JOHANNES KEPLER UNIVERSITÄT LINZ | JKU

TNF
Technisch-Naturwissenschaftliche Fakultät

Advanced Applications of the Holonomic Systems Approach

DISSERTATION

zur Erlangung des akademischen Grades

Doktor

im Doktoratsstudium der

Technischen Wissenschaften

Eingereicht von:
Christoph Koutschan

Angefertigt am:
Institut für Symbolisches Rechnen

Beurteilung:
Univ.-Prof. Dr. Peter Paule
Prof. Dr. Marko Petkovšek

Linz, September, 2009

Christoph Koutschan

Advanced Applications of the Holonomic Systems Approach

Doctoral Thesis

Research Institute for Symbolic Computation
Johannes Kepler University Linz

Advisor: Univ.-Prof. Dr. Peter Paule

Examiners:
Univ.-Prof. Dr. Peter Paule
Prof. Dr. Marko Petkovšek

This work has been supported by the FWF grants F013 and P20162.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Dissertation selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Linz, September 2009 Christoph Koutschan

Abstract

The holonomic systems approach was proposed in the early 1990s by Doron Zeilberger. It laid a foundation for the algorithmic treatment of holonomic function identities. Frédéric Chyzak later extended this framework by introducing the closely related notion of ∂-finite functions and by placing their manipulation on solid algorithmic grounds. For practical purposes it is convenient to take advantage of both concepts which is not too much of a restriction: The class of functions that are holonomic and ∂-finite contains many elementary functions (such as rational functions, algebraic functions, logarithms, exponentials, sine function, etc.) as well as a multitude of special functions (like classical orthogonal polynomials, elliptic integrals, Airy, Bessel, and Kelvin functions, etc.). In short, it is composed of functions that can be characterized by sufficiently many partial differential and difference equations, both linear and with polynomial coefficients. An important ingredient is the ability to execute closure properties algorithmically, for example addition, multiplication, and certain substitutions. But the central technique is called creative telescoping which allows to deal with summation and integration problems in a completely automatized fashion.

Part of this thesis is our Mathematica package `HolonomicFunctions` in which the above mentioned algorithms are implemented, including more basic functionality such as noncommutative operator algebras, the computation of Gröbner bases in them, and finding rational solutions of parameterized systems of linear differential or difference equations.

Besides standard applications like proving special function identities, the focus of this thesis is on three advanced applications that are interesting in their own right as well as for their computational challenge. First, we contributed to translating Takayama's algorithm into a new context, in order to apply it to an until then open problem, the proof of Ira Gessel's lattice path conjecture. The computations that completed the proof were of a nontrivial size and have been performed with our software. Second, investigating basis functions in finite element methods, we were able to extend the existing algorithms in a way that allowed us to derive various relations which generated a considerable speed-up in the subsequent numerical simulations, in this case of the propagation of electromagnetic waves. The third application concerns a computer proof of the enumeration formula for totally symmetric plane partitions, also known as Stembridge's theorem. To make the underlying computations feasible we employed a new approach for finding creative telescoping operators.

Kurzzusammenfassung

In den frühen 1990er Jahren beschrieb Doron Zeilberger den „holonomic systems approach“, der eine Grundlage für das algorithmische Beweisen von Identitäten holonomer Funktionen bildete. Dies wurde später von Frédéric Chyzak um den eng verwandten Begriff der ∂-finiten Funktionen erweitert, deren Handhabung er auf eine solide algorithmische Basis stellte. In der Praxis ist es von Vorteil, sich beider Konzepte gleichzeitig zu bedienen, was auch keine allzu große Einschränkung bedeutet: Die Klasse der Funktionen, die holonom und ∂-finit sind, umfasst viele elementare Funktionen (z.B. rationale und algebraische Funktionen, Logarithmen, Exponentialfunktionen, Sinus, etc.) ebenso wie eine Vielzahl von Speziellen Funktionen (z.B. klassische orthogonale Polynome, elliptische Integrale, Airy-, Bessel- und Kelvin-Funktionen, etc.). Grob gesagt sind dies Funktionen, die durch eine ausreichende Anzahl von partiellen Differential- und Differenzengleichungen (linear und mit Polynomkoeffizienten) charakterisiert werden können. Ein wichtiger Aspekt besteht in der Möglichkeit, Abschlusseigenschaften wie zum Beispiel Addition, Multiplikation und bestimmte Substitutionen algorithmisch auszuführen. Das zentrale Verfahren ist jedoch das „creative telescoping“, mit welchem Summations- und Integrationsprobleme vollkommen automatisiert behandelt werden können.

Als Teil dieser Arbeit haben wir besagte Algorithmen in unserem Mathematica-Paket `HolonomicFunctions` implementiert sowie einige Basisfunktionalitäten, darunter nichtkommutative Operatoralgebren, Gröbnerbasenberechnung in diesen und das Auffinden von rationalen Lösungen parametrisierter Systeme von linearen Differential- bzw. Differenzengleichungen.

Neben Standardanwendungen, wie dem Beweisen von Formeln für Spezielle Funktionen, liegt das Hauptaugenmerk dieser Arbeit auf drei neuartigen Anwendungen, die, obschon für sich interessant, auch aufgrund ihrer Rechenintensität eine Herausforderung darstellten. In der ersten wurde der Algorithmus von Takayama so umformuliert, dass er auf ein bis dato offenes Problem, Ira Gessels Gitterpfad-Vermutung, angewendet werden konnte. Die zum Beweis notwendigen Rechnungen waren sehr aufwändig und wurden mit unserer Software durchgeführt. Zweitens erweiterten wir bekannte Verfahren, um Beziehungen für die Basisfunktionen in Finite-Elemente-Methoden herzuleiten. Diese erbrachten eine beachtliche Beschleunigung der numerischen Simulationen, in diesem Fall von der Ausbreitung elektromagnetischer Wellen. Die dritte Anwendung besteht aus einem Computerbeweis der Abzählformel für totalsymmetrische plane partitions, auch bekannt als das Theorem von Stembridge. Um die nötigen Berechnungen zu ermöglichen, verwendeten wir einen neuen Ansatz zum Auffinden von „creative telescoping“-Operatoren.

Acknowledgments

"The direction in which education starts a man will determine his future life" (Plato). In this sense I would like to thank my teachers Achim Schmidt, Hans Grabmüller, and Volker Strehl, who induced the successful end of my PhD studies, and possibly much more, by providing me with a well-founded education in mathematics.

The best education is rather useless if you cannot exchange with other people: for answering my questions and for putting me up to new ideas, I thank Clemens Bruschek, Herwig Hauser, Ralf Hemmecke, Guenter Landsmann, Viktor Levandovskyy, Victor Moll, Brian Moore, Georg Regensburger, Markus Rosenkranz, Joachim Schöberl, Hans Schönemann, and Burkhard Zimmermann.

Special thanks go to Doron Zeilberger who stimulated my motivation with his enthusiasm and several prizes, and with whom to collaborate I enjoyed greatly. Frédéric Chyzak was so kind to facilitate my two visits to INRIA-Rocquencourt where I profited a lot from many inspiring discussions for which he reserved all his available time. I am deeply grateful to Marko Petkovšek for inviting me to Ljubljana and for agreeing to serve as my second advisor; his detailed comments were of great help! I also want to acknowledge the financial support of the Austrian Science Fund FWF, namely the grants F013 and P20162.

Two among my many colleagues deserve particular mention. Since we share the same office it is needless to say that they helped in all kinds of problems and questions, quickly and at any time. Veronika Pillwein did an excellent job in drawing my attention to numerous special function applications in numerical analysis. Manuel Kauers was so generous to let me participate in his mathematical knowledge and insight; by means of our daily discussions he contributed considerably to my work.

I would like to express my gratitude to my advisor Peter Paule who accepted me at RISC and who in the first place made my PhD studies possible by reliably arranging sufficient financial support. I highly enjoyed his vivid lectures and every minute of his scarce time that he spent on advising me. He wisely suggested the perfect thesis topic and infected me with his energy again and again.

Finally I would like to thank my parents and my girlfriend Martina for their unconditional support (thanks for all the cake!) and for their understanding and appreciation of my work (in spite of not understanding the subject).

Contents

Chapter 1

Introduction

1.1 Objectives

Based on Zeilberger's holonomic systems approach, Frédéric Chyzak did a great job in putting the treatment of holonomic functions on solid algorithmic grounds [24, 25, 28]. But we think that from the applications point of view, his work did not yet receive the attention that it deserves. The reasons for that could be that his thesis is written in French, or that his Maple implementation `Mgfun` suffered from certain weaknesses, and only since very recently the work on this package has been restarted. However, one objective of this thesis is to provide a new implementation (in Mathematica) of the related algorithms, which in any case is very desirable for several reasons: to enable interaction with many other Mathematica packages of the RISC combinatorics group, and to make comparison of results possible. Last but not least it served to increase our understanding of these algorithms.

We have put some effort to design a user-friendly software that may help mathematicians and other scientists in performing various tasks: the most natural application consists in proving identities, in particular those involving special functions. A large amount of identities that can be found in voluminous standard tables and books [74, 6, 9, 42] lies in the scope of our package. Besides proving that a given evaluation is correct, the software can also assist in finding a closed form for a sum or integral, by delivering a recurrence or differential equation for it, which then can possibly be solved by other means. Furthermore, in many applications like numerical analysis or particle physics, one is interested in certain relations for a given expression (like a combination of special functions or an integral); also in such cases our package can be of great help. Some applications forced us to approach the limit of what today's computers can accomplish, which lead us to optimize the package again and again, and which is mirrored in the title by the attribute "advanced".

The plan of this thesis is as follows. Chapter 2 introduces the notions of holonomic and ∂-finite functions, and investigates their qualities and closure properties (in particular two kinds of substitutions that are not described explicitly in Chyzak's work). Chapter 3 is dedicated to diverse summation and integration algorithms for holonomic and ∂-finite functions, all of them making use of the method of creative telescoping. Chapter 4 gives an overview of the functionality of our package `HolonomicFunctions` and many examples how this software may typically be applied. In Chapters 5, 6, and 7 three advanced applications of our software are presented, each of which required slight modifications and extensions of the classical algorithms of Zeilberger and Chyzak, in addition to considerable amounts of computing time.

Our software `HolonomicFunctions` is freely available from the RISC combinatorics software page

http://www.risc.uni-linz.ac.at/research/combinat/software/

1.2 Preliminaries

We assume that the reader is familiar with the theory of Gröbner bases developed by Bruno Buchberger in the 1960s [22]. This theory has been adapted to noncommutative polynomial rings, in a very general and theoretic fashion in [15], and more algorithmically but less general in [44]. For our purposes the latter suffices where a Gröbner basis theory for rings of solvable type is developed. We do not want to reproduce it here, but only mention a central property of such rings. If $(R, 0, 1, +, -, *)$ with $R = \mathbb{K}[X_1, \ldots, X_d]$ and $* : R^2 \to R$ is a ring of solvable type, then (by definition) there exist $0 \neq c_{ij} \in \mathbb{K}$ and $p_{ij} \in R$ such that

$$X_j * X_i = c_{ij} X_i X_j + p_{ij} \tag{1.1}$$

for all $1 \leq i \leq j \leq d$. Moreover, R is noetherian with respect to a term order $\prec$, if $p_{ij} \prec X_i X_j$ for all $1 \leq i \leq j \leq d$. In this situation the computation of Gröbner bases is possible. In noncommutative polynomial rings we will mostly deal with left ideals and left Gröbner bases. Nice introductions into the theory of Göbner bases are given in [31, 14].

1.3 Notations

This part (how could it be different) starts with the well-known mathematician's joke "let $\mathbb{K}$ be a field". In addition, we tacitly assume that $\mathbb{K}$ is computable, commutative and has characteristic 0, or in other words $\mathbb{Q} \subseteq \mathbb{K}$. We use bold letters for multivariables and multiindices, i.e., the polynomial ring $\mathbb{K}[x_1, \dots, x_d]$ in several variables may be abbreviated by $\mathbb{K}[\boldsymbol{x}]$, as well as the power product $x_1^{\alpha_1} \dots x_d^{\alpha_d}$ that we often will write as $\boldsymbol{x}^{\boldsymbol{\alpha}}$. We refer to such a power product as a *monomial*. By $\boldsymbol{\alpha} \leq \boldsymbol{\beta}$ for two multiindices $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{N}^d$ we mean that $\alpha_1 \leq \beta_1, \dots, \alpha_d \leq \beta_d$. When speaking about the *support* of a polynomial, we refer to the finite set of power products (monomials) whose coefficients are nonzero.

We use the notation ${}_R\langle G_1, \dots, G_r\rangle$ to refer to the left ideal in R that is generated by $G_1, \dots, G_d$, in symbols

$$ {}_R\langle G_1, \dots, G_r\rangle := \{R_1 G_1 + \dots + R_r G_r \mid R_i \in R\} $$

(and similarly $\langle G_1, \dots, G_r\rangle_R$ for right ideals).

Since we will work with operators all the time let us introduce the following notation: The bullet symbol $\bullet$ is used for operator application, i.e., $P \bullet f = P(f)$ means that the operator P is applied to f (where f can be a function, for example). The multiplication inside the operator algebra is denoted by the usual dot (which we sometimes omit): $P \cdot Q = PQ$. We introduce some operator symbols that will be used throughout.

- *Differential operator*: We will use the symbol D_x to denote the operator "partial derivative with respect to x", in other words, $D_x \bullet f = \frac{\partial f}{\partial x}$. Often in the literature this operator is referred to by ∂_x, but following Chyzak we would like to use the symbol ∂ in a more general sense (see below). Using the differential operator we can represent differential equations in a convenient way: Consider the Hermite polynomials $H_n(x)$ that fulfill the second-order differential equation $H_n''(x) - 2xH_n'(x) + 2nH_n(x) = 0$. In operator notation it translates to $D_x^2 - 2xD_x + 2n$.

- *Shift operator*: By S_n we denote the shift operator with respect to n, this means that $S_n \bullet f(n) = f(n+1)$. We will use the operator notation for expressing and manipulating recurrence relations. For example, the Fibonacci recurrence $F_{n+2} = F_{n+1} + F_n$ translates to $S_n^2 - S_n - 1$.

- *Forward difference*: The symbol Δ_n denotes the forward difference operator (which is the discrete analogue of the differential operator): $\Delta_n \bullet f(n) = f(n+1) - f(n)$.

- *Euler operator*: The symbol θ_x is sometimes used to denote the Euler operator which is $\theta_x = xD_x$.
- *q-shift operator*: We denote it by $S_{x,q}$ and it acts as $S_{x,q} \bullet f(x) = f(qx)$.
- *Generic operator*: As a unifying symbol we use ∂ that may stand for an arbitrary Ore operator (see Section 2.3) symbol (in particular any of the above).

We want to point out that the notion "operator" is occupied with two slightly different meanings: First, the word might refer to an operator symbol as introduced above, e.g., by calling D_x the "partial differential operator". Second, it can denote an equation that is given in operator notation, usually being a polynomial in the previous ones, e.g., the Hermite differential equation or the Fibonacci recurrence. So, we can speak of the operator $S_n^2 - S_n - 1$ which itself is a polynomial in the shift operator S_n. In order to avoid confusion we want to refer to the latter as "Ore operator" and reserve the general notion for operators of the first type. We will only deal with linear operators; these are operators that are expressible as polynomials as exemplified above.

A last word on the notion of variables: When dealing with multivariate functions $f(v_1, \ldots, v_d)$, the variables $v_1, \ldots, v_d$ are often of different natures. We call a variable on which we act with the shift or the difference operator (and that is usually supposed to take integer values only), a *discrete variable*. On the other hand, there are variables on which we act with the differential or the Euler operator; these variables we call *continuous variables* and they are allowed to take real or complex values in general.

Chapter 2

Holonomic and ∂-finite functions

2.1 Basic concepts

Before we give a formal definition of (multivariate) holonomic functions, we want to slowly approach this widely used concept. Let us for the moment consider a univariate formal power series $f \in \mathbb{K}[\![x]\!]$. If the derivatives $\frac{\mathrm{d}^i f}{\mathrm{d}x^i}$ of such a power series span a finite-dimensional $\mathbb{K}(x)$-vector space, then f is called *D-finite* (differentiably finite), a notion that has been introduced by Richard Stanley [79] in the early 1980s. He then proved that f being D-finite is equivalent to say that it fulfills a linear differential equation with polynomial coefficients (D-finite differential equation):

$$p_d(x)f^{(d)}(x) + \cdots + p_1(x)f'(x) + p_0(x)f(x) = 0, \quad p_i \in \mathbb{K}[x], p_d \neq 0. \tag{2.1}$$

In operator notation the above equation reads as $P \bullet f = 0$ where

$$P = p_d(x)D_x^d + \cdots + p_1(x)D_x + p_0(x) \tag{2.2}$$

and we will identify both objects with each other.

We now want to study the properties of the operator arithmetic. From the Leibniz law $(x \cdot f(x))' = x \cdot f'(x) + f(x)$ we can read off the following commutation rule: $D_x x = xD_x + 1$. Hence when dealing with differential operators we have to take into account that x does not commute with D_x. Operators of the form (2.2) are usually represented in the *Weyl algebra* A_1 (the index 1 indicates that one variable x and the corresponding differential operator D_x are involved). More precisely, the first Weyl algebra A_1 is defined as follows:

$$A_1 := A_1(\mathbb{K}) := \mathbb{K}\langle x, D_x\rangle / \langle D_x x - xD_x - 1\rangle$$

(we omit the field $\mathbb{K}$ when it is clear from the context). The angle brackets denote the free algebra in x and D_x from which we divide out the commutation relation from above. The *standard monomials*—those monomials where the variable x is on the left and the differential operator D_x is on the right so that they are of the form $x^\alpha D_x^\beta$—constitute a basis of $A_1(\mathbb{K})$ as a $\mathbb{K}$-vector space; it is called the *canonical basis.* The proof that this is indeed a basis can be found for example in [30, Chapter 1, Proposition 2.1]. We write elements of the Weyl algebra in *canonical form*, this means in expanded form as a linear combination of monomials from the canonical basis:

$$\sum_{(\alpha,\beta)\in I} c_{\alpha,\beta} x^\alpha D_x^\beta$$

where I is a finite index set in $\mathbb{N}^2$.

It is now natural to consider not only a single operator P as given in (2.2) but the *left ideal*

$$_{A_1}\langle P\rangle := \{QP \mid Q \in A_1\}$$

that is generated by this operator. The reason for doing so is obvious: We can multiply the relation (2.1) by x and it will still be true, as well as we can differentiate it, which corresponds to the multiplication $D_x \cdot P$. Hence all elements of this left ideal are annihilating operators of $f(x)$. On the other hand, the left ideal that contains all operators in A_1 which send $f(x)$ to zero, is called the *annihilator* of $f(x)$ and is denoted by

$$\operatorname{Ann}_{A_1}(f) := \{P \in A_1 \mid P \bullet f = 0\}.$$

Note that the two left ideals $\operatorname{Ann}_{A_1}(f)$ and $I = {}_{A_1}\langle P\rangle$ need not to be equal for several reasons. If P is not the minimal differential equation for $f(x)$ with respect to order, or if it contains a nonconstant polynomial content among its coefficients, then clearly I does not constitute the whole annihilator of $f(x)$. But there are more subtle reasons as the following example shows.

Example 2.1. *Let $f(x) = x^3$; then the homogeneous differential equation of minimal order and degree is $xf'(x) - 3f(x) = 0$, which is represented by the operator $P = xD_x - 3$. Nevertheless the left ideal ${}_{A_1}\langle P\rangle$ does not constitute the whole annihilator since D_x^4 (which also annihilates x^3) is not contained in it.*

In addition Stanley [79] introduced the notion of *P-finite* (or *P-recursive*) sequences. These are sequences $(a(n))_{n\in\mathbb{N}} \in \mathbb{K}^{\mathbb{N}}$ that fulfill a linear recurrence with polynomial coefficients (P-finite recurrence)

$$p_d(n)a(n+d)+\cdots+p_1(n)a(n+1)+p_0(n)a(n) = 0, \quad p_i \in \mathbb{K}[n], p_d \neq 0. \quad (2.3)$$

In operator notation equation (2.3) reads as $P \bullet a = 0$ where

$$P = p_d(n)S_n^d + \cdots + p_1(n)S_n + p_0(n). \tag{2.4}$$

In order to represent operators like (2.4) we introduce the *shift algebra*. Its commutation rule is $S_n n = (n+1)S_n = nS_n + S_n$ and hence the shift algebra can be defined by

$$\mathbb{K}\langle n, S_n\rangle / \langle S_n n - nS_n - S_n\rangle$$

as a discrete analogue of the Weyl algebra. It is well known that a P-finite recurrence can be transformed into a D-finite differential equation for the corresponding generating function and vice versa. Implementations like Maple's `gfun` [77] or the Mathematica package `GeneratingFunctions` [59] perform these operations. We close this section by a concrete example.

Example 2.2. *The error function* $\operatorname{erf}(x)$ *is defined via the probability integral*

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2)\,\mathrm{d}t$$

and a differential equation $\operatorname{erf}''(x) = -2x\operatorname{erf}'(x)$ *is easily derived from this definition. Thus the operator* $D_x^2 + 2xD_x$ *annihilates* $\operatorname{erf}(x)$*, and since* $\operatorname{erf}(x)$ *can be expanded into a power series in the point* $x = 0$*, we can say that it is D-finite. The coefficient sequence of its series expansion is P-finite and is annihilated be the recurrence operator* $(k^2 + 3k + 2)S_k^2 + 2k$*.*

In the following two sections we will discuss two approaches how D-finiteness and P-finiteness can be generalized to functions and sequences in several variables. This will lead to the classes of holonomic and ∂-finite functions.

2.2 Holonomic functions

From now on we deal with functions in several, say d continuous variables, or in other words $f : \mathbb{K}^n \to \mathbb{K}, (x_1, \ldots, x_d) \mapsto f(x_1, \ldots, x_d)$. The definition of holonomy for such functions can be quite involved and cumbersome—we try to keep it as simple as possible. For more elaborated expositions see [93, 23, 24]. Again we will consider operators that annihilate f; these are now partial differential equations with respect to the variables $x_1, \ldots, x_d$. To represent such operators we have to introduce the Weyl algebra in d variables:

$$A_d := A_d(\mathbb{K}) := \mathbb{K}\langle x_1, \ldots, x_d, D_{x_1}, \ldots, D_{x_d}\rangle$$

modulo the relations

$$\langle D_{x_i}x_i - x_iD_{x_i} - 1, x_ix_j - x_jx_i, D_{x_i}D_{x_j} - D_{x_j}D_{x_i}, x_iD_{x_j} - D_{x_j}x_i\rangle \quad (i \neq j).$$

This means that all generators commute except for the pairs x_i and D_{x_i} for which the commutation relations are $D_{x_i}x_i = x_iD_{x_i} + 1$. Although the Weyl algebra is very close to a commutative polynomial ring, its (left) ideals have quite different properties (see Example 2.5 below).

A multivariate function $f(x_1, \ldots, x_d)$ is called *holonomic* if there exists a left ideal of annihilating operators in A_d that has a certain property, namely being holonomic. This notion has its origin in D-module theory, where it describes A_d-modules of minimal Bernstein dimension. Informally but sufficient for our needs, the Bernstein dimension coincides with the notion of dimension that is known from commutative algebra and which uses the Hilbert polynomial. In the following we will define the Bernstein dimension and holonomic functions in more precise fashion.

We want to transport the main ideas without using the heavy algebraic machinery of filtrations, graded modules, etc. We try to keep this part as simple as possible, but without becoming faulty. The price we pay is that the following statements are not as general as they could be. On our way to holonomy we use two shortcuts: The first shortcut is not to talk about modules but only about ideals. To any left ideal I in A_d we can associate the left A_d-module A_d/I, so we are only dealing with the special case that interests us. The second shortcut consists in leaving away the definitions of filtration, filtered modules, and graded modules, and instead going directly to the definition of Hilbert polynomial and Bernstein dimension. A short remark for the algebraists: The Bernstein dimension of an A_d-module is the dimension of the graded associated module with respect to the Bernstein filtration (which filters along the total degree of both the x_i and D_{x_i}).

Let I be a left ideal in $A_d(\mathbb{K})$. Then analogously to commutative rings A_d/I is isomorphic to a $\mathbb{K}$-vector space whose basis is constituted by the monomials that cannot be reduced by I. Let $(A_d/I)_{\leq s}$ denote the (finite-dimensional) $\mathbb{K}$-vector space that has as its basis only monomials with total degree less than or equal to s. In other words, $(A_d/I)_{\leq s} = (A_d/I) \cap (A_d)_{\leq s}$ where $(A_d)_{\leq s}$ denotes the set of all elements in A_d with total degree at most s. Then we can define the *Hilbert function* of the left ideal I to be

$$HF_I(s) := \dim_{\mathbb{K}}(A_d/I)_{\leq s}.$$

It turns out that there exists a polynomial $HP_I(s)$ (called the *Hilbert polynomial* of the left ideal I) with the property that

$$HP_I(s) = HF_I(s) \quad \text{for all integers } s \geq s_0$$

for some $s_0 \in \mathbb{N}$. We want to call the degree of this polynomial the *Bernstein dimension* of I. Note that classically the Bernstein dimension is defined for left A_d-modules only, which agrees with our definition by looking at the module A_d/I.

Example 2.3. *Let's go back to univariate holonomic functions and revisit the left ideal $I = \langle xD_x - 3, D_x^4 \rangle$ from Example 2.1. Figure 2.3 shows monomials that can be reduced by I as solid dots whereas all monomials that cannot be reduced by I and hence form the $\mathbb{K}$-basis of A_1/I are depicted as open circles.*

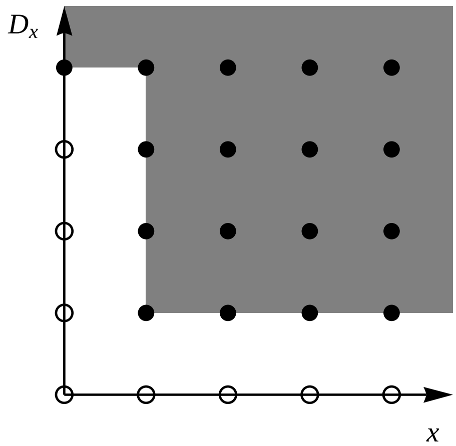

Figure 2.1: Shape of the annihilator of x^3.

