. Zeilberger. An algorithmic proof theory for hypergeometric (ordinary and "q") multisum/ integral identities, 1991. 59

[Zei90a] D. Zeilberger. A fast algorithm for proving terminating hypergeometric identities. *Discrete Math.*, 80:207–211, 1990. 64

[Zei90b] D. Zeilberger. A holonomic systems approach to special functions identities. *J. Comput. Appl. Math.*, 32(3):321–368, 1990. 39, 56, 59, 63, 64, 65

List of Algorithms

3.1 **Left normal form: LNF** . . . 27
3.2 **Reduced left normal form: RedLNF** . . . 28
3.3 **Left Buchberger's algorithm** . . . 30
4.1 **FGLM** . . . 44
4.2 **Extended FGLM for ∂-finiteness under sum** . . . 46
4.3 **Extended FGLM for ∂-finiteness under product** . . . 47
4.4 **Extended FGLM for ∂-finiteness under the action of an Ore operator** . . . 48
5.1 **Extended Takayama's algorithm** . . . 77
5.2 **Chyzak's algorithm** . . . 86

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Dissertation selbständig und ohne unerlaubte Hilfe angefertigt und andere als die in der Dissertation angegebenen Hilfsmittel nicht benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen sind, habe ich als solche kenntlich gemacht. Kein Teil dieser Arbeit ist in einem anderen Promotions- oder Habilitationsverfahren verwendet worden.