The first values of $HF_I(s)$ are listed in the following table:

s	0	1	2	3	4	5	6	...
$HF_I(s)$	1	3	5	7	8	9	10	...

Thus the Hilbert polynomial is $s + 4$ and it agrees with the Hilbert function for $s \geq 3$; hence the Bernstein dimension of I is 1. Note that omitting the second ideal generator D_x^4 does not change the Bernstein dimension (the Hilbert polynomial in this case would be $2s + 1$).

We have now prepared the stage for Joseph Bernstein's (his name is sometimes spelled I. N. Bernshtein) celebrated result.

Theorem 2.4 (Bernstein's inequality)**.** *If I is a proper left ideal in A_d then the Bernstein dimension of I is greater than or equal to d.*

As the definition of Bernstein dimension before, this statement can be formulated in a more general fashion, then addressing finitely generated left A_d-modules. The general version and its proof can be found in Bernstein's original paper [16, Theorem 1.3] or in the nice introductory book of Coutinho [30, Chapter 9, Theorem 4.2].

Since they play an important rôle in our work, we want to investigate in more detail some properties of left ideals in the Weyl algebra; in particular the fact that there are no zero-dimensional proper left ideals. This is a consequence of Bernstein's inequality (Theorem 2.4). Therefore, instead of a rigorous proof, we try to give some intuition about this fact (note also that Example 2.3 agrees with Theorem 2.4).

Example 2.5. *From commutative algebra we are very much used to zero-dimensional ideals, e.g., the maximal ideals in* $\mathbb{K}[x, y]$ *each of which is generated by* $\langle x - c_1, y - c_2\rangle$ *for some constants* $c_1, c_2 \in \mathbb{K}$. *If we try to construct such an ideal in* A_1, *i.e.,* $I = {}_{A_1}\langle x - c_1, D_x - c_2\rangle$, *we find that also 1 will be contained in this ideal and hence it is the whole ring:*

$$\begin{aligned} D_x \cdot (x - c_1) - x \cdot (D_x - c_2) &= xD_x + 1 - c_1 D_x - xD_x + c_2 x \\ &= 1 - c_1 c_2 + c_2 c_1 = 1 \pmod{I} \end{aligned}$$

Or, in other words, we have shown by the above calculation that the given polynomials do not form a left Gröbner basis since their S-polynomial does not reduce to 0. Also observe that the 1 that survives in the end comes exactly from the commutation relation of the Weyl algebra.

This example illustrates that the ideal structure in the Weyl algebra is quite different than in a commutative polynomial ring. Note also that there are no proper two-sided ideals in the Weyl algebra (in other words A_d is *simple*).

Definition 2.6. *We want to call a left ideal* I *in the Weyl algebra* A_d holonomic *if* $I = A_d$ *or if the Bernstein dimension of* I *equals* d, *i.e., if the Bernstein dimension is as small as possible (Theorem 2.4). A function* $f(x_1, \ldots, x_d)$ *(or any "object" on which the Weyl algebra* A_d *can act) is called a* holonomic function *with respect to the variables* $x_1, \ldots, x_d$ *if there exists a holonomic left ideal in* A_d *that annihilates* f.

The notion of *holonomic ideal* is a slight abuse of mathematical language and can lead to confusion. The reason is that an ideal in A_d gives rise to an A_d-module in two ways: First we can consider the elements of I as the carrier set of the module, or second we can take the quotient A_d/I. The holonomy of an ideal then coincides with holonomy in the D-module sense if the second option is taken.

If one deals with partial differential equations rather than operators, then often also the notion *holonomic system* is used. In this context one speaks of the system as being *maximally overdetermined*, in the sense that there are as many linear partial differential equations with polynomial coefficients as possible.

A nice property of holonomic ideals that is crucial for our work is the so-called *elimination property*. We will later see why it is so important: it justifies the termination of many algorithms in Chapter 3.

Theorem 2.7. *Let I be a holonomic ideal in $A_d(\mathbb{K})$. Then for any subset of $d+1$ elements among the generators $\{x_1, \ldots, x_d, D_{x_1}, \ldots, D_{x_d}\}$ there exists a nonzero operator in I that involves only these $d+1$ generators and is free of the remaining $d-1$ generators (we will also say that we can eliminate these $d-1$ generators).*

Proof. We follow the proof given by Zeilberger in [93] to whom it was shown by Bernstein himself. For an arbitrary but fixed $(d+1)$-subset of the generators $\{x_1, \ldots, x_d, D_{x_1}, \ldots, D_{x_d}\}$, let $\tilde{A}$ denote the subalgebra of A_d that is generated by those. We study the mapping $\varphi : \tilde{A} \to A_d/I, P \mapsto P \mod I$. From the definition it is clear that $\dim_{\mathbb{K}}(\tilde{A})_{\leq s} = \binom{s+d+1}{d+1} = O(s^{d+1})$ because these are just all exponent vectors $\boldsymbol{\alpha} \in \mathbb{N}^{d+1}$ with $|\boldsymbol{\alpha}| \leq s$. On the other hand the holonomy of I implies that $\dim_{\mathbb{K}}(A_d/I)_{\leq s} = O(s^d)$. Hence there exists an integer $t > 0$ so that

$$\dim_{\mathbb{K}}(\tilde{A})_{\leq t} > \dim_{\mathbb{K}}(A_d/I)_{\leq t}.$$

So if we restrict φ to $(\tilde{A})_{\leq t}$ it will be a linear map from a $\mathbb{K}$-vector space of higher dimension to one with a lower dimension. Therefore $\ker \varphi$ is nontrivial and its elements are the desired operators. □

An immediate consequence of Theorem 2.7 is that for a given holonomic ideal we can find ordinary differential equations in it, namely for each D_{x_i} we can eliminate the remaining operators $D_{x_j}, j \neq i$. From this fact it follows that a holonomic function can be uniquely defined by giving its holonomic annihilating ideal plus finitely many initial values. Thus only a finite amount of information is needed to completely describe a holonomic function, as pointed out by Zeilberger [93]. In most cases, however, we are interested in eliminating some of the x_i as we will see in Chapter 3.

As we pointed out, the attribute holonomic can be used in any class of objects for which the differentiation is explained, e.g., for rational functions, formal power series, C^∞-functions, analytic functions, and distributions. Moreover we can think of calling also other objects holonomic as soon as we can act on them with the Weyl algebra (this action then has to be explained). As an example we want to have a look at sequences, and in the first place it is not clear how a differential operator should act on a sequence. Let $(a(\boldsymbol{n}))_{\boldsymbol{n}\in\mathbb{N}^d} \in \mathbb{K}^{\mathbb{N}^d}$ be a sequence in the variables $\boldsymbol{n} = n_1, \ldots, n_d$. We define the action of an operator of the Weyl algebra A_d by the actions of its generators:

$$\left.\begin{array}{rcl} D_{x_i} \bullet a(\boldsymbol{n}) & := & (n_i+1)a(n_1,\ldots,n_{i-1},n_i+1,n_{i+1},\ldots,n_d) \\ x_i \bullet a(\boldsymbol{n}) & := & a(n_1,\ldots,n_{i-1},n_i-1,n_{i+1},\ldots,n_d). \end{array}\right\} \quad (2.5)$$

Example 2.8. *With this definition we can argue that the operator* $4x^2D_x - xD_x + 2x - 1 \in A_1$ *annihilates the univariate sequence of Catalan numbers* $C_n = \frac{1}{n+1}\binom{2n}{n}$*. Indeed if we apply it to the sequence* C_n *according to the above rules we get*

$$4(n-1)C_{n-1} - nC_n + 2C_{n-1} - C_n = (4n-2)C_{n-1} - (n+1)C_n$$

which is nothing else than the well-known recurrence for the Catalan numbers.

Upon closer inspection of (2.5) we perceive that this definition is not at random but mirrors an ulterior motive: The action on a sequence just corresponds to the action on its (multivariate) generating function $\sum a(\boldsymbol{n})\boldsymbol{x}^{\boldsymbol{n}}$. Therefore we could equivalently define a sequence to be holonomic if and only if its generating function is holonomic. Since the action of differential operators on sequences is not very intuitive it is more convenient to translate them into the shift algebra via $D_{x_i} = (n_i+1)S_{n_i}$ and $x_i = S_{n_i}^{-1}$ (we can always get rid of the negative powers of shift operators by multiplying through). The elimination property carries directly over to the shift algebra by considering the Euler operator $\theta_{x_i} = x_iD_{x_i}$ that translates to $\theta_{x_i} = n_i$. This reasoning generalizes to the mixed setting where both discrete and continuous variables are involved. Finally we want to mention that also for q-holonomic functions a corresponding theory has been developed [75].

Using the theory of D-modules it can be proven that holonomic functions share a couple of nice closure properties. If f and g are two holonomic functions then so are $f+g$ and $f \cdot g$. Furthermore if $f(x_1,\ldots,x_d)$ is a holonomic function then also $f(x_1,\ldots,x_{d-1},c), c \in \mathbb{K}$, $\int f(x_1,\ldots,x_d)\,\mathrm{d}x_d$, and $\int_a^b f(x_1,\ldots,x_d)\,\mathrm{d}x_d$ are holonomic functions (if they are defined at all). Similar things can be said about the closure properties of holonomic sequences with the integral being replaced by the summation quantifier. Unfortunately it is not so simple to execute these closure properties algorithmically using the representation with holonomic ideals. Instead we will use the ∂-finite representation (see the next section) to execute closure properties on the set of functions that interest us. For that reason we do not want to go into detail here, in particular we omit all the proofs of the above statements (they can be found for example in [93]).

A collection of functions and sequences that are holonomic are displayed in Figure 2.3 (see page 41).

2.3 ∂-finite functions

In order to specify the class of functions that we are mostly going to deal with in practice, we first have to introduce another kind of noncommutative operator algebra, named after the Norwegian mathematician Øystein Ore who studied noncommutative polynomials [63] of this type. These algebras serve as a unifying framework to represent differential and difference equations as well as mixed ones (the Weyl algebra is just a special instance of them). Additionally they allow more flexibility in handling the coefficients, for example for treating differential equations with rational instead of polynomial coefficients (this allows us to divide out polynomial contents).

Ore algebras

We start with the space of functions that we want to operate on and denote it by $\mathcal{F}$. Again let $\mathbb{K}$ denote the ground field which means that we can see $\mathcal{F}$ as a $\mathbb{K}$-algebra of functions. We assume that $\mathcal{F}$ is equipped with certain $\mathbb{K}$-linear endomorphisms whose properties we want to reproduce with the following algebraic construction. If for example the $\mathbb{K}$-linear endomorphisms in question are derivations, i.e., if they are satisfying the Leibniz law, then $\mathcal{F}$ is also called a differential field or differential algebra. In this case we would like to create an algebra of differential operators.

An *Ore algebra* $\mathbb{O}$ is a skew polynomial ring, also called an Ore polynomial ring, that is obtained by applying Ore extensions to some base ring. The elements of $\mathbb{O}$ are interpreted as operators that act on the functions in $\mathcal{F}$. In order to specify the coefficients of the skew polynomial ring we first fix a $\mathbb{K}$-algebra $\mathbb{A}$, the base ring, which has to be a subalgebra of $\mathcal{F}$. Since the elements of $\mathbb{A}$ will be part of the operator algebra $\mathbb{O}$ we have to define their action on elements of $\mathcal{F}$; this is achieved in a natural way by specifying that an element $a \in \mathbb{A}$ acts on $f \in \mathcal{F}$ as the operator "multiplication by a", i.e., $a \bullet f := af$.

Every Ore extension is based on a $\mathbb{K}$-linear map $\delta : \mathcal{F} \to \mathcal{F}$ so that $\delta(kf + g) = k\delta(f) + \delta(g)$ for all $f, g \in \mathcal{F}$ and $k \in \mathbb{K}$. Furthermore, δ has to be a σ-derivation at the same time which means that it has to fulfill the *skew Leibniz law*

$$\delta(fg) = \sigma(f)\delta(g) + \delta(f)g \quad \text{for all } f, g \in \mathcal{F} \tag{2.6}$$

where σ is some injective $\mathbb{K}$-linear endomorphism of $\mathcal{F}$ (additive and multiplicative), i.e.,

$$\sigma(kf+g) = k\sigma(f)+\sigma(g) \quad \text{and} \quad \sigma(fg) = \sigma(f)\sigma(g) \quad \text{for all } f, g \in \mathcal{F}, k \in \mathbb{K}.$$

An Ore extension adds a new symbol ∂ to the base ring $\mathbb{A}$ and hence yields a skew polynomial ring that is denoted by $\mathbb{O} = \mathbb{A}[\partial; \sigma, \delta]$ and whose elements are polynomials in ∂ with coefficients in $\mathbb{A}$. The addition in $\mathbb{O}$ is just the usual one but the multiplication is defined by associativity via the commutation rule

$$\partial a = \sigma(a)\partial + \delta(a) \quad \text{for all } a \in \mathbb{A}$$

(for which to make sense we have to claim that σ and δ can be restricted to $\mathbb{A}$). Note that in contrast to the Weyl algebra the noncommutativity is now between the "variables" of the polynomial ring and its coefficients (see also Section 4.8). The injectivity of σ ensures the nice property that $\deg_\partial(pq) = \deg_\partial(p) + \deg_\partial(q)$ for skew polynomials $p, q \in \mathbb{O}$. Further it is possible to perform right Euclidean division. If additionally σ is invertible, then also left Euclidean division can be done. More details on the properties of such skew polynomial rings can be found in the instructive introduction by Bronstein and Petkovšek [21]. The process of adding Ore extensions can be iterated to get $\mathbb{A}[\partial_1; \sigma_1, \delta_1][\partial_2; \sigma_2, \delta_2] \cdots$, whereat we assume that $\partial_i\partial_j = \partial_j\partial_i$ and $\sigma_i\delta_j = \delta_j\sigma_i$ for all i and j.

The last missing step is to define how the new symbol ∂ should act on the functions in $\mathcal{F}$. Depending on what operation on $\mathcal{F}$ one wants to represent, one defines either $\partial \bullet f := \delta(f)$ or $\partial \bullet f := \sigma(f)$ (the latter option is usually chosen when $\delta = 0$). By means of the action $\bullet : \mathbb{O} \times \mathcal{F} \to \mathcal{F}$ our function space $\mathcal{F}$ turns into an $\mathbb{O}$-module.

As examples for making the above abstract definitions clearer, we present the two Ore extensions that we will mainly use:

- $\sigma(f) = f$ and $\delta(f) = \frac{\mathrm{d}f}{\mathrm{d}x}$: The action of the new symbol ∂ is defined to be $\partial \bullet f := \delta(f)$ and with $\mathbb{A} = \mathbb{K}[x]$ we get the first Weyl algebra A_1 (where we tacitly assume that the univariate polynomial ring $\mathbb{K}[x]$ is a subalgebra of $\mathcal{F}$). In contrast if we set $\mathbb{A} = \mathbb{K}(x)$ then we get the algebra of linear ordinary differential equations with rational coefficients; this can be interpreted as the localization of A_1 with respect to $\mathbb{K}[x]$.

- $\sigma(f) = f|_{n\to n+1}$ and $\delta(f) = 0$: In this case the action of ∂ is defined to be $\partial \bullet f := \sigma(f)$. With $\mathbb{A} = \mathbb{K}[n]$ we get the first shift algebra.

Typically $\mathbb{A} = \mathbb{K}(v_1, \ldots, v_{j-1})[v_j, \ldots, v_d]$ where both extreme cases $j = 1$ and $j = d$ may be attained. In these cases we speak of a polynomial or rational Ore algebra respectively.

Definition of ∂-finite functions

The notion of ∂-finiteness is the main ingredient for most of the algorithms that will be presented in Chapter 3. Roughly speaking a function is called ∂-finite if all its "derivatives" span a finite-dimensional vector space over the rational functions (in this context we use the term "derivative" with a more general meaning and let it refer to the application of any operator). Whenever dealing with ∂-finite functions we work in rational Ore algebras.

Let $\mathbb{O} = \mathbb{A}[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}] = \mathbb{A}[\partial_1; \sigma_1, \delta_1] \cdots [\partial_d; \sigma_d, \delta_d]$ be an Ore algebra with $\mathbb{A}$ being a field (typically a rational function field). A left ideal $I \subseteq \mathbb{O}$ is called a *∂-finite ideal w.r.t.* $\mathbb{O}$ if $\dim_{\mathbb{A}}(\mathbb{O}/I) < \infty$, i.e. the $\mathbb{A}$-vector space $\mathbb{O}/I$ is of finite dimension. A function f is called *∂-finite w.r.t.* $\mathbb{O}$ if there exists a ∂-finite ideal in $\mathbb{O}$ that annihilates f. Often we just speak of f being a ∂-finite function meaning that there is an appropriate Ore algebra with respect to which f is ∂-finite. The ∂ herein is just a generic symbol and does not refer to a concrete Ore operator.

Let now $I \subseteq \mathbb{O}$ denote an annihilating ∂-finite ideal for the function f. We denote the set of all "derivatives" of f by $\mathbb{O}\bullet f := \{P\bullet f \mid P \in \mathbb{O}\}$. Due to the fact that $\mathbb{O}/I \cong \mathbb{O} \bullet f$ we can say that f is ∂-finite if all its "derivatives" constitute a finite-dimensional $\mathbb{A}$-vector space. For this statement to make sense we additionally have to make sure that the function f itself can be seen as element of an $\mathbb{A}$-vector space. An instance where this fact shall prevent us from declaring a function to be ∂-finite will be given in Example 2.13. For the moment we want to make the idea of ∂-finiteness demonstrative by looking at a very basic example.

Example 2.9. *We consider the function* $f(x, y) = \sin\left(\frac{x+y}{x-y}\right)$ *in two continuous variables. It is natural to act with* D_x *and* D_y *on that function, e.g.,*
$D_x^2 D_y \bullet \sin\left(\frac{x+y}{x-y}\right) =$

$$\frac{8\,(-2xy - y^2)}{(x-y)^5}\sin\left(\frac{x+y}{x-y}\right) + \frac{4\,(x^3 - 5xy^2 + 2y^3)}{(x-y)^6}\cos\left(\frac{x+y}{x-y}\right)$$

It is obvious that all partial derivatives of f with respect to x and y are of the form

$$r_1(x,y)\sin\left(\frac{x+y}{x-y}\right)+r_2(x,y)\cos\left(\frac{x+y}{x-y}\right)$$

where r_1 and r_2 are rational functions in $\mathbb{Q}(x,y)$. But this means that the derivatives of f span a two-dimensional $\mathbb{Q}(x,y)$-vector space. Hence f is a ∂-finite function with respect to the Ore algebra $\mathbb{Q}(x,y)[D_x;1,D_x][D_y;1,D_y]$.

For holonomic functions we have seen that by Theorem 2.7 there exists an ordinary linear differential equation for each variable. A similar statement holds for ∂-finite ideals.

Proposition 2.10. *A left ideal $I \subseteq \mathbb{O} = \mathbb{A}[\partial_1;\sigma_1,\delta_1]\cdots[\partial_d;\sigma_d,\delta_d]$ is ∂-finite if and only if I contains a rectangular system, i.e., $\{P_1(\partial_1),\dots,P_d(\partial_d)\} \subseteq I$. By that we mean that P_i depends only on the Ore operator ∂_i and contains none of the others.*

Proof. One direction is obvious: If we have a rectangular system and consider the left ideal I that is generated by its elements, then the dimension $\dim_{\mathbb{A}} \mathbb{O}/I \leq \prod_{i=1}^d \deg_{\partial_i} P_i < \infty$ and hence I is ∂-finite.

On the other hand, assume that $I \subseteq \mathbb{O}$ is ∂-finite with $\dim_{\mathbb{A}} \mathbb{O}/I = m$. We consider the sequence of power products $1, \partial_1, \partial_1^2, \dots$ each of which can be reduced to normal form modulo I. Since these normal forms are elements in a m-dimensional $\mathbb{A}$-vector space, we find a linear dependence at the latest when we go up until ∂_1^m. This linear dependence is nothing else but an element in I that involves only ∂_1 and none of the $\partial_i, i > 1$. Doing the same game for $\partial_2,\dots,\partial_d$ yields a rectangular system. □

The nice thing about ∂-finite functions is that again we have to specify only finitely many initial values in order to have a complete description of a concrete function (viewed as a formal power series). It may however happen that the leading coefficients of the annihilating operators introduce some singularities in which case more, possibly infinitely many, initial values have to be given (see also Section 7.3).

Example 2.11. *We want to study the Struve function $\boldsymbol{H}_n(z)$ that is a solution of the inhomogeneous second-order Bessel differential equation ([6, 12.1.1]):*

$$z^2\boldsymbol{H}_n''(z)+z\boldsymbol{H}_n'(z)+(z^2-n^2)\boldsymbol{H}_n(z)=\frac{4\left(\frac{z}{2}\right)^{n+1}}{\sqrt{\pi}\Gamma\left(n+\frac{1}{2}\right)}$$

We homogenize this equation by first constructing an annihilating operator $zD_z - n - 1$ *for the inhomogeneous part (which is easy since it is hyperexponential with respect to* z*) and by multiplying it to the differential equation* $z^2D_z^2 + zD_z + z^2 - n^2$ *from the left:*

$$P_1(D_z) = z^3D_z^3 - (n-2)z^2D_z^2 - \left(n^2+n-z^2\right)zD_z + (n^3+n^2-nz^2+z^2)$$

hence is an annihilating operator for $\boldsymbol{H}_n(z)$*. Similarly we can look up an inhomogeneous recurrence [6, 12.1.9] for the Struve function*

$$\boldsymbol{H}_{n-1}(z) + \boldsymbol{H}_{n+1}(z) = \frac{2n}{z}\boldsymbol{H}_n(z) + \frac{\left(\frac{z}{2}\right)^n}{\sqrt{\pi}\Gamma\left(n+\frac{3}{2}\right)}$$

which again can be made homogeneous in the same manner:

$$\begin{aligned} P_2(S_n) &= ((2n+5)S_n - z)\cdot\left(zS_n^2 - (2n+2)S_n + z\right) = \\ &= (2n+5)zS_n^3 - (4n^2+18n+z^2+20)S_n^2 + (4n+7)zS_n - z^2 \end{aligned}$$

(we multiplied the original equation by z *in order to clear denominators). Now* P_1 *and* P_2 *form a rectangular system which proves that* $\boldsymbol{H}_n(z)$ *is a ∂-finite function with respect to the Ore algebra* $\mathbb{O} = \mathbb{Q}(n,z)[S_n;S_n,0][D_z;1,D_z]$*. If we additionally provide* $3\cdot 3 = 9$ *initial values, this rectangular system describes the Struve function completely. But it is not a Gröbner basis (once again we have to suppress our intuition from commutative algebra that bases on Buchberger's product criterion). The Gröbner basis for the left ideal generated by* P_1 *and* P_2 *with respect to total degree order is*

$$\begin{aligned} \{&z^2D_z^2 + (-2nz-z)S_n - 2nzD_z + n^2 + n + z^2, \\ &zS_nD_z + (n+1)S_n - z, \\ &(2nz+3z)S_n^2 - (4n^2+10n+z^2+6)S_n - z^2D_z + 3nz + 3z\}. \end{aligned}$$

From the leading monomials D_z^2*,* S_nD_z*, and* S_n^2 *we can read off that there are 3 monomials* $1, D_z, S_n$ *under the staircase. Hence only 3 initial values suffice to determine a power series expansion around* $z = 0$ *for all* $n \in \mathbb{N}$ *in a unique way:*

$$\boldsymbol{H}_0(0) = \boldsymbol{H}_1(0) = 0 \quad \text{and} \quad \boldsymbol{H}_0'(0) = \frac{2}{\pi}.$$

Example 2.12. *In contrast to the previous example, Stirling numbers are not ∂-finite. Although the Stirling numbers of the first kind for example are killed by the mixed recurrence operator* $S_mS_n + mS_n - 1$*, there are no pure recurrences, neither in the first nor in the second parameter, for them. This means they do not possess a rectangular system and hence are not ∂-finite.*

Let us illustrate why it is so important that any object that we want to consider to be ∂-finite must be an element of an appropriate vector space. In the previous example, the Stirling numbers are in principal eligible for applying the definition of ∂-finite, but in the end there are simply not enough relations for them (what to do in such cases has recently been described in [27], see Section 2.6). In contrast to that we will now discuss an example which is not ∂-finite because the definition of ∂-finiteness cannot be applied.

Example 2.13. *The Kronecker delta $\delta_{m,n}$ is defined to be 1 if $m = n$ and 0 otherwise. Since m and n are discrete variables we introduce the Ore algebra $\mathbb{O} = \mathbb{Q}(m,n)[S_m; S_m, 0][S_n; S_n, 0]$. It is easy to verify that $\delta_{m,n}$ is annihilated by $(m-n+1)S_m+n-m$ and $(-m+n+1)S_n+n-m$. These two operators are a rectangular system and hence generate a ∂-finite left ideal in $\mathbb{O}$ (they even form a Gröbner basis), so one could be tempted to declare $\delta_{m,n}$ to be a ∂-finite function. Now observe that $\delta_{m,n}$ is also annihilated by the polynomial $m-n$, but the left ideal generated by $m-n$ is the whole ring $\mathbb{O}$ (we can remove a polynomial content in m and n). Hence in the ∂-finite setting we cannot distinguish between $\delta_{m,n}$ and the bivariate sequence that is identically 0. The reason is that we cannot interpret $\delta_{m,n}$ as an element of a $\mathbb{Q}(m,n)$-vector space.*

Closure properties

Similar to the holonomic functions, the class of ∂-finite functions shares some nice closure properties. Additionally these can be executed effectively and algorithmically in a relatively simple manner. The ∂-finite functions are closed under operator application, sum, product, algebraic substitutions for continuous variables, and rational-linear substitutions for discrete variables. In the following we prove that these closure properties indeed hold and we try to formulate the proofs in a way that gives the corresponding algorithms at the same time.

The algorithms for performing ∂-finite closure properties follow a similar principle as the celebrated FGLM algorithm [33] (named after its inventors Faugère, Gianni, Lazard, and Mora). For that reason we want to shortly describe this algorithm. The FGLM algorithm is designed for transforming a given Gröbner basis G_1 of a zero-dimensional ideal in $\mathbb{K}[\boldsymbol{x}]$ into a Gröbner basis G_2 for the same ideal with respect to a different term order $\prec_2$. It works by going systematically through the monomials of $\mathbb{K}[\boldsymbol{x}]$ starting with the set $T = \{\boldsymbol{x}^{\boldsymbol{0}}\} = \{1\}$: In each step we choose (and afterwards delete) the $\prec_2$-minimal monomial $\boldsymbol{x}^{\gamma}$ from T such that it is not divisible by the leading monomial of some element of G_2 that we might already have found. It now can be decided whether $\boldsymbol{x}^\gamma$ is the leading monomial of some new element of G_2 or whether it belongs to the set of monomials which cannot be reduced by G_2. For this purpose $\boldsymbol{x}^\gamma$ is reduced with G_1 to normal form representation (which is an element of the finite-dimensional $\mathbb{K}$-vector space $\mathbb{K}[\boldsymbol{x}]/\langle G_1\rangle$) and afterwards it is checked whether there is a linear dependence between this normal form and all normal forms that correspond to monomials for which we found earlier that they are under the stairs of G_2. If they are linearly dependent, this means that $\boldsymbol{x}^\gamma$ is the leading monomial of an element of G_2 (which now is given by exactly this linear dependence). On the other hand if all these normal forms are linearly independent then this indicates that $\boldsymbol{x}^\gamma$ cannot be reduced by G_2 and hence lies under the staircase of G_2. In this case we have to continue our search for leading monomials in all directions which means that we add the elements $x_1\boldsymbol{x}^\gamma, x_2\boldsymbol{x}^\gamma, \ldots$ to T. This procedure is repeated until G_2 is complete (which can easily be seen, e.g., by equating the vector space dimensions $\dim_{\mathbb{K}} \mathbb{K}[\boldsymbol{x}]/\langle G_1\rangle = \dim_{\mathbb{K}} \mathbb{K}[\boldsymbol{x}]/\langle G_2\rangle$). Having an idea of how the FGLM algorithm works will make the rest of this section much better understandable. The algorithm is displayed in Figure 2.2.

```
Input: Gröbner basis G_1 ⊂ K[x_1, ..., x_d], term order ≺_2
Output: Gröbner basis G_2 of ⟨G_1⟩ with respect to ≺_2
-----------------------------------------------------------
T := {1}, G_2 := {}, j := 1
while T ≠ {}
   T := T \ {t ∈ T | ∃g ∈ G_2 such that lm(g) divides t}
   t_j := min_{≺_2} T
   T := T \ {t_j}
   NF_j := normal form of t_j with respect to G_1
   if ({NF_i | 1 ≤ i ≤ j} are linearly dependent) then
      let c_i ∈ K (not all zero) with c_1 NF_1 + ··· + c_j NF_j = 0
      G_2 := G_2 ∪ {c_1 t_1 + ··· + c_j t_j}
   else
      T := T ∪ {x_i t_j | 1 ≤ i ≤ d}
      j := j + 1
return G_2
```

Figure 2.2: FGLM Algorithm

Theorem 2.14. *Let f be a ∂-finite function with respect to the Ore algebra $\mathbb{O} = \mathbb{A}[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}] = \mathbb{A}[\partial_1; \sigma_1, \delta_1] \cdots [\partial_d; \sigma_d, \delta_d]$. Then for any operator $P \in \mathbb{O}$ also $g = P \bullet f$ is a ∂-finite function with respect to $\mathbb{O}$.*

Proof. Since any "derivative" of g is also a "derivative" of f, or in other words $\mathbb{O} \bullet g = (\mathbb{O} \cdot P) \bullet f \subseteq \mathbb{O} \bullet f$, it is immediately clear that also g is ∂-finite. Moreover given a Gröbner basis $G \subset \mathbb{O}$ for an annihilating ∂-finite ideal of f, a Gröbner basis $\widetilde{G}$ for an annihilating ideal of g can be computed by an adjusted version of the FGLM algorithm. The monomials $\boldsymbol{\partial}^\gamma$ are tested in the same systematic way, whether they lie under the stairs of $\widetilde{G}$ or are the leading monomial of an element of $\widetilde{G}$. The only change takes place in the reduction step: instead of reducing the monomial $\boldsymbol{\partial}^\gamma$ one computes the normal form of $P \cdot \boldsymbol{\partial}^\gamma$ with respect to G. □

Example 2.15. *We consider the Hankel function of the first kind $H_n^{(1)}(x)$; sometimes it is also called Bessel function of the third kind, because it is a linear combination of Bessel functions of the first and second kind. We work in the Ore algebra $\mathbb{O} = \mathbb{Q}(n,x)[S_n; S_n, 0][D_x; 1, D_x]$ and choose a total degree order that breaks ties via $S_n > D_x$. The reduced Gröbner basis for the annihilator of $H_n^{(1)}(x)$ is*

$$\{xS_n + xD_x - n, x^2D_x^2 + xD_x + (x^2 - n^2)\}.$$

We want to compute the reduced Gröbner basis $\widetilde{G}$ of an annihilating ideal for $H_{n+1}^{(1)}(x) + H_n^{(1)}(x)$, thus $P = S_n + 1$. We start with the monomial 1 and reduce $(S_n + 1) \cdot 1$ to

$$S_n + 1 - \frac{1}{x}(xS_n + xD_x - n) = -D_x + \frac{n}{x} + 1.$$

Next we add the monomials S_n and D_x to the set of test monomials T. The minimal element in T is D_x and we reduce $(S_n + 1)D_x$ to $\frac{1}{x}(n + x + 1)D_x - \frac{1}{x^2}(n^2 + n - x^2)$. We find that the two normal forms are linearly independent, hence the monomials 1 and D_x will not be reducible by $\widetilde{G}$. Thus we add the monomials S_nD_x and D_x^2 to T. Next we take the smallest monomial S_n and reduce $(S_n + 1)S_n$ to $-\frac{1}{x}(2n + x + 2)D_x + \frac{1}{x^2}(2n^2 + nx + 2n - x^2)$. Now the three normal forms computed so far are linearly dependent:

$$-(2n^2 + 2nx + 2n + x)\begin{pmatrix} -1 \\ \frac{n}{x} + 1 \end{pmatrix} + 2x(n + x + 1)\begin{pmatrix} \frac{n+x+1}{x} \\ -\frac{n^2+n-x^2}{x^2} \end{pmatrix} +$$
$$x(2n + 2x + 1)\begin{pmatrix} -\frac{2n+x+2}{x} \\ \frac{2n^2+nx+2n-x^2}{x^2} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Hence $x(2n + 2x + 1)S_n + 2x(n + x + 1)D_x - (2n^2 + 2nx + 2n + x)$ is an element of $\widetilde{G}$. Last we have to care about the monomial D_x^2 from T (S_nD_x we will not have to consider since it is a multiple of the leading monomial S_n).

We find that there is a linear dependence between the first two normal forms (corresponding to 1 and D_x) and the normal form of $(S_n+1)D_x^2$. It delivers the second element of $\widetilde{G}$

$$x^2(2n+2x+1)D_x^2 + 2x(2n+x+1)D_x - (2n-2x+1)(n+x)(n+x+1).$$

which is now a complete Gröbner basis.

In the example we observe that the resulting Gröbner basis has the same number of monomials under its staircase as the input. Analyzing the algorithm reveals that the closure property of operator application never increases this dimension, but sometimes even reduces it. This is an important point and one should try to use this closure property whenever it is possible. If we used the closure property sum instead we would have ended up with a Gröbner basis of $\mathbb{Q}(n,x)$-dimension 4 under the staircase.

Theorem 2.16. *If f and g are ∂-finite functions with respect to some Ore algebra $\mathbb{O} = \mathbb{A}[\boldsymbol{\partial};\boldsymbol{\sigma},\boldsymbol{\delta}] = \mathbb{A}[\partial_1;\sigma_1,\delta_1]\cdots[\partial_d;\sigma_d,\delta_d]$, then $f+g$ is ∂-finite with respect to $\mathbb{O}$ as well. If additionally $\boldsymbol{\sigma}^{\boldsymbol{\alpha}}(f) \in \mathbb{O} \bullet f$ for $\boldsymbol{\alpha} \in \mathbb{N}^d$, then also $f \cdot g$ is ∂-finite with respect to $\mathbb{O}$.*

Proof. Since f is a ∂-finite function we know that we can rewrite any derivative $\boldsymbol{\partial}^{\boldsymbol{\gamma}} \bullet f, \boldsymbol{\gamma} \in \mathbb{N}^d$ as an $\mathbb{A}$-linear combination of $\{\boldsymbol{\partial}^{\boldsymbol{\alpha}} \bullet f \mid \boldsymbol{\alpha} \in U\}$ where $U \subset \mathbb{N}^d$ is a finite set of exponent vectors representing the monomials under the staircase of a ∂-finite annihilating ideal for f. Similarly every derivative of g can be expressed as an $\mathbb{A}$-linear combination of $\{\boldsymbol{\partial}^{\boldsymbol{\beta}} \bullet g \mid \boldsymbol{\beta} \in V\}$ for some other finite set $V \subset \mathbb{N}^d$.

In order to prove that $f+g$ is ∂-finite we rewrite an arbitrary derivative

$$\boldsymbol{\partial}^{\boldsymbol{\gamma}} \bullet (f+g) = \boldsymbol{\partial}^{\boldsymbol{\gamma}} \bullet f + \boldsymbol{\partial}^{\boldsymbol{\gamma}} \bullet g = \sum_{\boldsymbol{\alpha}\in U} a_{\boldsymbol{\alpha}}(\boldsymbol{\partial}^{\boldsymbol{\alpha}} \bullet f) + \sum_{\boldsymbol{\beta}\in V} b_{\boldsymbol{\beta}}(\boldsymbol{\partial}^{\boldsymbol{\beta}} \bullet g), \quad a_{\boldsymbol{\alpha}}, b_{\boldsymbol{\beta}} \in \mathbb{A}$$

from which it is clear that all derivatives of $f+g$ span a finite-dimensional vector space over $\mathbb{A}$; it is spanned by $\{\boldsymbol{\partial}^{\boldsymbol{\alpha}} \bullet f \mid \boldsymbol{\alpha} \in U\} \cup \{\boldsymbol{\partial}^{\boldsymbol{\beta}} \bullet f \mid \boldsymbol{\beta} \in V\}$ and hence its dimension is at most $|U|+|V|$.

A similar argument applies in the product case; applying the skew Leibniz law (2.6), any derivative $\boldsymbol{\partial}^{\boldsymbol{\gamma}} \bullet (f \cdot g), \boldsymbol{\gamma} \in \mathbb{N}^d$ can be rewritten as an $\mathbb{A}$-linear combination of products of derivatives of f and g:

$$\boldsymbol{\partial}^{\boldsymbol{\gamma}} \bullet (f \cdot g) = \sum_{\boldsymbol{\alpha},\boldsymbol{\beta}\in\mathbb{N}^d} \tilde{c}_{\boldsymbol{\alpha},\boldsymbol{\beta}} \cdot \big(\boldsymbol{\partial}^{\boldsymbol{\alpha}} \bullet f\big) \cdot \big(\boldsymbol{\partial}^{\boldsymbol{\beta}} \bullet g\big), \quad \tilde{c}_{\boldsymbol{\alpha},\boldsymbol{\beta}} \in \mathbb{A}. \tag{2.7}$$

For differential and shift operators this is trivially achieved by

$$\begin{aligned} D_x \bullet (f \cdot g) &= (D_x \bullet f) \cdot g + f \cdot (D_x \bullet g), \\ S_n \bullet (f \cdot g) &= (S_n \bullet f) \cdot (S_n \bullet g). \end{aligned}$$

In the general case we have to distinguish the cases $\partial_i \bullet f = \sigma_i(f)$ (which is fine) and $\partial_i \bullet f = \delta_i(f)$. In the latter we use that σ_i and δ_i commute:

$$\partial_i^n \bullet (f \cdot g) = \sum_{k=0}^{n} \binom{n}{k} \delta_i^{n-k}\big(\sigma_i^k(f)\big)\delta_i^k(g).$$

Equation (2.7) is then established by the condition $\sigma_i^k(f) \in \mathbb{O} \bullet f$.

But the derivatives of f and the derivatives of g themselves can be expressed as linear combinations of elements determined by U and V. So finally we get

$$\boldsymbol{\partial}^{\boldsymbol{\gamma}} \bullet (f \cdot g) = \sum_{\boldsymbol{\alpha} \in U} \sum_{\boldsymbol{\beta} \in V} c_{\boldsymbol{\alpha},\boldsymbol{\beta}} \cdot \big(\boldsymbol{\partial}^{\boldsymbol{\alpha}} \bullet f\big) \cdot \big(\boldsymbol{\partial}^{\boldsymbol{\beta}} \bullet g\big), \quad c_{\boldsymbol{\alpha},\boldsymbol{\beta}} \in \mathbb{A}.$$

Again it is now clear that the derivatives of $f \cdot g$ span a finite-dimensional $\mathbb{A}$-vector space whose dimension is at most $|U| \cdot |V|$. □

From the algorithmic point of view we proceed as follows. The input are two Gröbner bases for ∂-finite annihilating ideals of f and g in $\mathbb{O}$. They determine the sets $U, V \subset \mathbb{N}^d$ representing the monomials under the stairs of the respective Gröbner basis. We go through the monomials $\boldsymbol{\partial}^{\boldsymbol{\gamma}}$ in the same systematic way as in the FGLM algorithm. To each monomial $\boldsymbol{\partial}^{\boldsymbol{\gamma}}$ we have to compute a kind of "normal form". The normal form in the sum case corresponds to the vector

$$(a_{\boldsymbol{\alpha}_1}, a_{\boldsymbol{\alpha}_2}, \ldots, b_{\boldsymbol{\beta}_1}, b_{\boldsymbol{\beta}_2}, \ldots) \in \mathbb{A}^{|U|+|V|}.$$

In the product case the normal form is constituted by the coefficients $c_{\boldsymbol{\alpha},\boldsymbol{\beta}}$:

$$(c_{\boldsymbol{\alpha}_1,\boldsymbol{\beta}_1}, c_{\boldsymbol{\alpha}_1,\boldsymbol{\beta}_2}, \ldots, c_{\boldsymbol{\alpha}_2,\boldsymbol{\beta}_1}, c_{\boldsymbol{\alpha}_2,\boldsymbol{\beta}_2}, \ldots) \in \mathbb{A}^{|U|\cdot|V|}.$$

In order to compute these "normal forms", in particular in the last rewriting step, the reduction modulo the input Gröbner bases is used. Everything else is done exactly as in the FGLM algorithm.

The next closure property performs algebraic substitution of continuous variables. Although it is quite folklore, we want to state a basic fact that is needed later, in the following lemma.

Lemma 2.17. *Let $h(\boldsymbol{z})$ be a multivariate algebraic function, i.e., there exists a nonzero polynomial $p \in \mathbb{K}[h, \boldsymbol{z}]$ with $p(h(\boldsymbol{z}), \boldsymbol{z}) = 0$. Any derivative of $h(\boldsymbol{z})$ can be expressed as a polynomial in $h(\boldsymbol{z})$ with degree smaller than the degree of the minimal polynomial p.*

Proof. We differentiate the minimal polynomial p with respect to z_i:

$$\frac{\partial}{\partial z_i} p(h(\boldsymbol{z}), \boldsymbol{z}) = \frac{\partial p}{\partial h} \cdot \frac{\partial h}{\partial z_i} + \frac{\partial p}{\partial z_i} = 0.$$

Solving this equation for the derivative of h gives

$$\frac{\partial h}{\partial z_i} = -\frac{\partial p}{\partial z_i} \cdot \left(\frac{\partial p}{\partial h}\right)^{-1}.$$

After reducing the expression on the right-hand side modulo the minimal polynomial p (for the second factor we compute the modular inverse with the extended Euclidean algorithm we obtain the desired representation for $D_{z_i} \bullet h(\boldsymbol{z})$. Repeating this procedure iteratively, we get such a representation for arbitrary higher and mixed derivatives of $h(\boldsymbol{z})$. □

Theorem 2.18. *Let $f(\boldsymbol{x}, \boldsymbol{w})$ be a ∂-finite function with respect to the Ore algebra $\mathbb{O} = \mathbb{K}(\boldsymbol{x}, \boldsymbol{w})[\boldsymbol{D_x}; \boldsymbol{1}, \boldsymbol{D_x}][\boldsymbol{\partial_w}; \boldsymbol{\sigma_w}, \boldsymbol{\delta_w}]$ where $\boldsymbol{x} = x_1, \ldots, x_d$. Let further $h_1(\boldsymbol{z}), \ldots, h_d(\boldsymbol{z})$ be algebraic functions in $\boldsymbol{z} = z_1, \ldots, z_e$ which means that there are nonzero polynomials $p_1, \ldots, p_d \in \mathbb{K}[h, z_1, \ldots, z_e]$ such that $p_i(h_i(\boldsymbol{z}), z_1, \ldots, z_e) = 0$. Then the function $g(\boldsymbol{z}, \boldsymbol{w}) = f(h_1(\boldsymbol{z}), \ldots, h_d(\boldsymbol{z}), \boldsymbol{w})$ is ∂-finite w.r.t. the Ore algebra $\mathbb{O}' = \mathbb{K}(\boldsymbol{z}, \boldsymbol{w})[\boldsymbol{D_z}; \boldsymbol{1}, \boldsymbol{D_z}][\boldsymbol{\partial_w}; \boldsymbol{\sigma_w}, \boldsymbol{\delta_w}]$.*

Proof. We want to study the action of a differential operator on g. For $1 \leq i \leq e$, applying the chain rule we get

$$\begin{aligned} D_{z_i} \bullet g(\boldsymbol{z}, \boldsymbol{w}) &= (D_{x_1} \bullet f)\big(h_1(\boldsymbol{z}), \ldots, h_d(\boldsymbol{z}), \boldsymbol{w}\big) \cdot (D_{z_i} \bullet h_1(\boldsymbol{z})) + \cdots + \\ &\quad\; (D_{x_d} \bullet f)\big(h_1(\boldsymbol{z}), \ldots, h_d(\boldsymbol{z}), \boldsymbol{w}\big) \cdot (D_{z_i} \bullet h_d(\boldsymbol{z})). \end{aligned}$$

We can rewrite the derivatives of the algebraic functions $h_1(\boldsymbol{z}), \ldots, h_d(\boldsymbol{z})$ by Lemma 2.17 as polynomials in these respective functions. It is not difficult to see that we can throw other differential operators on the previous expression and after doing a similar rewriting plus reduction modulo the minimal polynomials p_i, we obtain an expression that involves some derivatives of f and each h_i occurs polynomially with powers smaller than the degree of its minimal polynomial p_i. Since f is ∂-finite we can rewrite all its derivatives as linear combinations of $U_1 \bullet f, \ldots, U_m \bullet f$ (where the U_i are the monomials under the staircase of the ∂-finite annihilating ideal of f). Summing up, we can express any arbitrary derivative of $g(\boldsymbol{w}, \boldsymbol{z})$ as a linear combination of

$$(U_i \bullet f)\big(h_1(\boldsymbol{z}), \ldots, h_d(\boldsymbol{z}), \boldsymbol{w}\big) \cdot h_1(\boldsymbol{z})^{j_1} \cdots h_d(\boldsymbol{z})^{j_d}$$

where $1 \leq i \leq m$ and $0 \leq j_l < \deg_h p_l$ for $1 \leq l \leq d$. Hence g is a ∂-finite function and its derivatives span a $\mathbb{K}(\boldsymbol{z}, \boldsymbol{w})$-vector space of dimension at most $m \prod_{l=1}^{d} \deg_h p_l$. □

We can execute the algebraic substitution algorithmically by again using an FGLM-like procedure. Since the output is supposed to be a Gröbner basis of some left ideal in $\mathbb{O}'$, we will go through the monomials of $\mathbb{O}'$. For each monomial that we have to consider we compute a normal form as described in the proof. Everything else is done as in the FGLM algorithm. The key point is to translate the action of an operator in $\mathbb{O}'$ on g to the action of an operator in $\mathbb{O}$ on f. This idea is even more exploited in the last closure property that we are going to discuss, the rational-linear substitution of discrete variables.

Theorem 2.19. *Let $f(\boldsymbol{k}, \boldsymbol{w})$ be a ∂-finite function with respect to the Ore algebra $\mathbb{O} = \mathbb{K}(\boldsymbol{k}, \boldsymbol{w})[\boldsymbol{S_k}; \boldsymbol{S_k}, \boldsymbol{0}][\boldsymbol{\partial_w}; \boldsymbol{\sigma_w}, \boldsymbol{\delta_w}]$ where $\boldsymbol{k} = k_1, \dots, k_d$. Then the result of the rational-linear substitution*

$$g(\boldsymbol{n}, \boldsymbol{w}) = f(c_1 + c_{1,1}n_1 + \dots + c_{1,e}n_e,\ \dots,\ c_d + c_{d,1}n_1 + \dots + c_{d,e}n_e, \boldsymbol{w})$$

where the c_i are arbitrary constants and the $c_{i,j}$ are rational numbers, is again ∂-finite with respect to the Ore algebra $\mathbb{O}' = \mathbb{K}(\boldsymbol{n}, \boldsymbol{w})[\boldsymbol{S_n}; \boldsymbol{S_n}, \boldsymbol{0}][\boldsymbol{\partial_w}; \boldsymbol{\sigma_w}, \boldsymbol{\delta_w}]$, provided that the annihilating relations for $f(\boldsymbol{k}, \boldsymbol{w})$ still hold when $k_1, \dots, k_d$ take (potentially non-integral) values implied by the c_i and $c_{i,j}$.

Proof. We study how shifts in the n_i translate to shifts in the k_j and therefore let the operator $S_{n_1}^{a_1} \cdots S_{n_e}^{a_e} \in \mathbb{O}'$ act on the substitution $k_j = c_j + c_{j,1}n_1 + \dots + c_{j,e}n_e$. The result will be

$$c_j + c_{j,1}(n_1 + a_1) + \dots + c_{j,e}(n_e + a_e) = k_j + \underbrace{\sum_{i=1}^{e} c_{j,i}a_i}_{=:s_j}.$$

Let t_j be the common denominator of the $c_{j,i}, 1 \le i \le e$. The quantity s_j then is a rational number with $s_j t_j \in \mathbb{Z}$. It is now clear that each shift of g translates to a shifted version of f of the form $f(\boldsymbol{k} + \boldsymbol{s}, \boldsymbol{w})$ with $\boldsymbol{s}$ being an element of the lattice generated by $(t_1, 0, \dots, 0), \dots, (0, \dots, 0, t_d)$. Since f is ∂-finite all instances $f(\boldsymbol{k} + \boldsymbol{s}, \boldsymbol{w})$ can be reduced to a finite set of such instances, its size being bounded by $\dim_{\mathbb{K}(\boldsymbol{k},\boldsymbol{w})}(\mathbb{O} \bullet f) \prod_{j=1}^{d} t_j < \infty$. Hence g is ∂-finite. □

We try to enlighten the argument of the above proof by an example. For sake of simplicity we choose a univariate one.

Example 2.20. *Given the ∂-finite function $f(k) = k!$ by its ∂-finite annihilating ideal $I = {}_{\mathbb{O}}\langle S_k - k - 1 \rangle$ with $\mathbb{O} = \mathbb{Q}(k)[S_k; S_k, 0]$, compute the ∂-finite annihilating ideal for $g(n) = f\left(\frac{n}{2}\right)$ with respect to $\mathbb{O}' = \mathbb{Q}(n)[S_n; S_n, 0]$. A* *monomial $S_n^a \in \mathbb{O}'$ translates in terms of the original function as $f\big(k+\frac{a}{2}\big)$. All such instances can be reduced with I to a linear combination of $f(k)$ and $f\big(k+\frac{1}{2}\big)$. The first two monomials under consideration, S_n^0 and S_n^1, translate to exactly these two basis elements. Finally S_n^2 translates to S_k whose normal form modulo I is $k+1$. At this point we find a linear dependence between the normal forms*

$$(-k-1)\begin{pmatrix}1\\0\end{pmatrix}+0\begin{pmatrix}0\\1\end{pmatrix}+\begin{pmatrix}k+1\\0\end{pmatrix}=\begin{pmatrix}0\\0\end{pmatrix}$$

which delivers (after substituting $k=\frac{n}{2}$ and clearing denominators) as result the annihilating operator $2S_n^2-n-2$.

2.4 Holonomic versus ∂-finite

We have now introduced two different classes of functions, namely holonomic and ∂-finite functions. This section gives a comparison of these two classes and answers questions like: what are the differences between holonomic and ∂-finite? Which functions lie in the intersection? Why do we need these two notions at all and why does it not suffice to consider just one of them?

An obvious difference concerns the kind of operators that can be treated. The definition of ∂-finiteness makes sense for all kinds of Ore operators whereas holonomy is defined for the differential setting in the first place, but can be extended to the shift setting using the loop way via the generating function and to q-calculus. Since we are basically interested only in these settings, the broader generality of ∂-finiteness is not exploited and negligible in our work.

Univariate functions

For functions in one variable, the definitions of holonomy and ∂-finiteness coincide provided that they are applicable. We can easily convince ourselves that this is the case: Let $f(x)$ be a ∂-finite function with respect to $\mathbb{O}=\mathbb{K}(x)[D_x;1,D_x]$; then there exists a homogeneous linear differential equation with coefficients in $\mathbb{K}(x)$ for $f(x)$. After clearing denominators we can consider the left ideal in the Weyl algebra A_1 generated by it. The Bernstein dimension is 1 (there is no other choice—dimension 2 happens only for the zero ideal) and hence $f(x)$ is holonomic. Conversely assume that $f(x)$ is a holonomic function and that it can be seen as a $\mathbb{K}(x)$-vector space element. Then any operator in the holonomic (and therefore nonzero) ideal for $f(x)$

involves the Ore operator D_x and witnesses that $f(x)$ is ∂-finite: there are no annihilating operators in the subalgebra $\mathbb{K}[x]$ because otherwise $f(x)$ could not be interpreted as a vector space element. A similar reasoning applies to the shift and the q-case.

Example 2.21. *The Dirac delta $\delta(x)$ distribution is a univariate example which is holonomic but not ∂-finite. As it is annihilated by a polynomial via $x\delta(x) = 0$ it cannot be interpreted as an element in a $\mathbb{K}(x)$-vector space.*

If we restrict the functions in question to formal power series then we do not have to care about this subtle difference the notions D-finite, holonomic, and ∂-finite are all equivalent.

Univariate sequences with finite support are trivially annihilated by a polynomial and therefore can not be considered to be ∂-finite. But definitely they are holonomic and also P-finite. For sequences with infinite support the notions P-finite, holonomic, and ∂-finite are all equivalent.

Differential setting

After having discussed the univariate situation, we now turn to the multivariate setting which we restrict for the moment to differential operators only. Doing so, one finds that holonomy and ∂-finiteness again coincide provided that the definitions are applicable. This result follows as a corollary from a deep theorem of Masaki Kashiwara [45].

Theorem 2.22. *Let A_d be the Weyl algebra in $\boldsymbol{x} = x_1, \dots, x_d$ and let $\mathbb{O}$ be the rational differential Ore algebra $\mathbb{K}(\boldsymbol{x})[\boldsymbol{D_x}; \mathbf{1}, \boldsymbol{D_x}]$. A left ideal $I \subseteq \mathbb{O}$ is ∂-finite if and only if $I \cap A_d$ (which is a left ideal in A_d) is holonomic.*

Proof. The backwards direction is simple to prove: Given a holonomic ideal, by the elimination property (Theorem 2.7) there exists (for all $1 \leq i \leq d$) a nonzero operator that involves only D_{x_i} and none of the remaining differential operators $D_{x_j}, j \neq i$. But these operators form a rectangular system and therefore generate a ∂-finite left ideal in $\mathbb{O}$ (Proposition 2.10).

The other direction in more difficult to show and we will not give the proof here. A proof that is adapted for this special situation and therefore more elementary than Kashiwara's, can be found in the appendix of [86]. □

Shift setting

Unfortunately, when considering multivariate sequences, the relation between holonomic and ∂-finite is not as close as in the differential setting. The reason is that an analogue of Theorem 2.22 does not exist for the shift case and there are functions that are ∂-finite but not holonomic. The most prominent example to illustrate this fact has been given by Wilf and Zeilberger [91].

Example 2.23. *The bivariate sequence* $f(k,n) = \frac{1}{k^2+n^2}$ *is easily identified to be ∂-finite since it is hypergeometric. Therefore it has two annihilating operators that generate a zero-dimensional ideal*

$$I = {}_{\mathbb{O}}\big\langle (k^2+2k+n^2+1)S_k + (-k^2-n^2), (k^2+n^2+2n+1)S_n + (-k^2-n^2)\big\rangle$$

in $\mathbb{O} = \mathbb{Q}(k,n)[S_k; S_k, 0][S_n; S_n, 0]$ *and the* $\mathbb{Q}(k,n)$*-vector space dimension of* $\mathbb{O}/I$ *is 1.*

Assume for now that $f(k,n)$ *is holonomic; then by the elimination property there exists a recurrence free of* k *so that*

$$\frac{p_1(n)}{(k+a_1)^2+(n+b_1)^2} + \cdots + \frac{p_d(n)}{(k+a_d)^2+(n+b_d)^2} = 0, \quad a_j, b_j \in \mathbb{N}.$$

For a fixed integer $n > 0$ *for which not all* $p_j(n)$ *are zero, we observe that each nonzero term introduces two poles* $\pm i(n+b_j)-a_j$*. All these poles are pairwise distinct and they cannot cancel away unless all* $p_j(n)$ *are zero. Therefore no such recurrence can exist and* $f(k,n)$ *is not holonomic.*

Conclusion

We have seen that holonomic and ∂-finite functions are closely related (see also Figure 2.3); in fact most of the "interesting" functions lie in their intersection. The reason for not restricting ourselves to just one of these two classes is that we need certain properties of either class. We want to make use of the fact that ∂-finite functions are much easier to handle: First the closure properties are are simpler to perform, and second, the base cases, i.e., the ∂-finite descriptions of basic expressions, are easier to obtain. Let's give two examples:

Example 2.24. *We continue with Example 2.1 where we studied annihilating ideals for the function* $f(x) = x^3$*. Treating* f *as a ∂-finite function, we observe that it is hyperexponential and hence take its first order differential equation* $xD_x - 3$*. It generates a ∂-finite ideal* I *in the Ore algebra* $\mathbb{O} = \mathbb{Q}(x)[D_x; 1, D_x]$ *and the* $\mathbb{Q}(x)$*-dimension of* $\mathbb{O}/I$ *is 1. Since it cannot be smaller (only the function that is identically 0 is annihilated by the whole ring) we know that we have the complete annihilator. Treating* f *as a holonomic function, we have seen that the first order differential does not* *generate the whole annihilator, since the annihilating operator D_x^4 is missing. However, in the ∂-finite ideal it is contained as the following calculation shows:*

$$D_x^4 = \left(\frac{1}{x} D_x^3\right) \cdot (x D_x - 3) .$$

This example illustrates a phenomenon that is of importance in practice. Sometimes we would like to determine a holonomic ideal in a polynomial Ore algebra from a given ∂-finite ideal in the corresponding rational Ore algebra. More concretely, let $I \subset \mathbb{O}_{\text{rat}}$ be a ∂-finite ideal in some rational Ore algebra $\mathbb{O}_{\text{rat}} = \mathbb{K}(\boldsymbol{v})[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$. We are interested in the intersection $I \cap \mathbb{O}_{\text{pol}}, \mathbb{O}_{\text{pol}} = K[\boldsymbol{v}][\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$, which in general is a quite difficult problem. In the pure differential case it is named *Weyl closure* and has been solved completely by Tsai ([88] for the univariate case, and [87] for the multivariate case). As soon as shift operators are involved the question is still open. An easy workaround which more or less works in practice is to cancel the denominators of the generators of $I \subset \mathbb{O}_{\text{rat}}$ in order to use them as generators of an ideal in $\mathbb{O}_{\text{pol}}$. The pitfall hereby is that the result in general will only be a subideal of $I \cap \mathbb{O}_{\text{pol}}$ (as was demonstrated in Example 2.24). We will refer to this phenomenon as *extension/contraction*.

Example 2.25. *We want to study annihilating ideals of orthogonal polynomials. Their differential equations and recurrence relations are well known. So if we take for example the Gegenbauer polynomials $C_n^m(x)$, we can easily obtain a ∂-finite description by looking up the corresponding relations and compute a Gröbner basis of them:*

$$\begin{aligned}\{&(n+1)S_n + (1-x^2)D_x + (-2mx - nx),\\ &2mS_m - xD_x + (-2m-n),\\ &(x^2-1)D_x^2 + (2mx+x)D_x + (-2mn - n^2)\}.\end{aligned}$$

The leading monomials being S_n, S_m, and D_x^2 we have just two monomials under the staircase. But we know that these polynomials are neither hypergeometric nor hyperexponential (which would leave only one monomial under the staircase), hence we have indeed found the complete annihilating ideal! In contrast, it would be much more difficult to prove that a given holonomic ideal is the complete annihilator of $C_n^m(x)$.

It is a hot topic in D-module theory to compute annihilators in the Weyl algebra. Recently, algorithms for determining the complete annihilator of f^s in A_d, f being a polynomial in $\mathbb{K}[x_1, \ldots, x_d]$, have been designed [60, 76] and implemented [56], but it is still a highly nontrivial task both from theoretical and implementational point of view. Note that in the ∂-finite setting it is rather trivial to get the annihilator of f^s, since it is hyperexponential in all the x_i.

So why do we not just forget about holonomicity and deal only with ∂-finite functions? The reason is the following: when justifying that some of the algorithms to be presented in Chapter 3 indeed terminate, we have to refer to the elimination property (Theorem 2.7) which is a property of holonomic ideals. In practice therefore we want to deal with functions that are both holonomic and ∂-finite for the abovely mentioned reasons. Figure 2.3 shows that this is not too much of a restriction.

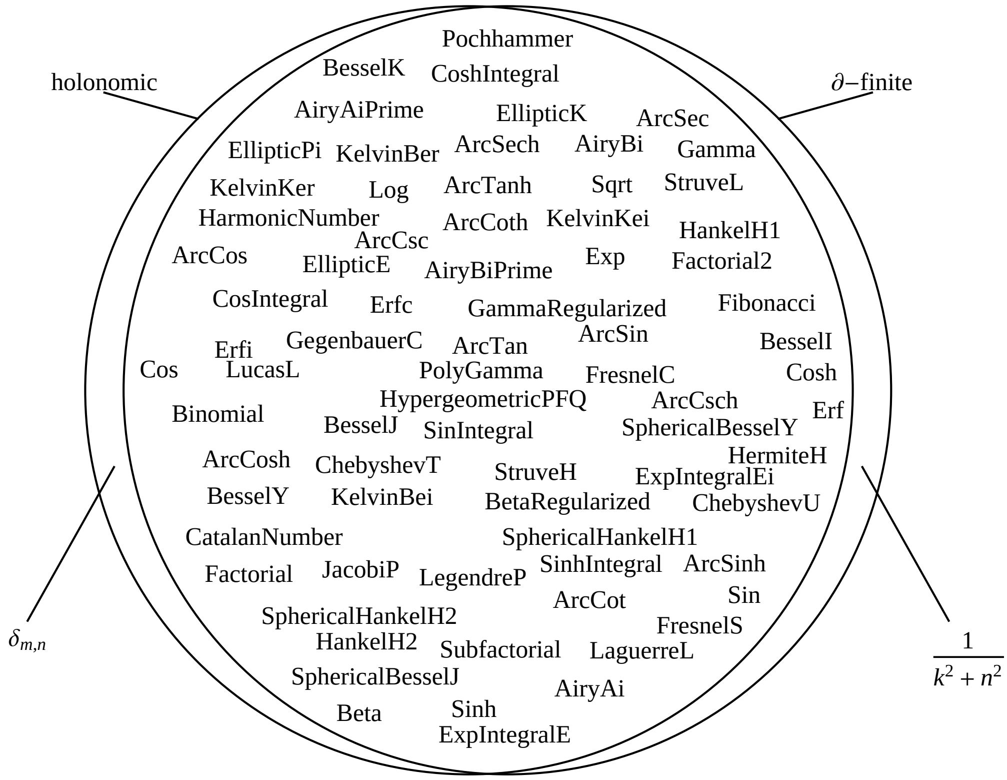

Figure 2.3: Holonomic and ∂-finite functions

2.5 Univariate versus multivariate

The use of univariate recurrences and differential equations is very classical [79, 58, 77, 59], and also the new Mathematica functionality `DifferenceRoot` and `DifferentialRoot` introduced in version 7.0 deals only with univariate such equations. That's why we want to spend some effort on discussing the essential differences between the usage of univariate and multivariate ∂-finite descriptions. A multivariate ∂-finite function $f(v_1, \ldots, v_d)$ can always be interpreted as a univariate ∂-finite function in one of these variables, say $f(v_1)$, treating the remaining variables $v_2, \ldots, v_d$ as parameters. Given a ∂-finite annihilating ideal of f with respect to the Ore algebra $\mathbb{O} = \mathbb{K}(\boldsymbol{v})[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$, there exists an annihilating operator that involves only ∂_1 which is guaranteed by the existence of a rectangular system (see Proposition 2.10). This operator then gives rise to the univariate ∂-finite description of $f(v_1)$.

First note that summation and integration, using the principle of creative telescoping (see Chapter 3), is only possible if we consider the input as a multivariate ∂-finite function. Second it is often the case that the multivariate description is less involved than the univariate one. In particular when the vector space dimension of $\mathbb{O} \bullet f$ is large then it happens that the mixed operators are of moderate size compared to the ordinary operators of the rectangular system whose order usually corresponds to the above mentioned vector space dimension. Just to give some concrete numbers: A mixed annihilating ideal of the product of Bessel functions $xJ_1(ax)I_1(ax)Y_0(x)K_0(x)$ has the leading monomials $\{D_a^3 D_x, D_a^4, D_a^2 D_x^3, D_a D_x^5, D_x^7\}$ which leave 16 monomials under the staircase and the degrees of the coefficients do not exceed 7 in a and x respectively; it easily fits on one page. The ordinary differential equation in x has order 16 (as expected) but coefficients that involve a and x with degrees up to 68 and 88, respectively; its size is about 700 kB. And third, the handling of the initial values is different: If we deal with multivariate ∂-finite descriptions we are usually left with checking finitely many initial values, whereas an ordinary operator forces us to check "infinitely" many initial values, in the sense that we have to consider initial values with the symbolic parameters, as the following examples illustrates.

Example 2.26. *In their delightful book "Irresistible Integrals" [19], George Boros and Victor Moll studied the integral*

$$\int_0^\infty \frac{\mathrm{d}x}{(x^4 + 2ax^2 + 1)^{m+1}} = \frac{\pi P_m^{(m+1/2, -m-1/2)}(a)}{2^{m+3/2}(a+1)^{m+1/2}} \tag{2.8}$$

where P denotes the Jacobi polynomials. Many proofs of how to tackle this integral are collected in [8]. We want to sketch how to prove it with computer algebra and present two strategies. In the first we view both sides as univariate ∂-finite functions with respect to $\mathbb{Q}(a)[D_a; 1, D_a]$. Using closure properties and an integration algorithm (see Chapter 3) we derive the same differential equation

$$4(a^2 - 1)D_a^2 + 4a(2m+3)D_a + 4m + 3$$

for both sides. Hence we have to compare two initial values, but let's concentrate on the first one ($a = 0$)*. We have to show that*

$$\int_0^\infty \frac{\mathrm{d}x}{(x^4+1)^{m+1}} = \frac{\pi P_m^{(m+1/2,-m-1/2)}(0)}{2^{m+3/2}} \tag{2.9}$$

for symbolic m*. On the other hand, if we consider* (2.8) *as an identity of bivariate* ∂*-finite functions in* a *and* m*, we get an additional operator*

$$(4m+4)S_m - 2aD_a - 4m - 3 \in \mathbb{Q}(a,m)[D_a;1,D_a][S_m;S_m,0]$$

that annihilates either side of (2.8)*. Now the problem of comparing the first initial value* ($a = 0$ *and* $m = 0$) *reduces to showing*

$$\int_0^\infty \frac{\mathrm{d}x}{(x^4+1)} = \frac{\pi P_0^{(1/2,-1/2)}(0)}{2^{3/2}} = \frac{\pi}{2\sqrt{2}}$$

which is much easier than condition (2.9)*. The same applies to the second initial value that has to be compared.*

This example also reveals another remarkable aspect: With the first strategy we prove that (2.8) *holds for all* m *for which we succeed to prove* (2.9)*, e.g., for all real numbers* $m > -\frac{3}{4}$*, whereas in the second approach* m *is restricted a priori to be a natural number.*

2.6 Non-holonomic functions

So far we have presented an algorithmic framework for ∂-finite functions, which are characterized by a property of their annihilating ideals, namely having only finitely many monomials under the staircase. Hence, in terms of Hilbert dimension, ∂-finite ideals are just the zero-dimensional ideals. Recently, Frédéric Chyzak, Manuel Kauers, and Bruno Salvy observed [27] that the ∂-finite framework can be extended to functions that possess higher-dimensional annihilating ideals. An instance for such a function was already shown in Example 2.12 where the Stirling numbers of the first kind were studied. In general it can be a very tedious task to prove that some given function is not holonomic, see [38, 35]. The following theorem is an important result in [27] and it prepares the stage for performing closure properties of non-holonomic functions.

Theorem 2.27. *Let $I_1, I_2 \subseteq \mathbb{O}$ be annihilating left ideals in the Ore algebra $\mathbb{O} = \mathbb{K}(\boldsymbol{x})[\boldsymbol{\partial_x}; \boldsymbol{\sigma_x}, \boldsymbol{\delta_x}]$ for f_1 and f_2, respectively. Then*

1. $\dim \operatorname{Ann}_{\mathbb{O}}(P \bullet f_1) \leq \dim I_1 \quad$ *for all* $P \in \mathbb{O}$,
2. $\dim \operatorname{Ann}_{\mathbb{O}}(f_1 + f_2) \leq \max(\dim I_1, \dim I_2)$,
3. $\dim \operatorname{Ann}_{\mathbb{O}}(f_1 f_2) \leq \dim I_1 + \dim I_2$, *if some technical conditions (as in Theorem 2.16) hold, which is always the case for the Ore algebras of practical relevance.*

The algorithms for executing the ∂-finite closure properties can be easily adopted to higher-dimensional annihilating ideals; it basically concerns the termination condition. The algorithms now stop as soon as an annihilating ideal of the predicted dimension is found. It can well be that the predicted dimension exceeds the number of involved Ore operators in which case no relation at all is expected to exist.

Example 2.28. *We focus on Stirling numbers once again, but now on both the first and the second kind. They are annihilated by $S_m S_n + m S_n - 1$ and $S_m S_n - (n+1) S_n - 1$, respectively, so that their annihilating ideal have dimension 1. Theorem 2.27 predicts the existence of a one-dimensional ideal for the sum of these two sequences. Indeed, we can find the recurrence*

$$(m+n+1)(m+n+2)S_m^2 S_n^2 + (m+n+1)(m^2+m-n^2-4n-4)S_m S_n^2 - \\ 2(m+n+2)^2 S_m S_n - m(n+2)(m+n+1)(m+n+3)S_n^2 - \\ (m+n+3)\left(m^2-n^2-3n-2\right)S_n + (m+n+2)(m+n+3).$$

In contrast, we cannot expect to find a recurrence for the product, since in this case Theorem 2.27 predicts dimension 2 which corresponds to the zero ideal.

In the following chapter we will explain the method of creative telescoping. We will describe the related algorithms for the holonomic/∂-finite setting only, but reveal already here that they can be generalized to non-holonomic functions as well. This is the main result of [27] and we refer to the original paper for further details. Some examples of non-holonomic function identities can be found in Section 4.6.

Chapter 3

Algorithms for Summation and Integration

The basic principle that makes all the algorithms of this chapter meaningful, is the idea of *creative telescoping* (we even were tempted to utilize it for entitling the whole chapter). To our knowledge, the first occurrence of this notion happened in 1979 in van der Poorten's report [89] on Apéry's proof of the irrationality of $\zeta(3)$. But for sure the principle was known and used long before as an ad hoc trick to solve sums and integrals. Just think of the practice of *differentiating under the integral sign*, that was made popular by Richard Feynman in his enjoyable book "Surely You're Joking, Mr. Feynman!" [34]. It was Doron Zeilberger who equipped creative telescoping with a concrete well-defined meaning and connected it to an algorithmic method.

The idea of creative telescoping is rather simple and works for summation problems as well as for integrals. The aim is to get relations (like recurrences, differential equations, etc.) for the expression in question, that may help in evaluating it or that may serve to prove an already given identity.

When we want to do a definite sum of the form $\sum_{k=a}^{b} f(k, \boldsymbol{w})$ then we search for creative telescoping operators that annihilate f and that are of the form

$$T = P(\boldsymbol{w}, \boldsymbol{\partial_w}) + \Delta_k Q(k, \boldsymbol{w}, S_k, \boldsymbol{\partial_w}) \tag{3.1}$$

where $\boldsymbol{\partial_w}$ stands for some Ore operators that act on the variables $\boldsymbol{w}$. The operator P is called the *principal part*, and we will refer to Q as the *delta part*. With such an operator T we can immediately derive a relation for the definite sum:

$$\begin{aligned}
0 &= \sum_{k=a}^{b} T(k, \boldsymbol{w}, S_k, \boldsymbol{\partial_w}) \bullet f(k, \boldsymbol{w}) \\
&= \sum_{k=a}^{b} P(\boldsymbol{w}, \boldsymbol{\partial_w}) \bullet f(k, \boldsymbol{w}) \;+\; \sum_{k=a}^{b} \big(\Delta_k Q(k, \boldsymbol{w}, S_k, \boldsymbol{\partial_w})\big) \bullet f(k, \boldsymbol{w}) \\
&= P(\boldsymbol{w}, \boldsymbol{\partial_w}) \bullet \sum_{k=a}^{b} f(k, \boldsymbol{w}) \;+\; \underbrace{\Big[Q(k, \boldsymbol{w}, S_k, \boldsymbol{\partial_w}) \bullet f(k, \boldsymbol{w})\Big]_{k=a}^{k=b+1}}_{\text{inhomogeneous part}}.
\end{aligned}$$

Depending on whether the *inhomogeneous part* evaluates to zero or not, we have P as an annihilating operator for the sum, or we get an inhomogeneous relation for the sum. In the latter case, if one is not happy with that, one can homogenize the relation by multiplying an annihilating operator for the inhomogeneous part to P from the left. Some of the algorithms in this section can also tackle multiple sums; in that case a telescoping operator of the form

$$P(\boldsymbol{w}, \boldsymbol{\partial_w}) + \Delta_{k_1} Q_1(\boldsymbol{k}, \boldsymbol{w}, \boldsymbol{S_k}, \boldsymbol{\partial_w}) + \cdots + \Delta_{k_j} Q_j(\boldsymbol{k}, \boldsymbol{w}, \boldsymbol{S_k}, \boldsymbol{\partial_w}) \tag{3.2}$$

where $\boldsymbol{k} = k_1, \ldots, k_j$ are the summation variables, is desired.

Similarly one can derive annihilating operators for a definite integral $\int_a^b f(x, \boldsymbol{w})\,\mathrm{d}x$. In this case we look for creative telescoping operators that annihilate f and that are of the form

$$T = P(\boldsymbol{w}, \boldsymbol{\partial_w}) + D_x Q(x, \boldsymbol{w}, D_x, \boldsymbol{\partial_w}). \tag{3.3}$$

Again it is straightforward to deduce a relation for the integral

$$\begin{aligned}
0 &= \int_a^b T(x, \boldsymbol{w}, D_x, \boldsymbol{\partial_w}) \bullet f(x, \boldsymbol{w})\,\mathrm{d}x \\
&= \int_a^b P(\boldsymbol{w}, \boldsymbol{\partial_w}) \bullet f(x, \boldsymbol{w})\,\mathrm{d}x + \int_a^b \big(D_x Q(x, \boldsymbol{w}, D_x, \boldsymbol{\partial_w})\big) \bullet f(x, \boldsymbol{w})\,\mathrm{d}x \\
&= P(\boldsymbol{w}, \boldsymbol{\partial_w}) \bullet \int_a^b f(x, \boldsymbol{w})\,\mathrm{d}x + \Big[Q(x, \boldsymbol{w}, D_x, \boldsymbol{\partial_w}) \bullet f(x, \boldsymbol{w})\Big]_{x=a}^{x=b}
\end{aligned}$$

which may be homogeneous or inhomogeneous. Analogously to the summation case, some algorithms can treat multiple integrals in which case the desired telescoping operators have the form

$$P(\boldsymbol{w}, \boldsymbol{\partial_w}) + D_{x_1} Q_1(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{D_x}, \boldsymbol{\partial_w}) + \cdots + D_{x_j} Q_j(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{D_x}, \boldsymbol{\partial_w}). \tag{3.4}$$

where now $\boldsymbol{x} = x_1, \ldots, x_j$ are the integration variables.

In practice it happens very often that the inhomogeneous part vanishes. The reason for that is because many sums and integrals run over *natural boundaries*. This concept is often used, e.g., in Takayama's algorithm, to argue a priori that there will be no inhomogeneous parts after telescoping. For that purpose, we define that $\sum_{k=a}^{b} f$ resp. $\int_a^b f\,\mathrm{d}x$ has natural boundaries if for any arbitrary operator $P \in \mathbb{O}$ for a suitable Ore algebra $\mathbb{O}$ the expression $\big[P \bullet f\big]_{k=a}^{k=b+1}$ resp. $\big[P \bullet f\big]_{x=a}^{x=b}$ evaluates to zero. Typical examples for natural boundaries are sums with finite support, or integrations over the whole real line that involve something like $\exp(-x^2)$. Likewise contour integrals along a closed path do have natural boundaries.

All algorithms that will be presented in the following rely on this principle and their aim is, given an annihilating ideal for the summand/integrand, to compute creative telescoping operators that give rise to an annihilating ideal for the sum/integral. In order to illustrate how the algorithms work we chose—for sake of clarity—one of the simplest possible examples: the binomial sum $\sum_{k=0}^{n} \binom{n}{k}$. It is so simple that all computations can be carried out by hand so that the demonstrations are very instructive. But this should not give the impression that the following algorithms can only solve trivial examples. At the end of this chapter as well as in Chapter 4 we will give examples that are much more involved and hard to solve by hand. As input we will always start with the ∂-finite annihilating ideal of the summand $\binom{n}{k}$. Since it is hypergeometric we find that it is generated by the two operators $G_1 = (n-k+1)S_n - (n+1)$ and $G_2 = (k+1)S_k + (k-n)$ that form a Gröbner basis in $\mathbb{Q}(k,n)[S_k; S_k, 0][S_n; S_n, 0]$ with respect to total degree order (actually: with respect to any term order!).

3.1 Zeilberger's slow algorithm

In his seminal paper "A holonomic systems approach to special functions identities" [93], Doron Zeilberger proposed a general method for tackling (multiple) integration and summation problems of the form

$$\sum_{v_1=a_1}^{b_1} \cdots \sum_{v_l=a_l}^{b_l} \int_{a_{l+1}}^{b_{l+1}} \cdots \int_{a_j}^{b_j} f(\boldsymbol{v}, \boldsymbol{w})\,\mathrm{d}v_{l+1} \ldots \mathrm{d}v_j \tag{3.5}$$

where f is a holonomic function in the summation and integration variables $\boldsymbol{v} = v_1, \ldots, v_j$ and in the additional parameters $\boldsymbol{w}$ (which should be nonempty). We will act with the difference operators $\Delta_{v_1}, \ldots, \Delta_{v_l}$ on the summation variables (in practice it is more natural and more efficient to work with shift operators but for now notation becomes simpler by using delta operators). Of course, in the q-case we have to consider q-difference operators instead (we mention this here once and omit this aspect in the following to prevent further confusion). Similarly we want to act with the differential operators $D_{v_{l+1}}, \ldots, D_{v_j}$ on the integration variables. We will use the shorthand notation $\boldsymbol{\partial_v}$ to refer to the Ore operators $\Delta_{v_1}, \ldots, \Delta_{v_l}, D_{v_{l+1}}, \ldots, D_{v_j}$. Finally we denote by $\boldsymbol{\partial_w}$ the unspecified Ore operators that act on the variables $\boldsymbol{w}$ (but have to restrict $\boldsymbol{\partial_w}$ to the standard Ore operators for which the notion of holonomy makes sense). For our purposes we have to assume that the function f allows us to write the integration and summation quantifiers in any possible order; otherwise creative telescoping would be impractical. Therefore (3.5) represents the most general problem that is in the spirit of the holonomic systems approach.

Shortly after [93], Zeilberger came up with the algorithm that nowadays is known as Zeilberger's algorithm and which solves the special case of single (and not multiple) sums (and not integrals) of proper hypergeometric (and not general holonomic) summands. He himself named this the *fast algorithm*, and in consequence the one that we are discussing now was named by him the *slow algorithm* [92, 94, 69].

As input, Zeilberger's slow algorithm takes a holonomic annihilating ideal for $f(\boldsymbol{v}, \boldsymbol{w})$. The main step towards a creative telescoping relation consists in brute-force elimination of the summation/integration variables $\boldsymbol{v}$. Note that it is guaranteed by holonomy (see elimination property, Theorem 2.7) that nontrivial $\boldsymbol{v}$-free operators exist. Once such operators are found they can be easily transformed into the form (3.2)+(3.4) by successive divisions with remainder: The principal part will be the last remainder that survives after all divisions. Theoretically it could happen that the remainder is zero in which case the $\boldsymbol{v}$-free operator is rather useless for our purposes. Assume that we run into this unlucky case; then it suffices to multiply the $\boldsymbol{v}$-free operator by some suitable power product $\boldsymbol{v}^{\boldsymbol{\alpha}}$. Of course, the result will not be $\boldsymbol{v}$-free any more, but the (now nonzero) principal part will be so (for more details see [90]). Hence the original problem has been solved.

So far we have not said how to find $\boldsymbol{v}$-free operators. Zeilberger in his original paper suggested Sylvester's dialytic elimination, which was studied further by Peter Paule [65]. Later Zeilberger admitted that the use of noncommutative Gröbner bases might be preferable: By choosing a monomial order where the variables $v_1, \ldots, v_j$ are lexicographically greater than the other variables $\boldsymbol{w}$ and all the Ore operators $\boldsymbol{\partial_v}$ and $\boldsymbol{\partial_w}$, we obtain the desired result. Usually a block order with two blocks is chosen to achieve the elimination.

In practice the following problem arises: How can we get a holonomic annihilating ideal for the function f? In Section 2.3 we have only discussed how to obtain ∂-finite annihilating ideals that, by definition, are part of a rational Ore algebra. In the following we want to denote by $I \subseteq \mathbb{O}_{\text{rat}}$ a given annihilating ∂-finite ideal for f where

$$\mathbb{O}_{\text{rat}} = \mathbb{K}(\boldsymbol{v}, \boldsymbol{w})[\boldsymbol{\partial_v}; \boldsymbol{\sigma_v}, \boldsymbol{\delta_v}][\boldsymbol{\partial_w}; \boldsymbol{\sigma_w}, \boldsymbol{\delta_w}].$$

But to apply the above method we need an annihilating holonomic ideal in the polynomial Ore algebra. Unfortunately we do not know in general how to get that. And in fact, it suffices to work in an Ore algebra $\mathbb{O}$ where only the variables $\boldsymbol{v}$ to be eliminated occur polynomially:

$$\mathbb{O} = \mathbb{K}(\boldsymbol{w})[\boldsymbol{v}][\boldsymbol{\partial_v}; \boldsymbol{\sigma_v}, \boldsymbol{\delta_v}][\boldsymbol{\partial_w}; \boldsymbol{\sigma_w}, \boldsymbol{\delta_w}].$$

This has the advantage that the computations are much less involved compared to the elimination in a purely polynomial Ore algebra. But it still remains to get an annihilating ideal in the algebra $\mathbb{O}$. In other words we need to determine $I \cap \mathbb{O}$, which we already mentioned in Section 2.4 to be a difficult problem. Therefore we use the workaround to cancel denominators and then use the generators of I for generating an annihilating ideal of f in $\mathbb{O}$ (which due to extension/contraction will be only a subideal of $I \cap \mathbb{O}$ in general) and to perform the elimination in this ideal.

Example 3.1. *Interpreting the two operators G_1 and G_2 in the Ore algebra $\mathbb{Q}(n)[k][S_k; S_k, 0][S_n; S_n, 0]$ they are presented as $G_1 = -kS_n + (n+1)S_n - (n+1)$ and $G_2 = kS_k + k + S_k - n$ where we already ordered the terms with respect to a monomial order that eliminates k. The elimination of k can here be obtained by hand calculation:*

$$S_k G_1 + S_n G_2 = kS_n + (n+1)S_k S_n - (n+1)S_k - (n+1)S_n.$$

The leading term can be reduced by G_1 and after removing the content $(n+1)$ we end up with the k-free operator $T = S_k S_n - S_k - 1$ (which is nothing else but Pascal's rule). Note that if we worked in the polynomial Ore algebra $\mathbb{Q}[k, n][S_k; S_k, 0][S_n; S_n, 0]$ we would not have found Pascal's rule but only a multiple of it (which again illustrates the extension/contraction problem). Division with remainder by $S_k - 1$ rewrites T to a telescoping operator

$$T = S_n - 2 + (S_k - 1)(S_n - 1).$$

The sum has natural boundaries (because of finite support), so we can neglect the delta part. Together with the initial condition $\binom{0}{0} = 1$ the principal part $S_n - 2$ gives rise to the solution 2^n.

In order to bypass the extension/contraction problem, we came up with a third method how to eliminate the variables $\boldsymbol{v}$. It is based on ansatz and coefficient comparison. In our general setting the ansatz with unknown coefficients has the form

$$\sum_{(\boldsymbol{\alpha},\boldsymbol{\beta})\in M} c_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\boldsymbol{w}) \cdot \boldsymbol{\partial}_{\boldsymbol{v}}^{\boldsymbol{\alpha}} \boldsymbol{\partial}_{\boldsymbol{w}}^{\boldsymbol{\beta}}$$

where M is a finite index set of exponent vectors (structure set). The ansatz is reduced with the Gröbner basis of the given ∂-finite ideal I, and the coefficients of the normal form are set to zero. Additionally a coefficient comparison with respect to all variables $\boldsymbol{v}$ is done to enforce the solutions $c_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ to be free of $\boldsymbol{v}$. It remains to solve a linear system over $\mathbb{K}(\boldsymbol{w})$. Since a priori it is not known which structure set will indeed deliver a solution, one has to loop, say over the total degree of the ansatz, until a solution is found. This ansatz method being a special case of the polynomial ansatz to be described in Section 3.4, we omit further details (concerning optimizations) here and refer to the below extensive description. To complete this section we demonstrate the elimination via ansatz on the binomial sum example.

Example 3.2. *To find a k-free recurrence for the binomial coefficient we start with the ansatz*

$$\sum_{\substack{(\alpha,\beta)\in\mathbb{N}^2 \\ \alpha+\beta\leq d}} c_{\alpha,\beta}(n) S_k^{\alpha} S_n^{\beta}$$

where we loop over the total degree d. To keep things short we omit the first two steps that would deliver no solution. The ansatz for total degree $d = 2$ is

$$c_{00} + c_{01}S_n + c_{10}S_k + c_{02}S_n^2 + c_{11}S_kS_n + c_{20}S_k^2.$$

The normal form with respect to the Gröbner basis $\{G_1, G_2\}$ is (after clearing denominators):

$$\begin{aligned}
&(k+1)(k+2)(k-n-2)(k-n-1)c_{00} - \\
&(k+1)(k+2)(n+1)(k-n-2)c_{01} - \\
&(k+2)(k-n-2)(k-n-1)(k-n)c_{10} + \\
&(k+1)(k+2)(n+1)(n+2)c_{02} + \\
&(k+2)(n+1)(k-n-2)(k-n-1)c_{11} + \\
&(k-n-2)(k-n-1)(k-n)(k-n+1)c_{20}.
\end{aligned}$$

After coefficient comparison we write the linear system in matrix form. We cleared the content in each equation, and to make it fit on this page, give the *transposed matrix:*

$$\begin{pmatrix} 2 & -3n^2-5n & n^2-3n-5 & 2n & 1 \\ 2 & -3n^2-7n-4 & n^2-1 & n+1 & 0 \\ 2n & -n^3+3n^2+10n+4 & 4-3n^2 & -3n-1 & -1 \\ 2 & -3n^2-9n-6 & n^2+3n+2 & 0 & 0 \\ 2n+2 & -n^3+5n+4 & -2n^2-3n-1 & -n-1 & 0 \\ n^2-n & 4n^3+6n^2-2n-2 & 6n^2+6n-1 & 4n+2 & 1 \end{pmatrix}.$$

The nullspace of the matrix is spanned by the single vector $(-1,0,-1,0,1,0)^T$ *which corresponds to the same k-free recurrence that we found with Gröbner basis elimination. Although in this example the computations are much more involved compared to Example 3.1, this method is often much faster and gives smaller results (see Example 3.7 and Section 6.3).*

3.2 Takayama's algorithm

When studying Zeilberger's slow algorithm it turns out that due to the nature of this algorithm also the delta parts are free of the summation and integration variables, in contrast to what is indicated in (3.1) and (3.3). Somehow, as soon as we have eliminated all the variables $\boldsymbol{v}$ we have already done too much (we stick to the notation introduced in the previous section because also Takayama's algorithm can do multiple sums and integrals). Gert Almqvist and Doron Zeilberger [7] were the first who observed that the complete elimination imposes more restriction than necessary, and that it would not do any harm if the delta parts contained also the variables $\boldsymbol{v}$. This fact was then exploited by Nobuki Takayama who constructed an "infinite dimensional analog of Gröbner basis" [85, 84]. But he formulated his algorithm only in the context of the Weyl algebra and in a quite theoretical fashion. It were Frédéric Chyzak and Bruno Salvy [29] who proposed optimizations that are important in practice and extended the application domain of the algorithm to more general Ore algebras (in particular shift algebras). The idea in short is the following: While in Zeilberger's slow algorithm, first $v_1, \ldots, v_j$ were eliminated and then the delta parts were divided out, the order is now reversed. In Takayama's algorithm we first reduce modulo the right ideals $\partial_{v_1}\mathbb{O}, \ldots, \partial_{v_j}\mathbb{O}$ and then perform the elimination of $\boldsymbol{v}$. The consequence is that the delta parts are not computed at all because everything that would contribute to them is thrown away in the first step. Hence we have to assume a priori that the inhomogeneous parts will vanish, i.e., we have to assure natural boundaries. On the other hand this strategy makes the algorithm much faster since the division step reduces the size of the data considerably before starting with the elimination. Last but not least we can find bigger annihilating ideals with Takayama's algorithm (i.e., shorter recurrences or differential equations with lower order) because compared to Zeilberger's slow algorithm we allow more freedom in the delta parts.

There is one technical complication in this approach. The fact that we are computing in a noncommutative algebra restricts us in the computations after having divided out the *right* ideals (since we started with a *left* ideal, there is no ideal structure any more). In particular, we are no longer allowed to multiply by either of the variables $\boldsymbol{v}$ from the left. We can easily convince ourselves that otherwise we would get faulty results: Assume we have written an operator in the form $P + D_{v_i}Q$. Multiplying it by v_i and then reducing it by $D_{v_i}\mathbb{O}$ leads to $v_iP - Q$ since we have to rewrite $v_iD_{v_i}$ as $D_{v_i}v_i - 1$. Because v_i does not commute with D_{v_i} we get the additional term $-Q$ in the result which we lose if we first remove $D_{v_i}Q$ and then multiply by v_i. Similar things happen in connection with the delta operator.

In order to find a $\boldsymbol{v}$-free operator one needs an elimination procedure that avoids multiplication by the variables $\boldsymbol{v}$. Let now $\{G_1, \ldots, G_r\} \subset \mathbb{O}$ be a finite set of operators that annihilate $f(\boldsymbol{v}, \boldsymbol{w})$, in the Ore algebra that allows elimination of the summation and integration variables $\boldsymbol{v}$. Such a set of operators can be obtained for example using the workaround mentioned in the previous section (starting with a ∂-finite ideal in the rational Ore algebra $\mathbb{O}_{\text{rat}}$). But usually then $\{G_1, \ldots, G_r\}$ will not constitute a Gröbner basis in the new algebra $\mathbb{O}$; in some examples it is of advantage to compute a Gröbner basis of $\{G_1, \ldots, G_r\} \subset \mathbb{O}$ as a preprocessing step (option `Saturate` in our command `Takayama`). Let $G'_1, \ldots, G'_r \in \mathbb{K}(\boldsymbol{w})[\boldsymbol{v}][\boldsymbol{\partial_w}; \boldsymbol{\sigma_w}, \boldsymbol{\delta_w}]$ denote the corresponding reductions modulo the right ideals $\partial_{v_1}\mathbb{O}, \ldots, \partial_{v_j}\mathbb{O}$. For $1 \leq i \leq r$ we can write

$$G'_i(\boldsymbol{v}, \boldsymbol{w}, \boldsymbol{\partial_w}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^j} \boldsymbol{v}^{\boldsymbol{\alpha}} G_{i,\boldsymbol{\alpha}}(\boldsymbol{w}, \boldsymbol{\partial_w})$$

where $G_{i,\boldsymbol{\alpha}} \in \mathbb{O}' = \mathbb{K}(\boldsymbol{w})[\boldsymbol{\partial_w}; \boldsymbol{\sigma_w}, \boldsymbol{\delta_w}]$. Elimination of $\boldsymbol{v}$ now amounts to finding a linear combination

$$P_1(\boldsymbol{w}, \boldsymbol{\partial_w}) \begin{pmatrix} G_{1,\boldsymbol{\alpha}_1} \\ G_{1,\boldsymbol{\alpha}_2} \\ G_{1,\boldsymbol{\alpha}_3} \\ \vdots \end{pmatrix} + \cdots + P_r(\boldsymbol{w}, \boldsymbol{\partial_w}) \begin{pmatrix} G_{r,\boldsymbol{\alpha}_1} \\ G_{r,\boldsymbol{\alpha}_2} \\ G_{r,\boldsymbol{\alpha}_3} \\ \vdots \end{pmatrix} = \begin{pmatrix} P(\boldsymbol{w}, \boldsymbol{\partial_w}) \\ 0 \\ 0 \\ \vdots \end{pmatrix}$$

for some operators $P_1, \ldots, P_r \in \mathbb{O}'$. With $\boldsymbol{\alpha}_1 = (0, \ldots, 0)$ the vector on the right-hand side corresponds to the desired $\boldsymbol{v}$-free operator. More algebraically speaking, the elimination happens in an $\mathbb{O}'$-module that is generated by the above vectors. It is achieved by computing a Gröbner basis of this module with an ordering that first compares positions in the vectors and then breaks ties by some monomial order ("position over term").

In general, this will not work yet since we cannot expect to succeed in the elimination without multiplying by $\boldsymbol{v}$ at all. We only have to ensure that this multiplication takes place before dividing out the delta parts. Hence we include multiples of the G_i by power products of $\boldsymbol{v}$ from the very beginning. By M we denote the module that is generated by the vectors that correspond to the G_i and their $\boldsymbol{v}$-multiples. Note that the elements of M have infinite dimension and also that $(P(\boldsymbol{w}, \boldsymbol{\partial_w}), 0, \dots)^T \in M$ if and only if there exist $Q_1, \dots, Q_j \in \mathbb{O}$ such that

$$P + \partial_{v_1} Q_1 + \cdots + \partial_{v_j} Q_j \tag{3.6}$$

is a telescoping relation in the ideal ${}_{\mathbb{O}}\langle G_1, \dots, G_r\rangle$. For practical purposes we have to truncate the module M by considering only elements up to a certain dimension d, i.e., which have zeros in all positions greater than d. The natural choice for d is the determined by the powers of $\boldsymbol{v}$ that appear in $G_1, \dots, G_r$. So in the first run only $\boldsymbol{v}$-multiples of such G_i are added that do not involve $v_1, \dots, v_j$ with their highest powers occurring in $\{G_1, \dots, G_r\}$. But we are not guaranteed that for any telescoping operator of the form (3.6) the principal part P is an element of the truncated module. In the unlucky case that no $\boldsymbol{v}$-free operator is found, more multiples by power products of $\boldsymbol{v}$ have to be included and the bound d has to be increased. In the case of multiple summations and integrations ($j > 1$) there are several options how to increase the dimension of the module elements. In the pseudo-code description of the algorithm (see Figure 3.1) we decided to increase the maximal degree of each of the variables $v_1, \dots, v_j$ by one. But also finer strategies, for example increasing in each step only the degree of one single variable, are possible. A final remark concerns the termination of the algorithm: it is guaranteed provided that the input is holonomic. Then by the elimination property there exists a $\boldsymbol{v}$-free operator and a fortiori an operator of the form (3.6).

Example 3.3. *We now want to find a recurrence for the binomial sum with Takayama's algorithm. Recall that it has natural boundaries and therefore the algorithm is applicable. We first observe that both G_1 and G_2 are of degree 1 in k. Hence we can start right away with these two, without having to include certain multiples by powers of k. The next step is to divide out the delta parts. The first operator $G_1 = (n-k+1)S_n - (n+1)$ is free of S_k and hence is fine. The second operator rewrites as $G_2 = (S_k - 1)k + 2k - n$ which gives $2k - n$ modulo the right ideal generated by $S_k - 1$. We represent* *these two elements in the module whose positions correspond to the powers of k and find a linear combination between them:*

$$2\begin{pmatrix}(n+1)S_n-(n+1)\\-S_n\end{pmatrix}+S_n\begin{pmatrix}-n\\2\end{pmatrix}=\begin{pmatrix}(n+1)S_n-2(n+1)\\0\end{pmatrix}.$$

The vector on the right-hand side represents the k-free operator that we have found. After removing the content we end up with the same principal part as with Zeilberger's slow algorithm in Example 3.1.

```
Input: set of summation and integration variables {v_1,...,v_j},
       {G_1,...,G_r} ⊂ O = K(w)[v][∂_v; σ_v, δ_v][∂_w; σ_w, δ_w]
       where ∂_{v_i} = Δ_{v_i} or ∂_{v_i} = D_{v_i} for 1 ≤ i ≤ j
Output: principal parts {P_1,...,P_s} of some telescoping operators
        with respect to ∂_{v_1},...,∂_{v_j} in _O⟨G_1,...,G_r⟩
-----------------------------------------------------------------
d := (d_1,...,d_j) := (max_{1≤i≤r} deg_{v_1} G_i, ..., max_{1≤i≤r} deg_{v_j} G_i)
P := ∅
while P = ∅
   H := ∅
   for i := 1 to r
      e := (deg_{v_1} G_i, ..., deg_{v_j} G_i)
      H := H ∪ {v^α · G_i | α ≤ d − e}
   H' := {H_i mod ⟨∂_{v_1},...,∂_{v_j}⟩_O | H_i ∈ H}
   P := module Gröbner basis of H' eliminating v_1,...,v_j
   P := {P_i ∈ P | deg_v P_i = 0}
   d := (d_1 + 1, ..., d_j + 1)
return P
```

Figure 3.1: Takayama's algorithm

Last but not least we want to mention that Takayama's algorithm as we presented it here, can be adapted (then being more in the flavor how Takayama originally presented it) for computing also the delta parts. In this case the quotients of the divisions modulo the right ideals are not thrown away, but kept and propagated through the whole run of the module Gröbner basis computation. In the end (and also in between) they can be reduced to normal form modulo the input ideal. In our implementation this can be achieved by using `Takayama` with the option `Extended` (see also page 93).

3.3 Chyzak's algorithm

In Section 3.5 we will drastically demonstrate that Zeilberger's slow algorithm is usually very inefficient. Takayama's algorithm has the disadvantage that natural boundaries have to be assured a priori, which might be the case or not, and even if so, it can be hard to prove. Chyzak's algorithm [26] overcomes these handicaps, but suffers from another one: It can only deal with single sums and single integrals (multiple ones can only be solved by applying the algorithm recursively). But still, for most summation and integration problems that can be treated in the frame of the holonomic systems approach, it is the means of choice.

We will introduce Chyzak's algorithm, both for indefinite and definite summation (and integration), as a generalization of classical algorithms due to Gosper and Zeilberger, which we recall only very briefly for this purpose. For readers who are not familiar with these algorithms we point to the numerous excellent expositions [70, 72, 53].

Indefinite problems

For indefinite hypergeometric summation—i.e., given $f(k)$ hypergeometric find a hypergeometric antidifference $g(k)$ so that $f(k) = g(k+1) - g(k)$—there is Gosper's algorithm [41, 66] which decides whether there exists $g(k)$ of the form $q(k)f(k)$ for some rational function $q(k)$, and in the affirmative case computes it. The algorithm is a complete decision procedure which either gives back the desired function $g(k)$ allowing to evaluate the sum expression $\sum_{k=a}^{b} f(k) = g(b+1) - g(a)$ in closed form for arbitrary summation bounds, or proves that no such antidifference exists. There is also a version of Gosper's algorithm that handles indefinite integrals of hyperexponential functions [7]. Gosper's algorithm has been generalized to ∂-finite functions by Frédéric Chyzak, then dealing both with indefinite sums and indefinite integrals.

In Gosper's algorithm we look for an antidifference $g(k)$ that is a rational function multiple of the summand. This is equivalent to claiming that $g(k)$ lies in the (one-dimensional) $\mathbb{K}(k)$-vector space that is spanned by all shifts of the summand $f(k)$. The telescoping equation $g(k+1) - g(k) = f(k)$ is written in operator notation as

$$\Delta_k q(k) - 1 \in \operatorname{Ann}_{\mathbb{O}} f(k) \quad \text{with } \mathbb{O} = \mathbb{K}(k)[\Delta_k; S_k, \Delta_k]$$

where $q(k)$ is the rational function to be found.

Let $f(v, \boldsymbol{w})$ from now on denote a ∂-finite function with respect to the Ore algebra $\mathbb{O} = \mathbb{K}(v, \boldsymbol{w})[\partial_v; \sigma_v, \delta_v][\boldsymbol{\partial_w}; \boldsymbol{\sigma_w}, \boldsymbol{\delta_w}]$, characterized by its annihilating left ideal that is given by a Gröbner basis $G \subset \mathbb{O}$. In the case where v

is a discrete variable and $\partial_v = \Delta_v$, the goal is to compute the indefinite sum $\sum_v f(v, \boldsymbol{w})$. In the other case when v is a continuous variable and $\partial_v = D_v$, we want to do the indefinite integral $\int f(v, \boldsymbol{w})\,\mathrm{d}v$.

Analogously to Gosper's algorithm, it is natural to search for the antidifference (resp. antiderivative) in the finite-dimensional vector space that is spanned by the shifts and derivatives of the summand (resp. integrand). As we have learned in Section 2.3, each vector space element can be represented as

$$\big(q_1U_1 + \cdots + q_mU_m\big) \bullet f, \quad q_i \in \mathbb{K}(v, \boldsymbol{w}), U_i \in \mathbb{O},$$

where $U = \{U_1, \ldots, U_m\}$ denotes the set of $\mathbb{O}$-monomials that lie under the staircase of G. Hence the telescoping problem for ∂-finite functions is the following: Find rational functions $q_1, \ldots, q_m \in \mathbb{K}(v, \boldsymbol{w})$ such that

$$\partial_v \cdot \big(q_1(v, \boldsymbol{w})U_1 + \cdots + q_m(v, \boldsymbol{w})U_m\big) - 1 \in {}_{\mathbb{O}}\langle G\rangle \subseteq \operatorname{Ann}_{\mathbb{O}} f. \tag{3.7}$$

This operator can be seen as an ansatz for the unknown functions q_i. Sufficiently many equations for determining them are obtained by using the condition that (3.7) has to lie in the left ideal generated by G. This is the case if and only if its normal form modulo G is zero. At this point it is now clear how the algorithm proceeds: It starts with the ansatz (3.7) and reduces it with the Gröbner basis G; a system of equations is obtained by equating the coefficients of the normal form to zero. If this system admits rational function solutions for the q_i then we have found an antidifference (resp. antiderivative), otherwise it does not exist within our search space. The only step that deserves a little more attention is to look at what kind of system appears in the algorithm and how to solve it. The fact that in the ansatz the Ore operator ∂_v is on the left forces some commutations in order to transform the ansatz to standard representation (i.e., to bring all Ore operators to the right). Due to the commutation rules

$$\begin{aligned}
\Delta_v q_i(v, \boldsymbol{w}) &= q_i(v+1, \boldsymbol{w})\Delta_v + q_i(v+1, \boldsymbol{w}) - q_i(v, \boldsymbol{w}),\\
D_v q_i(v, \boldsymbol{w}) &= q_i(v, \boldsymbol{w})D_v + \frac{\partial}{\partial v} q_i(v, \boldsymbol{w}),
\end{aligned}$$

shifts (resp. derivatives) of the unknown rational functions are introduced. Hence we end up with a coupled system of difference (resp. differential) equations. It is not difficult to see that this system will be always linear and of first order. Fortunately there are algorithms how to find all rational solutions of such a system: Either by direct methods as proposed by Abramov and Barkatou [3, 11], or by uncoupling the system (by Gaussian elimination or special uncoupling algorithms like Zürcher's algorithm [97] or the algorithm by Abramov and Zima [5], see also [37]) and iteratively solving the scalar difference (resp. differential) equations with Abramov's algorithm [1, 2, 4].

Example 3.4. *Let's use Chyzak's algorithm for computing the antiderivative of the Hermite polynomials* $\int H_n(x)\,\mathrm{d}x$*. A* ∂*-finite annihilating ideal for* $H_n(x)$ *is given by its Gröbner basis*

$$G = \{D_x + S_n - 2x, S_n^2 - 2xS_n + 2n + 2\} \subset \mathbb{Q}(n,x)[S_n; S_n, 0][D_x; 1, D_x]$$

from which we can read off the monomials under the staircase $U = \{1, S_n\}$*. Consequently we start with the ansatz* $D_x(q_1(x) + q_2(x)S_n) - 1$ *which in standard representation rewrites to* $q_2(x)D_xS_n + q_1(x)D_x + q_2'(x)S_n + q_1'(x) - 1$*. Reducing it with* G *delivers the two equations*

$$\begin{aligned} q_2'(x) - q_1(x) &= 0, \\ 2nq_2(x) + q_1'(x) + 2xq_1(x) + 2q_2(x) - 1 &= 0. \end{aligned}$$

Substituting the first one into the second gives a scalar equation that yields $q_2(x) = \frac{1}{2(n+1)}$ *with Abramov's algorithm. This gives immediately rise to the solution*

$$\int H_n(x)\,\mathrm{d}x = \frac{H_{n+1}(x)}{2(n+1)}.$$

Definite problems

Doron Zeilberger's celebrated *fast algorithm* [92, 94, 69] for definite hypergeometric summation makes essential use of Gosper's indefinite summation algorithm. In a similar manner, Chyzak's indefinite summation (resp. integration) algorithm can be turned into an algorithm for definite summation (resp. integration). Hence it can be viewed as an extension of Zeilberger's fast algorithm to general ∂-finite functions (and this basically is the title of Chyzak's paper [26]).

Given a proper hypergeometric (i.e. hypergeometric and holonomic) term $f(n,k)$, Zeilberger's fast algorithm computes a recurrence for the definite sum $\sum_{k=a}^{b} f(n,k)$. The first step of this algorithm is to execute Gosper's algorithm. If it does not yield a solution, then a linear combination of f and some of its n-shifts with undetermined coefficients $p_i(n)$, e.g., $p_0(n)f(n,k) + p_1(n)f(n+1,k)$, is tried whether it is indefinitely summable with a parameterized version of Gosper's algorithm that additionally solves for the p_i. The order of these n-shifts is increased until a result is obtained (which by theory is guaranteed to happen finally):

$$p_0(n)f(n,k) + \cdots + p_d(n)f(n+d,k) = g(n,k+1) - g(n,k)$$

for some rational functions $p_i \in \mathbb{Q}(n)$ and $g(n,k) = q(n,k)f(n,k)$ is a rational function multiple of $f(n,k)$.

As in the indefinite case, this idea of creative telescoping can be translated to the ∂-finite setting. As before, let $f(v, \boldsymbol{w})$ denote a ∂-finite function with respect to the Ore algebra $\mathbb{O} = \mathbb{K}(v, \boldsymbol{w})[\partial_v; \sigma_v, \delta_v][\boldsymbol{\partial_w}; \boldsymbol{\sigma_w}, \boldsymbol{\delta_w}]$. Analogously to Zeilberger's fast algorithm, the ansatz for a creative telescoping operator looks like

$$\underbrace{\sum_{\boldsymbol{\alpha} \in A} p_{\boldsymbol{\alpha}}(\boldsymbol{w}) \boldsymbol{\partial}_{\boldsymbol{w}}^{\boldsymbol{\alpha}}}_{\text{principal part}} \quad + \quad \partial_v \cdot \underbrace{\big(q_1(v, \boldsymbol{w}) U_1 + \dots + q_m(v, \boldsymbol{w}) U_m\big)}_{\text{delta part}} \tag{3.8}$$

where A is a finite set of exponent vectors. In each step, indefinite summation (resp. integration) with the algorithm of Section 3.3 is tried. We start with $A = \{(0, \dots, 0)\}$ which corresponds exactly to Chyzak's indefinite algorithm ($p_{\mathbf{0}}$ can be set to 1). If f turns out to be indefinitely summable (resp. integrable) then the algorithm stops otherwise it proceeds with an augmented set A. In the following steps more parameters $p_{\boldsymbol{\alpha}}$ are involved and we have to take care of these when solving the coupled system. But this is no problem since all mentioned algorithms for solving coupled systems, generalize to parameterized systems. The set A here plays the same rôle as the set $T = \{t_1, \dots, t_j\}$ of monomials in the FGLM algorithm (see Figure 2.2) for which a linear dependency between the corresponding normal forms is searched. And exactly in the same systematic way as in the FGLM algorithm, elements are added or removed from A. This strategy ensures that the principal parts form a Gröbner basis of a ∂-finite ideal in the end (if such an ideal exists at all). The existence is guaranteed provided that $f(v, \boldsymbol{w})$ is not only ∂-finite but also holonomic.

It is clear that in the ansatz (3.8) it suffices to restrict the support of the delta part to the monomials under the staircase: For any telescoping operator that exists inside ${}_{\mathbb{O}}\langle G\rangle$, we can reduce the delta part to normal form by the input Gröbner basis G, obtaining a linear combination with rational function coefficients of the monomials $U_1, \dots, U_m$.

Example 3.5. *Since in our running example the summand is hypergeometric, Chyzak's algorithm just specializes to Zeilberger's fast algorithm. The first step reveals that $\binom{n}{k}$ is not definitely summable. In the next step we start with the ansatz $p_0 + p_1 S_n + (S_k - 1)q(k)$. After reducing it with the Gröbner basis $\{G_1, G_2\}$, we obtain the equation*

$$(k-n-1)\big((k-n)q(k+1) + (k+1)q(k)\big) = (k+1)\big((k-n-1)p_0 - (n+1)p_1\big)$$

which admits the solution $q(k) = \frac{ck}{n-k+1}$, $p_0 = -2c$, and $p_1 = c$ for an arbitrary constant c. Once again we have discovered the recurrence $S_n - 2$.

In Section 4.5 a more advanced (non-hypergeometric) example is given where all steps are displayed in detail.

3.4 Polynomial ansatz

The basic idea of what we propose in this section is very simple: We also start with an ansatz in order to find a telescoping operator. But in contrast to Chyzak's algorithm we avoid the expensive uncoupling and solving of difference or differential equations. The method to be presented is applicable to multiple summation and integration problems, so we reuse the notation introduced in Section 3.1. We start with an ansatz that involves the summation and integration variables $\boldsymbol{v}$ polynomially:

$$\underbrace{\sum_{\boldsymbol{\alpha}\in A} p_{\boldsymbol{\alpha}}(\boldsymbol{w})\boldsymbol{\partial}_{\boldsymbol{w}}^{\boldsymbol{\alpha}}}_{=P(\boldsymbol{w},\boldsymbol{\partial}_{\boldsymbol{w}})} \quad + \quad \partial_{v_1} \cdot \underbrace{\sum_{(\boldsymbol{\beta},\boldsymbol{\gamma})\in B} \sum_{\boldsymbol{\mu}\in M} q_{\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\mu}}(\boldsymbol{w})\boldsymbol{v}^{\boldsymbol{\mu}}\boldsymbol{\partial}_{\boldsymbol{v}}^{\boldsymbol{\beta}}\boldsymbol{\partial}_{\boldsymbol{w}}^{\boldsymbol{\gamma}}}_{=Q_1(\boldsymbol{v},\boldsymbol{w},\boldsymbol{\partial}_{\boldsymbol{v}},\boldsymbol{\partial}_{\boldsymbol{w}})} \quad + \quad \dots \tag{3.9}$$

where A, B, and M are finite index sets (structure set). The unknown $p_{\boldsymbol{\alpha}}$ and $q_{\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\mu}}$ to solve for are rational functions in $\boldsymbol{w}$ and they can be computed using pure linear algebra. Recall that in Chyzak's ansatz the unknowns q_i also depended on $\boldsymbol{v}$ which causes the system to be first-order and coupled. The prize that we pay now is that the shape of the ansatz is not at all clear from the beginning: The sets A, B, and M need to be fixed, whereas in Chyzak's algorithm we have to loop only over the support of the principal part. Our approach is similar to the generalization of Sister Celine Fasenmyer's technique [32, 72] that is used in Kurt Wegschaider's `MultiSum` package [90, Section 3.5.2] (which can deal with multiple summations of hypergeometric terms). We proceed by reducing the ansatz with the Gröbner basis G of the given ∂-finite ideal for f, obtaining a normal form representation of the ansatz. Since we wish this relation to be in the ideal, the normal form has to be identically zero. Equating the coefficients of the normal form to zero and performing coefficient comparison with respect to $\boldsymbol{v}$ delivers a linear system for all the unknowns that has to be solved over $\mathbb{K}(\boldsymbol{w})$.

Trying out for which structure set A, B, M the ansatz (3.9) delivers a nontrivial solution can be a time-consuming tedious task. Additionally, once a solution is found it still can happen that it does not fit to our needs: It can well happen that all $p_{\boldsymbol{\alpha}}$ are zero in which case the result is useless. Hence the question is: Can we simplify the search for a good ansatz, for example, by using homomorphic images? Clearly the size of the coefficients can be reduced by computing modulo a prime number if the ground field $\mathbb{K}$ admits that, e.g., $\mathbb{K} = \mathbb{Q}$. But in practice this does not downsize the computational complexity too much—still we have multivariate polynomials in $\boldsymbol{v}$ and $\boldsymbol{w}$ that can grow dramatically during the reduction process, on which we want to focus now. We cannot get rid of the variables $\boldsymbol{v}$ since they are needed later for the coefficient comparison. It is also true that we cannot just plug in some concrete integers for $\boldsymbol{w}$: We would lose the feature of noncommutativity that $\boldsymbol{w}$ share with $\boldsymbol{\partial_w}$. And the noncommutativity plays a crucial rôle during the reduction process, in the sense that omitting it we get a wrong result. Let's have a closer look what happens and recall how the normal form computation works (see Figure 3.2).

Input: operator $R \in \mathbb{O}$, Gröbner basis $G = \{G_1, \dots, G_r\} \subset \mathbb{O}$ Output: normal form of R modulo the left ideal ${}_{\mathbb{O}}\langle G \rangle$

while exists $1 \le i \le r$ with $\operatorname{lm}(G_i) \mid \operatorname{lm}(R)$
   $H := (\operatorname{lm}(R)/\operatorname{lm}(G_i)) \cdot G_i$
   $R := R - (\operatorname{lc}(R)/\operatorname{lc}(H)) \cdot H$
return R

Figure 3.2: Normal form computation (lm and lc denote the leading monomial and the leading coefficient of an operator, respectively)

Note that we do the multiplication of the operator that we want to reduce with in two steps: First multiply by the appropriate power product of Ore operators, and second adjust the leading coefficient. The reason for that is because the first step can change the leading coefficient. Note also that R is never multiplied by anything. This gives rise to a modular version of the normal form computation that does respect the noncommutativity (see Figure 3.3). We define $h : \mathbb{O} \to \mathbb{O}'$ with $\mathbb{O}' = \mathbb{K}'(\boldsymbol{v})[\boldsymbol{\partial_v}; \boldsymbol{\sigma_v}, \boldsymbol{\delta_v}][\boldsymbol{\partial_w}; \boldsymbol{\sigma_w}, \boldsymbol{\delta_w}]$ to be an insertion homomorphism that plugs in some concrete integer values for the variables $\boldsymbol{w}$ and reduces all coefficients modulo a prime number (if this is appropriate regarding $\mathbb{K}$). Also if there are additional parameters in $\mathbb{K}$, if for example $\mathbb{K} = \mathbb{Q}(a_1, a_2, \dots)$, these can be replaced by the homomorphism h. Thus most of the computations are done in the homomorphic image and the coefficient growth is moderate compared to before.

After the modular reduction we have to do coefficient comparison with respect to the remaining variables $\boldsymbol{v}$ an what we get is a matrix with constant entries (usually from a finite field $\mathbb{Z}_p$). Nullspace computation of such matrices is very fast and carries no weight in practice. The modular computations can as well be used to prove that a relation of a certain shape does not exist.

Assume now that we have found a promising structure set by means of modular computations. Before starting the real computation we make the ansatz as small as possible by leaving away all unknowns that are 0 in the modular solution. With very high probability they will be 0 in the final solution too—in the opposite case we will realize this unlikely event since

Input: operator $R \in \mathbb{O}$, Gröbner basis $G = \{G_1, \dots, G_r\} \subset \mathbb{O}$, insertion homomorphism h
Output: modular normal form of R modulo the left ideal ${}_{\mathbb{O}}\langle G \rangle$

$R := h(R)$
while exists $1 \le i \le r$ with $\operatorname{lm}(G_i) \mid \operatorname{lm}(R)$
  $H := h((\operatorname{lm}(R)/\operatorname{lm}(G_i)) \cdot G_i)$
  $R := R - (\operatorname{lc}(R)/\operatorname{lc}(H)) \cdot H$
return R

Figure 3.3: Modular normal form computation

then the system will turn out to be unsolvable. In [90] a method called Verbaeten's completion is used in order to recognize superfluous terms in the ansatz a priori. We were thinking about a generalization of that, but since the modular computation is negligibly short compared to the rest, we don't expect to gain much and do not investigate this idea further.

Other optimizations concern the way how the reduction is performed. With a big ansatz that involves hundreds of unknowns it is nearly impossible to do it in the naive way. The only possibility to achieve the result at reasonable cost is to consider each monomial in the support of the ansatz separately. After having computed the normal forms of all these monomials we can combine them in order to obtain the normal form of the ansatz. Last but not least it pays off to make use of the previously computed normal forms. This means that we sort the monomials that we would like to reduce according to the monomial order in which the Gröbner basis is given. Then for each monomial we have to perform one reduction step and then plug in the normal forms that we have already (since all monomials that occur in the support after the reduction step are smaller).

3.5 Concluding example

To conclude this chapter we want to study an example that is not so trivial as the running example of the previous sections and which shows the different performances of the above described algorithms more clearly. In contrast to the summation example of before we now look at an integration problem that was sent to Peter Paule by Gilbert Strang [83]:

$$\int_{-1}^{1} \left(\frac{P_{2k+1}(x)}{x} \right)^2 \mathrm{d}x = 2 \tag{3.10}$$

where $P_k(x)$ denotes the family of Legendre polynomials. This example somehow closes the circle of our work: It was chosen by Peter Paule in a proposal to prolongate the special research program SFB F013 in Linz which then was granted, providing financial support for our work. Additionally this proposal was one source of inspiration that in the beginning helped us to get familiar with the topic.

Having a closer look at the integral (3.10) we observe that for fixed integer k the integrand $f_k(x) = (P_{2k+1}(x)/x)^2$ is a polynomial that is symmetric around $x = 0$. This tells us already that the integral cannot have natural boundaries: for some derivative we will get that $f_k^{(d)}(-1) = -f_k^{(d)}(1) \neq 0$. An easy calculation shows that $[f_1'(x)]_{x=-1}^{x=1} = [5x(5x^2-3)]_{x=-1}^{x=1} = 20$ for example. Using the ∂-finite closure properties multiplication and substitution, a ∂-finite annihilating ideal for $f_k(x)$ is obtained:

$$
\begin{aligned}
\{\, & 4(k+2)^2(2k+5)^2(4k+5)^2S_k^2 + (4k+5)(4k+7)(4k+9)(x-1)x(x+1)\\
&\quad \times (16k^2x^2 - 8k^2 + 56kx^2 - 28k + 45x^2 - 23)\, D_x -\\
&\quad (4k+7)^2\left(16k^2x^2 - 8k^2 + 56kx^2 - 28k + 45x^2 - 23\right)^2 S_k +\\
&\quad 2(2k+3)(4k+9)(256k^4x^4 - 256k^4x^2 + 48k^4 + 1664k^3x^4 - 1696k^3x^2 +\\
&\quad 340k^3 + 3968k^2x^4 - 4144k^2x^2 + 890k^2 + 4120kx^4 - 4430kx^2 + 1018k +\\
&\quad 1575x^4 - 1750x^2 + 429),\\
& (4k+5)(x-1)x(x+1)D_xS_k + (4k+5)(x-1)x(x+1)D_x -\\
&\quad 4(k+1)\left(4kx^2 - 2k + 5x^2 - 2\right) S_k + 2(2k+3)\left(4kx^2 - 2k + 5x^2 - 3\right),\\
& (4k+5)^2(x-1)^2x^2(x+1)^2D_x^2 +\\
&\quad 2(4k+5)(x-1)x(x+1)\left(8k^2x^2 - 4k^2 + 30kx^2 - 16k + 25x^2 - 14\right) D_x -\\
&\quad 8(k+1)^2(2k+3)^2S_k +\\
&\quad 2(2k+3)\left(8k^3 + 48k^2x^4 - 56k^2x^2 + 36k^2 + 120kx^4 - 142kx^2 + 54k +\right.\\
&\quad \left.75x^4 - 90x^2 + 27\right)\}
\end{aligned}
\tag{3.11}
$$

Example 3.6. *We want to apply Zeilberger's slow algorithm to* (3.10)*. The first step consists in finding an element in the ideal* (3.11) *that is free of the integration variable x. Let's follow the classical approach: switch to a (partially) polynomial Ore algebra $\mathbb{Q}(k)[x][S_k; S_k, 0][D_x; 1, D_x]$ and apply elimination techniques. Although the annihilating ideal looks relatively innocent, the elimination via Gröbner bases turns out to be quite hard. It takes several hours with our implementation* `OreGroebnerBasis` *and the result contains some elements of MegaByte-size. In total there are 4 x-free polynomials each of which can be separated into principal part and delta part. At this point we should check whether the inhomogeneous parts that come from integrating over the delta parts evaluate to zero. Since these delta parts are quite huge, it is a good idea to reduce them with the annihilating ideal* (3.11) *before* *applying them to the integrand and evaluating at the boundaries. Altogether this is quite a cumbersome work and for more complicated integrands this task may easily become infeasible. In our example all inhomogeneous parts vanish and we are left with the* 4 *principal parts as annihilating operators for the integral. We finally compute their greatest common divisor in order to obtain a recurrence of smaller order (which is* 8 *in this case, with coefficients of degree* 26 *in* k*).*

Example 3.7. *Again we want to try* (3.10) *with Zeilberger's slow algorithm but now perform the elimination with the ansatz*

$$\sum_{i+j\leq d} c_{i,j}(k) S_k^i D_x^j$$

proposed at the end of Section 3.1. Looping over the total degree d *one has to go up to* $d = 12$ *until an* x*-free operator is found (its size is about 100 kB and it takes about one minute to compute it with our command* `FindRelation`*). As before the inhomogeneous part evaluates to zero, and the surviving principal part is of order* 7*. It is a multiple of the minimal recurrence* $S_k - 1$*. Comparing this result with Example 3.6 we observe the effect of extension/contraction that prevented us from finding the smallest* x*-free operator. This example however teaches that the smallest recurrence that can be found with Zeilberger's slow algorithm, is of order* 7*.*

The two examples 3.6 and 3.7 have demonstrated that Zeilberger's original algorithm is often not favorable in practice. For the relatively simple integral (3.10) both versions took quite some while to come up with a result which additionally overshoots the minimal recurrence by far. The reason lies in the restriction of the delta part by eliminating the variable x completely. In the following we apply the algorithms that overcome this deficiency.

Example 3.8. *Takayama's algorithm applied to* (3.10) *delivers (after 0.2 seconds) the second order recurrence operator*

$$\begin{aligned} &(k+2)^2(2k+5)^2(4k+5)S_k^2 - \\ &(4k+7)\left(8k^4+56k^3+150k^2+182k+83\right)S_k + \\ &(k+1)^2(2k+3)^2(4k+9). \end{aligned} \tag{3.12}$$

But this result should not be trusted immediately since the integral does not have natural boundaries. Using our command `Takayama` *with the additional option* `Extended -> True` *we can force Takayama's algorithm to propagate the delta parts through the whole computation (which is not very efficient however: in this instance it takes around ten times longer). Again we don't get an inhomogeneous part and the final result is the recurrence* (3.12)*.*

Example 3.9. *Chyzak's algorithm is guaranteed to find the smallest recurrence that exists on the level of the integrand (that need not necessarily coincide with the shortest recurrence for the whole integral). For* (3.10) *it computes the following creative telescoping operator*

$$\underbrace{1-S_k}_{=P(k,S_k)}+\underbrace{D_x\left(\frac{x^2-x^4}{2\left(2k^2+5k+3\right)}D_x+\frac{x}{4k+5}S_k+\frac{-4kx^3+3kx-5x^3+4x}{4k^2+9k+5}\right)}_{=Q(k,x,S_k,D_x)}$$

in about 0.5 seconds with our command `CreativeTelescoping`. *Since now also the delta part is relatively small, it is easily verified that* $[Q \bullet f_k(x)]_{x=-1}^{x=1}$ *evaluates to zero. Hence the principal part* P *annihilates the integral, and together with the first initial value*

$$\int_{-1}^{1}\left(\frac{P_1(x)}{x}\right)^2\mathrm{d}x=\int_{-1}^{1}1\,\mathrm{d}x=2$$

we have found that Strang's integral evaluates to 2 *for all integers* $k\geq 0$.

Example 3.10. *We use the polynomial ansatz described in Section 3.4 for finding a telescoping relation for* (3.10). *We have seen that there is much freedom how to choose the structure set of the ansatz. After trying around a little bit, it turns out that the ansatz*

$$c_1(k)S_k+c_0(k)\quad+\quad D_x\cdot\left(\sum_{i=0}^{3}\sum_{j=0}^{2}\sum_{m=0}^{1}c_{i,j,m}(k)x^iS_k^jD_x^m\right)$$

delivers a result with nontrivial principal part. The computation time for reducing this ansatz with the given annihilating ideal (3.11) *and solving the linear system is about 1 second and yields the telescoping relation*

$$\begin{aligned}&2(k+1)(2k+3)(4k+5)^2(4k+7)(4k+9)^2(S_k-1)+\\&D_x\Big(-(2k+5)(4k+5)(8k^3-4k^2x^2+48k^2-13kx^2+92k-9x^2+55)\,S_k^2D_x-\\&\quad 4(k+1)(k+2)(2k+5)(4k+5)(4k+9)xS_k^2+\\&\quad (4k+7)\left(4k^2+14k+11\right)\left(8k^2+28k+23\right)S_kD_x+\\&\quad (4k+5)(4k+7)(4k+9)x(64k^3x^2-48k^3+336k^2x^2-240k^2+572kx^2-\\&\qquad 388k+315x^2-205)S_k-\\&\quad (k+1)(4k+9)(16k^3+8k^2x^2+72k^2+30kx^2+100k+25x^2+44)D_x-\\&\quad 4(k+1)(k+2)(2k+3)(4k+5)(4k+9)(8k+17)x\Big).\end{aligned}$$

We observe that—up to a polynomial factor—the same principal part as in Chyzak's algorithm (Example 3.9) is obtained. Moreover if this relation is simplified, in particular if the delta part is reduced to normal form with respect to the annihilating ideal of $f_k(x)$, *then we end up with exactly the same telescoping relation as it was output by Chyzak's algorithm.*

Chapter 4

The HolonomicFunctions Package

This chapter is dedicated to the presentation of the software developed in the frame of this thesis. All the algorithms for summation and integration of ∂-finite functions that have been presented in Chapter 3 have been implemented (see their demonstration in Section 4.6). In order to prepare the input for these algorithms, the package provides commands for executing the ∂-finite closure properties, or more generally speaking, for computing annihilating ideals for given functions, see Section 4.3. In the core we had to implement a framework for the (noncommutative) arithmetic of Ore polynomials. Section 4.1 shortly describes the concepts and data types that were introduced for that purpose. An important ingredient in Chyzak's algorithm is to find rational solutions of parameterized coupled systems of differential or difference equations. Since this functionality is not provided by Mathematica, it had to be implemented from scratch and is also included in the package; the corresponding commands are explained in Section 4.5. Finally, and not only because this thesis evolved from RISC that was founded by Bruno Buchberger, noncommutative Gröbner bases play a rôle in the above mentioned algorithms. Section 4.2 describes how they can be computed in Ore algebras using our package `HolonomicFunctions`. A handbook with systematic descriptions of all available commands is in preparation [54].

Besides demonstrating our software, this chapter has a second intention, namely to present typical examples, applications and tasks in which our package can be of help. Emphasis is put on a set of identities involving Bessel functions (that is what Chapter BS is about) that was sent to Peter Paule by Frank Olver who is the Mathematics Editor for the DLMF (Digital Library of Mathematical Functions) project at the National Institute of Standards and Technology. In his e-mail [62] he wrote:

> The writing of DLMF Chapter BS by Leonard Maximon and myself is now largely complete; however, a problem has arisen in connection with about a dozen formulas from Chapter 10 of Abramowitz and Stegun for which we have not yet tracked down proofs, and the author of this chapter, Henry Antosiewicz, died about a year ago. Since it is the editorial policy for the DLMF not to state formulas without indications of proofs, I am hoping that you will be willing to step into the breach and supply verifications by computer algebra methods... I will fax you the formulas later today...

When this request arrived in 2006, all of Olver's problems were solved by members of Peter Paule's group, namely Stefan Gerhold, Manuel Kauers, Carsten Schneider, and Burkhard Zimmermann by using computer algebra packages that were available at that time [39], which was quite some challenge. We will show how this problem set can nowadays be solved with help of `HolonomicFunctions` which reduces the work to be done by a human being even more. We concentrate on the computer algebra aspects that are covered by our package, and allow ourselves to neglect extra arguments (e.g. analytic continuation to the whole complex plane, questions of convergence, etc.). For Olver's problem set, these arguments can be found in [39].

The presentation is organized in a bottom-up fashion. This structure sometimes has the effect that in Section x a certain result is computed in several steps in a somewhat complicated manner, and then in Section $x+1$ it is described how to get the very same result with a one line command—this has to be considered when reading the first sections! On the other hand, if you are impatient and eager to prove lots of special function identities, you may immediately jump to Section 4.6 where the most interesting applications are exhibited, in a way how these problems should be done.

The chapter contains a whole Mathematica session that runs throughout all sections (sometimes variables that have been assigned a value will be used later). The inputs are given exactly in the way how you have to type them into Mathematica and the outputs are displayed exactly as Mathematica gives them back (so do not wonder when stumbling across strange symbols like ⅈ or ⅇ—this is Mathematica's notation for the imaginary unit, respectively the base of the natural logarithm, for example). No computation presented in this chapter takes more than a few seconds. We start the session by loading the package:

```
In[1]:= << HolonomicFunctions.m
```

HolonomicFunctions package by Christoph Koutschan, RISC-Linz, Version 1.0 (25.09.2009) — Type ?HolonomicFunctions for help

4.1 Arithmetic with Ore polynomials

Most of the computations that our package performs are based on the non-commutative arithmetic of Ore polynomials. Recently [67] an innocent-looking integral due to Wallis

$$\int_0^{\frac{\pi}{2}} \cos(x)^{2m}\,\mathrm{d}x$$

has attracted attention. Here it may serve for demonstration since its solution requires some non-standard use of the operator framework which forces us to work on this low level. But we do not want to conceal that Wallis's integral can be computed in a more standard way after performing the obvious substitution.

Although the integrand is not ∂-finite, this integral can be done by a variant of Takayama's algorithm. Of course a certain trick has to be applied, which requires some steps to be done by hand. The integration variable is x which means that we have to work with partial derivatives with respect to that variable. There is another parameter, namely m, for which it seems more natural to act by shift on it. Let's prepare the stage for this algebraic context.

The package `HolonomicFunctions` provides the user with the following Ore operators:

- the partial derivative D_x is entered as `Der[x]` (the command `D` is already occupied by Mathematica and `D[x]` is immediately simplified to `x`),
- the shift operator S_n is entered as `S[n]`,
- the difference operator Δ_n is entered as `Delta[n]`,
- the Euler operator $\theta_x = xD_x$ is entered as `Euler[x]`,
- and the q-shift operator $S_{x,q}$ has to be entered as `QS[x,q^n]`. This means that a new variable x is introduced that represents the expression q^n.

It is recommended to use `Der` instead of `Euler` for differential equations, and `S` instead of `Delta` for recurrences. But this is only for practical reasons since more testing and optimizations were done on this side. We just want to point out that it is also possible to define other kinds of operators, as long as they can be introduced as Ore extensions. The commands `OreSigma`, `OreDelta`, and `OreAction` are serving this purpose.

Before being able to write down a single relation in operator notation we have to address shortly the usage of Ore algebras. They are defined with the command `OreAlgebra`, where everything that is supposed to occur polynomially only is given in the arguments. In the first place this refers to the Ore operators that a particular Ore algebra is equipped with; of course, it does not make much sense to divide by operators like D_x or $S_m^2 - 1$. Let's create an algebra that contains these two Ore operators, and that could serve for example for expressing recurrences and differential equations related to the integrand of Wallis's integral

In[2]:= **OreAlgebra[Der[*x*], S[*m*]]**

Out[2]= $\mathbb{K}(m,x)[D_x;1,D_x][S_m;S_m,0]$

The algebra is displayed using the classical notation for Ore extensions introduced in Chapter 2. In order to construct Ore extensions of a ring instead of a field, as it is the case for example in the Weyl algebra, the following syntax has to be used:

In[3]:= **OreAlgebra[*x*, Der[*x*]]**

Out[3]= $\mathbb{K}[x][D_x;1,D_x]$

In other words, what has to be given as arguments to the command `OreAlgebra` are exactly the generators of the Ore algebra, viewed as a polynomial ring over some coefficient field. Note that the order in which the generators are given prescribes the order of the generators in the standard monomials, and polynomials in the respective algebra will always be converted to canonical form, their monomials being displayed in that order. Thus using the command `OreAlgebra[Der[x],x]`, you can work in the Weyl algebra where D_x is always commuted to the left so that its monomials are of the form $D_x^\alpha x^\beta$.

The return value of `OreAlgebra` is of the type `OreAlgebraObject`, a construct that contains information about how to compute with elements of this algebra: In the first place this is the set of generators of the algebra, but also instructions how to treat the coefficients (e.g., whether they should be kept in expanded or factored form) as well as possible algebraic extensions.

Back to Wallis's integral: Looking at the summand $f(m,x) = \cos(x)^{2m}$ immediately delivers a recurrence in m, namely $f(m+1,x) = \cos(x)^2 f(m,x)$. Differentiating the summand one obtains the simple differential equation $\cos(x)\frac{\mathrm{d}}{\mathrm{d}x} f(m,x) = -2m\sin(x) f(m,x)$. Note that the annihilating ideal generated by these two relations is not ∂-finite in the classical sense: The coefficients are not in $\mathbb{Q}(m,x)$ (and this is the trick that we mentioned above). We now demonstrate how to express these relations in terms of operators and how to manipulate them. The package `HolonomicFunctions` provides a data type `OrePolynomial` that makes computations with Ore polynomials very convenient. Such an object carries the Ore algebra in which it lives as well as the monomial order in which its terms are sorted. It is displayed with the coefficients always being on the left side, the power products ordered according to how the generators are given in the algebra, and the terms ordered by the given monomial order. An Ore polynomial can be transferred to a different algebra or monomial order by means of the commands `ChangeOreAlgebra` and `ChangeMonomialOrder`. There is a simple way to convert relations that are given in traditional notation into operator form using the data type `OrePolynomial`:

In[4]:= **wall = ToOrePolynomial[{f[m + 1, x] == Cos[x]^2 f[m, x], Cos[x] D[f[m, x], x] == −2m Sin[x] f[m, x]}, f[m, x]]**

Out[4]= $\{S_m - \mathrm{Cos}[x]^2,\ \mathrm{Cos}[x]D_x + 2m\,\mathrm{Sin}[x]\}$

When the command `ToOrePolynomial` is used with the above syntax, the relations will be transformed to Ore polynomials in a rational Ore algebra that is generated by the involved Ore operators, and where everything else goes to the coefficient field. In order to see in which algebra these two Ore polynomials live we type

In[5]:= **OreAlgebra[wall]**

Out[5]= $\mathbb{K}(m, x)[S_m; S_m, 0][D_x; 1, D_x]$

Note that the coefficients are not investigated further; on display are only the variables m and x that correspond to the Ore operators of the algebra. The fact that also $\sin(x)$ and $\cos(x)$ occurs (or possibly other parameters) is subsumed in the unspecific symbol $\mathbb{K}$.

Alternatively `ToOrePolynomial` can take relations in operator notation plus (optionally) an explicit Ore algebra and outputs the corresponding `OrePolynomial` objects. The inverse, i.e., converting an `OrePolynomial` object to a standard Mathematica expression being a (commutative) polynomial in the generators of the algebra, can be achieved by using `Normal`:

In[6]:= **ToOrePolynomial[{Cos[x] Der[x] + 2m Sin[x], S[m] − Cos[x]^2}, OreAlgebra[S[m], Der[x]]]**

Out[6]= $\{\mathrm{Cos}[x]D_x + 2m\,\mathrm{Sin}[x],\ S_m - \mathrm{Cos}[x]^2\}$

In[7]:= **Normal /@ %**

Out[7]= $\{\mathrm{Cos}[x]\,\mathrm{Der}[x] + 2m\,\mathrm{Sin}[x],\ \mathrm{S}[m] - \mathrm{Cos}[x]^2\}$

To verify that these two operators indeed annihilate the integrand, we apply them to $\cos(x)^{2m}$ and hope that the result simplifies to zero:

In[8]:= **ApplyOreOperator[wall, Cos[x]^(2m)] // Simplify**

Out[8]= $\{0, 0\}$

We now aim at a recurrence relation for Wallis's integral by applying Takayama's technique. Since we want to do integration with respect to x, the delta part (that we are not going to compute) has the form $D_x Q$ for some operator Q and we have to find a principal part that does neither involve x nor D_x. Note that the variable x appears only in $\sin(x)$ and $\cos(x)$ but not apart. We now should add certain multiples to the elements of `wall` in order to make elimination possible. The highest total degree with respect to $\sin(x)$ and $\cos(x)$ is 2, hence we want to include multiples of the differential relation by $\sin(x)$ as well as by $\cos(x)$. The commands `OrePlus`, `OreTimes`, and `OrePower` serve for such computations:

In[9]:= **OreTimes[Sin[x], wall[[2]]]**

Out[9]= $\mathrm{Sin}[x]\mathrm{Cos}[x]D_x + 2m\,\mathrm{Sin}[x]^2$

Note that it is not necessary to convert the first factor to the `OrePolynomial` data type; it is automatically transformed into such by using the Ore algebra of the other factor. To make this somewhat cumbersome notation simpler, `HolonomicFunctions` uses Upvalues for Ore polynomials: This means that you can use the symbols `+`, `**`, and `^` for doing arithmetic with Ore polynomials; note the double star that stands for `NonCommutativeMultiply`. Using the single star `*` also works, but the result may not be what you want since Mathematica might reorder the factors before calling `OreTimes`. Thus we get a new set of annihilating operators by

In[10]:= **wall = Join[wall, {Sin[x] ** wall[[2]], Cos[x] ** wall[[2]]}]**

Out[10]= $\{S_m - \mathrm{Cos}[x]^2,\ \mathrm{Cos}[x]D_x + 2m\,\mathrm{Sin}[x],\ \mathrm{Sin}[x]\,\mathrm{Cos}[x]D_x + 2m\,\mathrm{Sin}[x]^2,$
$\mathrm{Cos}[x]^2 D_x + 2m\,\mathrm{Sin}[x]\,\mathrm{Cos}[x]\}$

The next step consists in reducing these Ore polynomials with the right ideal generated by D_x. In order to do so, we first commute D_x to the left (which means to include $\sin(x)$ and $\cos(x)$ to the generators of the algebra):

In[11]:= **wall = ChangeOreAlgebra[wall,**
OreAlgebra[Der[x], Sin[x], Cos[x], S[m]]]

Out[11]= $\{-\mathrm{Cos}[x]^2 + S_m,\ D_x\mathrm{Cos}[x] + (2m+1)\,\mathrm{Sin}[x],\ D_x\,\mathrm{Sin}[x]\,\mathrm{Cos}[x] +$
$(2m+1)\,\mathrm{Sin}[x]^2 - \mathrm{Cos}[x]^2,\ D_x\mathrm{Cos}[x]^2 + (2m+2)\,\mathrm{Sin}[x]\,\mathrm{Cos}[x]\}$

Next we set D_x to zero. Do not try to use Mathematica's `ReplaceAll`. Due to the special data type that we use this will give faulty results! Instead use the following command:

In[12]:= **wall = OrePolynomialSubstitute[wall, {Der[x] → 0}]**

Out[12]= $\{-\mathrm{Cos}[x]^2 + S_m,\ (2m+1)\,\mathrm{Sin}[x],\ (2m+1)\,\mathrm{Sin}[x]^2 - \mathrm{Cos}[x]^2,$
$(2m+2)\,\mathrm{Sin}[x]\,\mathrm{Cos}[x]\}$

The above steps somehow let Mathematica forget about the relations between sin and cos since they are now treated as independent elements of the algebra. Hence we shall not forget to include the Pythagorean identity:

In[13]:= **AppendTo [wall, Cos[x]² + Sin[x]² − 1] ;**

Now the task is to combine these Ore polynomials (without multiplying by $\sin(x)$ and $\cos(x)$ any more) in a way that the result is completely free of x. In this particular example it is so simple that we can achieve the elimination by hand:

In[14]:= **wall[[3]] − (2m + 1) ∗∗ wall[[5]]**

Out[14]= $(-2m-2)\,\mathrm{Cos}[x]^2 + (2m+1)$

In[15]:= **% − (2m + 2) ∗∗ wall[[1]]**

Out[15]= $(-2m-2)S_m + (2m+1)$

The last line contains the desired recurrence. It is of order 1 and by computing the initial value for $m = 0$, i.e., $\int_0^{\pi/2} 1\,\mathrm{d}x = \frac{\pi}{2}$, we can readily read off the solution:

$$\int_0^{\frac{\pi}{2}} \cos(x)^{2m}\,\mathrm{d}x = \frac{\pi}{2} \cdot \frac{\left(\frac{1}{2}\right)_m}{m!}.$$

There is one question left that we have to care about: Does Wallis's integral have natural boundaries and are we therefore allowed to apply Takayama's algorithm at all? Does $[P \bullet \cos(x)^{2m}]_0^{\pi/2} = 0$ hold for arbitrary operators P in our Ore algebra? Clearly not: choose $P = 1$ and get $[\cos(x)^{2m}]_0^{\pi/2} = -1$. Fortunately, it is very easy in this example to obtain the delta part: We just have to carry the delta parts of the input relations through the elimination process. The only relation with nontrivial delta part that was used is the third element in `wall`. Hence the delta part of the telescoping relation is $D_x \sin(x)\cos(x)$. We could perform exactly the same computation as before, only omit the step where we substituted $D_x \to 0$. Now it is immediately checked that $[\sin(x)\cos(x)^{2m+1}]_0^{\pi/2} = 0$.

We want to stress that such strange algebras (with $\sin(x)$ and $\cos(x)$ for example) have to be used with caution. If we had tried to solve Wallis's integral with Zeilberger's slow algorithm, we would have encountered the problem that $\mathbb{K}[\sin(x), \cos(x)][D_x; 1, D_x]$ is not a ring of solvable type when we fix an elimination order on the monomials that eliminates $\sin(x)$ and cos(x), e.g., by requiring that these two are lexicographically greater than D_x and by claiming that $\cos(x) \prec \sin(x)$. The relation

$$D_x \cdot \cos(x) = \cos(x) \cdot D_x - \sin(x)$$

witnesses that condition (1.1) is not fulfilled. In this situation Buchberger's algorithm is not guaranteed to terminate.

Without explicit demonstration we want to mention some more commands for treating Ore polynomials whose names are quite self-explanatory: `NormalizeCoefficients`, `LeadingCoefficient`, `LeadingPowerProduct`, `LeadingTerm`, `LeadingExponent`, `OrePolynomialDegree`, `Support`, and `OrePolynomialListCoefficients`. Some of these commands anyway will be used later in this chapter. There are some further constructions (Mathematica Upvalues) to manipulate Ore polynomials on the coefficient level: the commands `Factor`, `Expand`, and `Together` act directly on the coefficients of an Ore polynomial. So if `p` is an Ore polynomial and you want its coefficients to be displayed in factored form, just type `Factor[p]`. In further computations then the coefficients will always be kept in factored form.

4.2 Noncommutative Gröbner bases

We have implemented Buchberger's algorithm [22], incorporating the improvements suggested in [14, Chapter 5.5] which concern a clever use of the chain criterion (Buchberger's second criterion). The product criterion (Buchberger's first criterion) cannot be applied due to noncommutativity. It is well known that the selection strategy for critical pairs can dramatically influence the runtime of Buchberger's algorithm. As default we use the sugar strategy that has been proposed in [40] and that is nowadays considered as one of the best strategies for critical pair selection. However, in certain instances a different strategy may be of advantage; the option `Method` allows the user to choose between the following strategies:

- `"sugar"`: the strategy proposed in [40] that mimics the Gröbner basis computation in a homogeneous ideal.
- `"normal"`: the normal strategy has been proposed by Bruno Buchberger himself; it takes the pair with the smallest lcm of the leading monomials.
- `"elimination"`: a greedy strategy that is designed for elimination problems; it prefers pairs where the total degree of the variables to be eliminated is minimal (in the lcm of the leading monomials).

- "pairsize": pairs of small "size" are treated first; we take as the size of a pair the product of the ByteCounts of the two polynomials.

As a simple introductory example we can use our implementation with the command OreGroebnerBasis in order to confirm our intuition about the non-existence of zero-dimensional ideals in the Weyl algebra that we elaborated in Example 2.5. We can compute the left Gröbner basis of D_x^m and x^n for certain integers m and n and always get $\{1\}$.

In[16]:= **OreGroebnerBasis[{Der[x]^5, x^7}, OreAlgebra[x, Der[x]], MonomialOrder → DegreeLexicographic]**

Out[16]= $\{1\}$

To convince ourselves that there is really a combination of D_x^5 and x^7 that gives 1, we can use the option Extended. It additionally computes a matrix M that transforms the given input polynomials $\{f_1, \ldots, f_k\}$ into the Gröbner basis $\{g_1, \ldots, g_m\}$ via

$$\begin{pmatrix} g_1 \\ \vdots \\ g_m \end{pmatrix} = M \cdot \begin{pmatrix} f_1 \\ \vdots \\ f_k \end{pmatrix}.$$

The output in this case is a list whose first element is the Gröbner basis and whose second element is the matrix M.

In[17]:= **{g, m} = OreGroebnerBasis[{Der[x]^5, x^7}, OreAlgebra[x, Der[x]], Extended → True]**

Out[17]= $$\Big\{\{1\},\ \Big\{\Big\{-\frac{1}{190080}x^{11}D_x^6 - \frac{11}{28800}x^{10}D_x^5 - \frac{55}{5184}x^9D_x^4 - \frac{55}{384}x^8D_x^3 - \frac{55}{56}x^7D_x^2 - \frac{77}{24}x^6D_x - \frac{77}{20}x^5,\ \frac{1}{190080}x^4D_x^{11} - \frac{1}{43200}x^3D_x^{10} + \frac{1}{12960}x^2D_x^9 - \frac{1}{5760}xD_x^8 + \frac{1}{5040}D_x^7\Big\}\Big\}\Big\}$$

In[18]:= **m[[1, 1]] ** Der[x]^5 + m[[1, 2]] ** x^7**

Out[18]= 1

In most cases we will compute in rational Ore algebras. The following example again illustrates that Buchberger's first criterion (product criterion) is not applicable in noncommutative polynomial rings. We study the annihilating ideal for Bessel functions, namely the modified Bessel functions of the first kind $I_n(z)$. The function $I_n(z)$ represents one of the two solutions of the modified Bessel differential equation. It is also well known that $I_n(z)$

satisfies a recurrence in n of order 2. We convert these two relations to Ore polynomials:

In[19]:= **bess = ToOrePolynomial[{z^2 Der[z]^2 + z Der[z] − n^2 − z^2, z S[n]^2 + (2 + 2n) S[n] − z}]**

Out[19]= $\{z^2 D_z^2 + zD_z + (-n^2 - z^2),\ zS_n^2 + (2n+2)S_n - z\}$

(omitting the second argument, the Ore algebra, `HolonomicFunctions` automatically will determine a suitable algebra in which the given expressions can be represented). Buchberger's criterion now would tell us that this is already a Gröbner basis since the leading power products do not have anything in common, or in other words whose lcm equals their product:

In[20]:= **LeadingPowerProduct /@ bess**

Out[20]= $\{D_z^2,\ S_n^2\}$

But this is actually not a Gröbner basis as a quick computation reveals (we do not have to specify the Ore algebra again, since the input consists already of objects of type `OrePolynomial`):

In[21]:= **bess = OreGroebnerBasis[bess]**

Out[21]= $\{-zD_z + zS_n + n,\ -zS_n^2 + (-2n-2)S_n + z\}$

Using the command `OreReduce` it is easily verified that the differential equation is still contained in the ideal generated by the above two elements.

In[22]:= **OreReduce[z^2 Der[z]^2 + z Der[z] − n^2 − z^2, bess]**

Out[22]= 0

The normal form of the differential equation modulo the Gröbner basis is 0 which means that it is contained in the left ideal generated by the elements of `bess`.

It is also possible to compute Gröbner bases in modules (this is for example needed in Takayama's algorithm). The elements of a module are given as follows: Consider a module element of the form $(a_1, \ldots, a_k), a_i \in \mathbb{O}$ for some Ore algebra $\mathbb{O}$. The positions 1 through k will be indicated by introducing "position variables". This can be a set of k extra indeterminates $p_1, \ldots, p_k$ so that the module element is represented by $p_1a_1 + \cdots + p_ka_k$. Alternatively (and preferably) it suffices to introduce only one position variable p and mark the positions by powers of this particular variable. In that case the module element translates to $a_1 + pa_2 + \cdots + p^{k-1}a_k$. For the Gröbner basis computation the positions of these extra variables within the generators of the Ore algebra have to be given by the option `ModuleBasis`. We again have a look at Wallis's integral and this time do the elimination by means of Gröbner bases:

In[23]:= **OreGroebnerBasis[wall, OreAlgebra[Sin[x], Cos[x], S[m]], MonomialOrder → EliminationOrder[2], ModuleBasis → {1, 2}]**

Out[23]= $\{(-2m-2)S_m + (2m+1),\ \mathrm{Sin}[x],\ (-2m-2)\,\mathrm{Cos}[x]^2 + (2m+1),\ \mathrm{Sin}[x]\,\mathrm{Cos}[x],\ (-2m-2)\,\mathrm{Sin}[x]^2 + 1\}$

The first element of the Gröbner basis is the desired x-free recurrence.

For transferring a given Gröbner basis of a zero-dimensional ideal into a Gröbner basis with respect to a different monomial order, there is the algorithm FGLM [33] due to Faugère, Gianni, Lazard, and Mora. Since a similar idea is behind the algorithms for ∂-finite closure properties, we have included it in the package. The function call is as follows (the second argument may be omitted if the Ore algebra for the output does not change):

In[24]:= **FGLM[bess, OreAlgebra[S[n], Der[z]], Lexicographic]**

Out[24]= $\{z^2 D_z^2 + zD_z + (-n^2 - z^2),\ zS_n - zD_z + n\}$

4.3 ∂-finite closure properties

In Section 2.3 we have seen that the class of ∂-finite functions shares a couple of closure properties: sum and product, linear substitutions for discrete variables, algebraic substitutions for continuous variables, and application of an operator. These closure properties can be executed with our package `HolonomicFunctions` using the commands `DFinitePlus`, `DFiniteTimes`, `DFiniteSubstitute`, and `DFiniteOreAction`. Although in practice they will not often be used—for compound expressions the command `Annihilator` automatically executes the closure properties in the background—we want to shortly introduce them. Note that all these commands, as well as others that deal with ∂-finite functions and that will be discussed in the following sections, can only deal with rational Ore algebras (only Ore operators are admissible among its generators). We will give some examples while proving one of Olver's problems, identity (10.2.30) from [6]:

$$\frac{1}{z}\sinh\sqrt{z^2 - 2itz} = \sum_{n=0}^{\infty} \frac{(-it)^n}{n!}\sqrt{\frac{\pi}{2z}}\, I_{-n+\frac{1}{2}}(z). \tag{4.1}$$

By looking at asymptotic expansions of Bessel functions for large orders (see [6, Section 9.3]) it can be proven that the sum on the right-hand side converges only if the condition

$$2|t| < |z| \tag{4.2}$$

is fulfilled. We want to point out that Mathematica is not able to simplify the sum even if this condition is given in the assumptions. An annihilating ideal for the left-hand side can be readily computed using closure properties. Starting from the differential equation for the hyperbolic sine, the algebraic substitution is done:

In[25]:= **DFiniteSubstitute[{ToOrePolynomial[Der[x]^2 − 1]}, {x → Sqrt[z^2 − 2Izt]}, Algebra → OreAlgebra[Der[t], Der[z]]]**

Out[25]= $\{(t+\mathbb{i}z)D_t - zD_z,\ (2t^2z+3\mathbb{i}tz^2-z^3)D_z^2+t^2D_z+(\mathbb{i}t^3-3t^2z-3\mathbb{i}tz^2+z^3)\}$

For continuous substitutions it is important to use different variables for the input and the output; a substitution of the form $x \to \sqrt{1-x^2}$ will not work. Hence also the Ore algebra in which the output shall be represented has to be specified using the option `Algebra`.

It remains to perform the multiplication by $\frac{1}{z}$ and we obtain an annihilating ideal for the left-hand side:

In[26]:= **lhs = DFiniteTimes[%, ToOrePolynomial[{z Der[z] + 1, Der[t]}, OreAlgebra[Der[t], Der[z]]]]**

Out[26]= $\{(t+\mathbb{i}z)D_t - zD_z - 1,\ (2t^2z^2+3\mathbb{i}tz^3-z^4)D_z^2+(5t^2z+6\mathbb{i}tz^2-2z^3)D_z + (\mathbb{i}t^3z-3t^2z^2+t^2-3\mathbb{i}tz^3+z^4)\}$

In order to tackle the right-hand side, we first have to get a handle on the summand, which means computing an annihilating ideal for it. Similarly as before we first perform the discrete substitution for the expression $I_{-n+\frac{1}{2}}(z)$. We take the annihilating ideal for the Bessel function $I_n(z)$ computed earlier. The same command as for continuous substitution is used, only that for discrete substitutions it is possible to use the same variable on both sides of the substitution.

In[27]:= **DFiniteSubstitute[bess, {n → −n + 1/2}]**

Out[27]= $\{2zD_z - 2zS_n + (1-2n),\ zS_n^2 + (2n+1)S_n - z\}$

In the special case where we wish to multiply a ∂-finite function f by a factor that is hypergeometric resp. hyperexponential in all variables under consideration, there is an extra command `DFiniteTimesHyper` that computes the output relations directly without performing the closure property algorithm. Therefore it does not require the input (the annihilating ideal of f) to be a Gröbner basis. This applies to the factor $\frac{(-it)^n}{n!}\sqrt{\frac{\pi}{2z}}$ that is still missing: it is hypergeometric in n and hyperexponential in both t and z. Since now the variable t enters, we have to add an annihilating relation for $I_{-n+\frac{1}{2}}(z)$ with respect to D_t, that is just D_t, and convert everything to Ore polynomials in the same algebra (this can be done using `ToOrePolynomial`):

In[28]:= **smnd = DFiniteTimesHyper[ToOrePolynomial[Append[%, Der[*t*]], OreAlgebra[S[*n*], Der[*t*], Der[*z*]]], (−I*t*)^*n*/*n*! Sqrt[Pi/(2*z*)]]**

Out[28]= $\{(\mathbb{i}nz + \mathbb{i}z)S_n - tzD_z + (nt - t),\ -z^2D_z^2 - 2zD_z + (n^2 - n + z^2), -tD_t + n\}$

This example will be continued in the next section where a detailed demonstration of Chyzak's algorithm is done to solve the summation problem.

Concerning the usage of ∂-finite closure properties the previous calculations have shown that it can be cumbersome to compute an annihilating ideal for a given compound expression for two reasons: First because you have to compute Gröbner bases and repeatedly apply closure properties, basically as often as the number of nodes in the expression tree. Second because you have to know (or to look up) the defining equations for the functions that are contained in the given expression. For some of them it can even be that you are not sure whether they are ∂-finite or not. The command `Annihilator` bypasses all these troubles. It takes an arbitrarily nested mathematical expression and computes an annihilating ideal for it by recursively applying closure properties (this of course works only if the input indeed represents a ∂-finite function—with some exceptions that we will discuss later). Moreover it recognizes a huge collection of special functions and for each of them a set of annihilating operators is stored. If you are interested in the differential equation say for the Whittaker hypergeometric function $W_{k,m}(z)$, you can just "look it up"!

In[29]:= **Annihilator[WhittakerW[*k*, *m*, *z*], Der[*z*], Head → *w*]**

Out[29]= $\{w[z]\left(4kz - 4m^2 - z^2 + 1\right) + 4z^2w''[z]\}$

The second argument, an Ore operator or a list of Ore operators, specifies what kind of relations you are looking for. The option `Head` is more or less a gimmick and causes the output to be given as relations using the given function name. But usually we will omit this option and work with Ore polynomials instead (this is also preferable when you want to do further computations with `HolonomicFunctions`). If we also omit the second argument, `Annihilator` tries to find out himself for which Ore operators relations exist (although in many cases this works quite well, we recommend to specify the Ore operators—this is faster and more reliable).

In[30]:= **Annihilator[WhittakerW[*k*, *m*, *z*]]**

Out[30]= $\{(-2kz + 2mz + z)S_m + (4mz + 2z)D_z + (2kz - 4m^2 - 4m - 1),$
$2S_k + 2zD_z + (2k - z),\ 4z^2D_z^2 + (4kz - 4m^2 - z^2 + 1)\}$

The function `Annihilator` always returns a Gröbner basis with respect to the rational Ore algebra that is generated by the given operators. So we could have achieved the annihilating ideals for the left-hand side and the summand in (4.1) in a single line!

To some extent `Annihilator` can also handle functions that are not ∂-finite. Following the ideas of Chyzak, Kauers, and Salvy [27] we have enlarged `Annihilator`'s database of mathematical functions with some functions that are not ∂-finite, and also the implementation of the closure properties has been adjusted accordingly. Hence we can try the integrand of Wallis's integral:

In[31]:= **Annihilator[Cos[x]^(2m), {S[m], Der[x]}]**

Annihilator::nondf : The expression $\mathrm{Cos}[x]^{2m}$ is not recognized to be ∂-finite. The result will not generate a zero-dimensional ideal.

Out[31]= $\{-S_m D_x^2 + (-4m^2 - 8m - 4)S_m + (4m^2 + 6m + 2)\}$

We obtain only one single relation—too few to generate a zero-dimensional ideal in the Ore algebra $\mathbb{Q}(m,x)[S_m; S_m, 0][D_x; 1, D_x]$; this fact is always indicated by a warning. We also observe that this relation is different from the two operators that we used to solve the integral. The reason is that `Annihilator` only returns operators whose coefficients are rational functions in the variables that correspond to the given Ore operators. In fact the output is normalized in a way that there are no denominators and no content. The dimension of an ideal can be computed as follows:

In[32]:= **AnnihilatorDimension[%]**

Out[32]= 1

which confirms the observation that the ideal was not zero-dimensional.

As an application of `Annihilator` we solve a problem taken from a paper by Stoll and Zeng [82]. The authors define the function

$$U_n(x,a) = \begin{cases} \sqrt{a}^k x U_k\left(\frac{-a+x^2-1}{2\sqrt{a}}\right), & n = 2k+1 \\ \sqrt{a}^k \left(U_k\left(\frac{-a+x^2-1}{2\sqrt{a}}\right) + \sqrt{a}U_{k-1}\left(\frac{-a+x^2-1}{2\sqrt{a}}\right)\right), & n = 2k \end{cases}$$

and are interested in its differential equation with respect to x (separately for even and odd n). In the definition, $U_k(x)$ denotes the Chebyshev polynomial of the second kind. The case $n = 2k+1$ we can solve right away:

In[33]:= **Annihilator[Sqrt[a]^((n − 1)/2) x ChebyshevU[(n − 1)/2, (x^2 − a − 1)/(2 Sqrt[a])], Der[x]]**

Out[33]= $\{(a^2x^2 - 2ax^4 - 2ax^2 + x^6 - 2x^4 + x^2)D_x^2 + (-3a^2x + 6ax + 3x^5 - 3x)D_x + (3a^2 - 6a - n^2x^4 - 2nx^4 + 3)\}$

If we do the same for the case $n = 2k$ we get a much more complicated result:

In[34]:= **Annihilator[Sqrt[a]^{n/2}(ChebyshevU[n/2, (x^2 − a − 1)/(2 Sqrt[a])] +Sqrt[a] ChebyshevU[n/2−1, (x^2−a−1)/(2 Sqrt[a])]), Der[x]]**

Out[34]= $\{(a^4x^3 - 4a^3x^5 - 4a^3x^3 + 6a^2x^7 + 4a^2x^5 + 6a^2x^3 - 4ax^9 + 4ax^7 + 4ax^5 - 4ax^3 + x^{11} - 4x^9 + 6x^7 - 4x^5 + x^3)D_x^4 + (-6a^4x^2 + 4a^3x^4 + 24a^3x^2 + 24a^2x^6 - 4a^2x^4 - 36a^2x^2 - 36ax^8 + 16ax^6 - 4ax^4 + 24ax^2 + 14x^{10} - 36x^8 + 24x^6 + 4x^4 - 6x^2)D_x^3 + (15a^4x - 60a^3x - 2a^2n^2x^5 - 4a^2nx^5 + 6a^2x^5 + 90a^2x + 4an^2x^7 + 4an^2x^5 + 8anx^7 + 8anx^5 - 72ax^7 - 12ax^5 - 60ax - 2n^2x^9 + 4n^2x^7 - 2n^2x^5 - 4nx^9 + 8nx^7 - 4nx^5 + 51x^9 - 72x^7 + 6x^5 + 15x)D_x^2 + (-15a^4 + 60a^3 + 2a^2n^2x^4 + 4a^2nx^4 - 6a^2x^4 - 90a^2 + 8an^2x^6 - 4an^2x^4 + 16anx^6 - 8anx^4 - 24ax^6 + 12ax^4 + 60a - 10n^2x^8 + 8n^2x^6 + 2n^2x^4 - 20nx^8 + 16nx^6 + 4nx^4 + 45x^8 - 24x^6 - 6x^4 - 15)D_x + (n^4x^7 + 4n^3x^7 - 4n^2x^7 - 16nx^7)\}$

It is a differential equation of order 4 and since the one for $n = 2k + 1$ was of order 2, we suspect that there should be one of order 2 for the even case, too. Indeed, by slightly reformulating the input we can achieve this goal. The closure property "application of an operator" that has not been used so far will help. We observe that the two summands in the parentheses are very similar. In fact they can be rewritten as the application of an operator to a single instance of U:

$$U_{\frac{n}{2}}\left(\tfrac{-a+x^2-1}{2\sqrt{a}}\right) + \sqrt{a}\, U_{\frac{n}{2}-1}\left(\tfrac{-a+x^2-1}{2\sqrt{a}}\right) = (S_n^2 + \sqrt{a}) \bullet U_{\frac{n}{2}-1}\left(\tfrac{-a+x^2-1}{2\sqrt{a}}\right).$$

In all cases where such a rewriting can be done it is preferable to use the closure under operator application (since there the vector space dimension of the annihilating ideal cannot grow).

In[35]:= **DFiniteTimes[Annihilator[Sqrt[a]^(n/2), {S[n], Der[x]}], DFiniteOreAction[Annihilator[ChebyshevU[n/2 − 1, (x^2 − a − 1)/(2 Sqrt[a])], {S[n], Der[x]}], S[n]^2 + Sqrt[a]]]**

Out[35]= $\{(-a^3x + a^2nx^3 + 3a^2x^3 + 3a^2x - 2anx^5 - 2anx^3 - 3ax^5 - 2ax^3 - 3ax + nx^7 - 2nx^5 + nx^3 + x^7 - x^5 - x^3 + x)D_x^2 + (a^3 - 3a^2nx^2 + a^2x^2 - 3a^2 + 6anx^2 - 5ax^4 + 6ax^2 + 3a + 3nx^6 - 3nx^2 + 3x^6 + 5x^4 - 7x^2 - 1)D_x + (3an^2x^3 + 6anx^3 - n^3x^5 - 3n^2x^5 - 3n^2x^3 - 2nx^5 - 6nx^3),$
$(2a - 2nx^2 - 2x^2 - 2)S_n^2 + (a^2x - 2ax^3 - 2ax + x^5 - 2x^3 + x)D_x + (-anx^2 - 2ax^2 + 2a + nx^4 - nx^2 + 2x^4 - 2)\}$

The Ore operator S_n has to be included in the Ore algebra because it appears in the second argument of `DFiniteOreAction`. Again this result can be obtained by a single call of `Annihilator`. The operator application is encoded by the name `ApplyOreOperator`:

In[36]:= **% === Annihilator[Sqrt[a]^{n/2} ApplyOreOperator[S[n]^2 + Sqrt[a], ChebyshevU[n/2 − 1, (x^2 − a − 1)/(2 Sqrt[a])]], {S[n], Der[x]}]**

Out[36]= True

Both differential equations agree with the results of Stoll and Zeng who found it by other means (ad-hoc ansatz for the polynomial coefficients).

In the input line 36 it was crucial that the expression given to the command `Annihilator` was not evaluated. Otherwise `ApplyOreOperator` would have delivered the same expression that we had at the beginning. For this reason, `Annihilator` carries the attribute `HoldFirst`. In a similar manner the symbol `D` can be used in the first argument of `Annihilator`; and instead of letting Mathematica evaluate the differentiation the corresponding closure property is carried out.

4.4 Finding relations by ansatz

In Section 3.1 we have investigated how to use an ansatz in order to eliminate variables that are part of the coefficients of Ore polynomials. Actually this method is much more general: by prescribing the structure set we can ask for relations in an ideal that are of a certain shape. We have implemented the command `FindRelation` that makes use of all the optimizations described in Section 3.4. There are several options which serve for specifying the properties of the desired operators.

- `Eliminate` gives a set of variables to be eliminated; this option is used in the coefficient comparison.
- The use of the option `Support` forces to search only for operators with the given support.
- With `Pattern` only those monomials are considered whose exponent vector is matched by the given pattern.

As an example we want to turn to the widely used hypergeometric function ${}_2F_1$ that has been introduced [36] by Carl Friedrich Gauß at the beginning of the nineteenth century. Besides many other important properties of this function, Gauß investigated what he called contiguous functions, nowadays better known as contiguous relations:

> *Functionem* ipsi $F(\alpha, \beta, \gamma, x)$ *contiguam* vocamus, quae ex illa oritur, dum elementum primum, secundum, vel tertium unitate vel augetur vel diminuitur, manentibus tribus reliquis elementis.[1]

[1] As *contiguous function* to $F(\alpha, \beta, \gamma, x)$ we denominate such a function that emerges from the latter by either augmenting or diminishing the first, second, or third element by one whereat the other three elements remain unaffected.

In the following Gauß gives a complete list of all 15 possible such contiguous relations for his function F (that is the ${}_2F_1$ in our present-day notation). What must have been a tedious task at the times where the word "computer" referred to some pitiable human being, we can now achieve in a second. We start with the annihilating ideal for the hypergeometric function with respect the Ore algebra that is generated by the three shift operators:

In[37]:= **ann = Annihilator[Hypergeometric2F1[*a*, *b*, *c*, *x*], {S[*a*], S[*b*], S[*c*]}]**

Out[37]= $\{(bcx - bc)S_b + (abx - acx - bcx + c^2x)S_c + (acx + bc + c^2(-x)),$
$(acx - ac)S_a + (abx - acx - bcx + c^2x)S_c + (ac + bcx + c^2(-x)),$
$(abx - acx - ax - bcx - bx + c^2x + 2cx + x)S_c^2 +$
$(acx + ax + bcx + bx - 2c^2x + c^2 - 3cx + c - x)S_c + (c^2x - c^2 + cx - c)\}$

Then we have to build the set of all supports for the 15 contiguous relations, that correspond to all 2-subsets of the set of the 6 forward and backward shifts. In order to get rid of the inverse shifts we use our command `NormalizeVector` that takes a list of rational functions, multiplies them with the common denominator and removes the common polynomial content. This set is then given to `FindRelation` by means of the option `Support`. For better readability we factor the coefficients:

In[38]:= **supps = NormalizeVector[Append[#, 1]]& /@ Subsets[{S[*a*], 1/S[*a*], S[*b*], 1/S[*b*], S[*c*], 1/S[*c*]}, {2}];**

In[39]:= **ApplyOreOperator[Factor[FindRelation[ann, Support → supps]], *F*[*a*, *b*, *c*, *x*]] // TableForm**

Out[39]= $(-a + c - 1)F[a, b, c, x] + (a(-x) + 2a + bx - c - x + 2)F[a + 1, b, c, x] +$
$(a + 1)(x - 1)F[a + 2, b, c, x]$
$(b - a)F[a, b, c, x] - bF[a, b + 1, c, x] + aF[a + 1, b, c, x]$
$(-b+c-1)F[a, b, c, x]+(a+b-c+1)F[a, b+1, c, x]+a(x-1)F[a+1, b+1, c, x]$
$c(a+bx-cx)F[a, b, c, x]+x(a-c)(b-c)F[a, b, c+1, x]+ac(x-1)F[a+1, b, c, x]$
$-cF[a, b, c, x] + (c - a)F[a, b, c + 1, x] + aF[a + 1, b, c + 1, x]$
$(-a+c-1)F[a, b, c, x]+(a+b-c+1)F[a+1, b, c, x]+b(x-1)F[a+1, b+1, c, x]$
$(a - c + 1)F[a, b + 1, c, x] + (-b + c - 1)F[a + 1, b, c, x] +$
$(x - 1)(a - b)F[a + 1, b + 1, c, x]$
$cF[a, b, c, x] + c(x - 1)F[a + 1, b, c, x] + x(b - c)F[a + 1, b, c + 1, x]$
$(c-a)F[a, b, c+1, x]+c(x-1)F[a+1, b, c, x]+(a+bx-cx)F[a+1, b, c+1, x]$
$(-b + c - 1)F[a, b, c, x] + (ax + b(-x) + 2b - c - x + 2)F[a, b + 1, c, x] +$
$(b + 1)(x - 1)F[a, b + 2, c, x]$
$c(ax+b-cx)F[a, b, c, x]+x(a-c)(b-c)F[a, b, c+1, x]+bc(x-1)F[a, b+1, c, x]$
$-cF[a, b, c, x] + (c - b)F[a, b, c + 1, x] + bF[a, b + 1, c + 1, x]$
$cF[a, b, c, x] + c(x - 1)F[a, b + 1, c, x] + x(a - c)F[a, b + 1, c + 1, x]$
$(c-b)F[a, b, c+1, x]+c(x-1)F[a, b+1, c, x]+(ax+b-cx)F[a, b+1, c+1, x]$
$c(c + 1)(x - 1)F[a, b, c, x] - (c + 1)(-ax - bx + 2cx - c + x)F[a, b, c + 1, x] +$
$x(a - c - 1)(b - c - 1)F[a, b, c + 2, x]$

Modulo backwards shifting, our result looks very much like the table in the original paper—the only difference are two little typos in the formulas 4 and 11 where an x has been forgotten.

4.5 Rational solutions of systems of differential and difference equations

Finding rational solutions of parameterized coupled systems of differential and difference equations is a subtask in Chyzak's algorithm. We show how this task can be carried out with the package `HolonomicFunctions`. We continue with identity (4.1) from Section 4.3 and solve the summation problem with Chyzak's algorithm. Recall that we have already computed an annihilating ideal `smnd` for the summand and that now we have to find telescoping relations of the form $P_i + \Delta_n Q_i$. The P_i are supposed to be free of n and S_n, and we want to find so many of them that their principal parts P_i generate a zero-dimensional ideal. For the Q_i in the ansatz we have to know the monomials under the stairs of the Gröbner basis `smnd`:

In[40]:= **UnderTheStaircase[smnd]**

Out[40]= $\{1, S_n\}$

Chyzak's algorithm loops over the support of the principal parts P_i. To shorten this process, we look at the annihilating ideal `lhs` that has been computed for the left-hand side. Of particular interest is the support of its generators:

In[41]:= **Support[lhs]**

Out[41]= $\{\{D_t, D_z, 1\}, \{D_z^2, D_z, 1\}\}$

Hoping that the annihilating ideal that we are going to find for the sum is the same as that one for the left-hand side, we try our luck with a principal part that has the same support as the first generator of `lhs`. Hence the ansatz looks as follows:

In[42]:= **ansatz = a_3Der[t]+a_2Der[z]+a_1+(S[n]−1) ** (c_1[n]+c_2[n] ** S[n])**

Out[42]= $a_3\mathrm{Der}[t] + a_2\mathrm{Der}[z] + a_1 + (\mathrm{S}[n] - 1) ** (c_2[n] ** \mathrm{S}[n] + c_1[n])$

Bringing the ansatz to canonical form as an Ore polynomial reveals the fact why later a coupled difference system has to be solved.

In[43]:= **ToOrePolynomial[ansatz, OreAlgebra[Der[t], Der[z], S[n]]]**

Out[43]= $c_2[n+1]S_n^2 + a_3 D_t + a_2 D_z + (c_1[n+1] - c_2[n])S_n + (a_1 - c_1[n])$

The commutation of S_n to the right causes shifted instances of the unknowns

c_1 and c_2 to appear. The last computation also shows that the coefficients of an `OrePolynomial` need not be polynomials or rational functions in the respective variables but can be anything, even unspecified functions in these variables.

The next step consists in reducing the ansatz with the annihilating ideal of the summand and setting all the coefficients of the normal form to zero. The command `OrePolynomialListCoefficients` returns the list of nonzero coefficients of an Ore polynomial ordered according to the monomial order.

In[44]:= **OreReduce[ansatz, smnd]**

$$\text{Out[44]=}\ \left(\frac{\mathbb{i}a_2n^2}{(n+2)t}+\frac{3\mathbb{i}a_2n}{(n+2)t}+\frac{2\mathbb{i}a_2}{(n+2)t}+\frac{2\mathbb{i}ntc_2[n+1]}{(n+2)z}+\frac{\mathbb{i}tc_2[n+1]}{(n+2)z}+\frac{nc_1[n+1]}{n+2}-\frac{nc_2[n]}{n+2}+\frac{2c_1[n+1]}{n+2}-\frac{2c_2[n]}{n+2}\right)S_n+\left(\frac{3a_3n^2}{(n^2+3n+2)\,t}+\frac{2a_3n}{(n^2+3n+2)\,t}+\frac{2a_2n^2}{(n^2+3n+2)\,z}-\frac{a_2n}{(n^2+3n+2)\,z}-\frac{2a_2}{(n^2+3n+2)\,z}+\frac{a_1n^2}{n^2+3n+2}+\frac{3a_1n}{n^2+3n+2}+\frac{2a_1}{n^2+3n+2}+\frac{a_3n^3}{(n^2+3n+2)\,t}+\frac{a_2n^3}{(n^2+3n+2)\,z}-\frac{t^2c_2[n+1]}{n^2+3n+2}-\frac{n^2c_1[n]}{n^2+3n+2}-\frac{3nc_1[n]}{n^2+3n+2}-\frac{2c_1[n]}{n^2+3n+2}\right)$$

In[45]:= **eq = Collect[#, {a_1, a_2, a_3, c_1[_], c_2[_]}, Together]& /@ OrePolynomialListCoefficients[%]**

$$\text{Out[45]=}\ \left\{\frac{\mathbb{i}a_2(n+1)}{t}+\frac{\mathbb{i}c_2[n+1](2nt+t)}{(n+2)z}+c_1[n+1]-c_2[n],\ \frac{a_3n}{t}+\frac{a_2(n-1)}{z}+a_1-\frac{t^2c_2[n+1]}{n^2+3n+2}-c_1[n]\right\}$$

This coupled system has to be solved for unknown rational functions $c_1, c_2 \in \mathbb{Q}(n,t,z)$ and for the parameters $a_1, a_2, a_3 \in \mathbb{Q}(t,z)$. The uncoupling here is trivial and can be done by hand.

In[46]:= **eq[[1]] /. $c_1[n+1] \to ((\text{eq}[[2]] + c_1[n])$ /. $n \to n+1)$**

$$\text{Out[46]=}\ \frac{\mathbb{i}a_2(n+1)}{t}+\frac{a_3(n+1)}{t}+\frac{a_2n}{z}+a_1-\frac{t^2c_2[n+2]}{(n+1)^2+3(n+1)+2}+\frac{\mathbb{i}c_2[n+1](2nt+t)}{(n+2)z}-c_2[n]$$

We obtain a scalar difference equation of order 2 for c_2 and have to find rational solutions of it. The command `RSolveRational` does the job by executing Abramov's algorithm [1, 2, 4]. Plugging the solution into one of the original equations immediately gives the solution for c_1.

In[47]:= **sol = RSolveRational[%, $c_2[n]$, ExtraParameters → {a_1, a_2, a_3}]**

Out[47]= $\left\{\left\{c_2[n] \to 0,\ a_1 \to \frac{\mathrm{C}[1]}{z},\ a_2 \to \mathrm{C}[1],\ a_3 \to \frac{\mathrm{C}[1](-t - \mathbb{i}z)}{z}\right\}\right\}$

In[48]:= **eq[[1]] /. c_2[_] → 0 /. First[sol]**

Out[48]= $c_1[n+1] + \frac{\mathbb{i}\,\mathrm{C}[1]\,(n+1)}{t}$

In a similar fashion, `DSolveRational` is designed for finding rational solutions of linear differential equations. The procedures `RSolvePolynomial` and `DSolvePolynomial` perform the subtasks of finding polynomial solutions of linear difference resp. differential equations.

We can verify whether the above computations are correct by plugging the solution into our original ansatz and reducing it with the Gröbner basis:

In[49]:= **ansatz /. First[sol] /. $c_1[n]$ → −I n C[1]/t**

Out[49]= $\frac{\mathrm{C}[1]\,\mathrm{Der}[t]\,(-t - \mathbb{i}z)}{z} + \mathrm{C}[1]\,\mathrm{Der}[z] + (\mathrm{S}[n]-1){*}{*}\left(0{*}{*}\,\mathrm{S}[n] - \frac{\mathbb{i}\,\mathrm{C}[1]\,n}{t}\right) + \frac{\mathrm{C}[1]}{z}$

In[50]:= **OreReduce[%, smnd]**

Out[50]= 0

Indeed we have found a telescoping relation. It remains to investigate the delta part. As we can set the arbitrary constant C[1] to 1 we have to evaluate the expression

$$\left[\frac{n(-i)^{n+1}t^{n-1}}{n!}\sqrt{\frac{\pi}{2z}}I_{-n+\frac{1}{2}}(z)\right]_0^\infty$$

at the boundaries. For $n = 0$ this clearly vanishes, but for $n \to \infty$ this is not so easy to see (also Mathematica is not able to compute this limit). Similar to the statement of the identity (4.1) we have to use asymptotic expansions of Bessel functions together with condition (4.2) to show that the limit tends indeed to 0 (but since our package cannot address such tasks we do not want to go more into detail here). Hence the principal part is an annihilating operator for the sum (in fact the same as found for the left-hand side):

In[51]:= **NormalizeCoefficients[
ToOrePolynomial[a_3Der[t] + a_2Der[z] + a_1] /. First[sol]]**

Out[51]= $(-t - \mathbb{i}z)D_t + zD_z + 1$

Most of the subtasks that have been accomplished above are combined in the command `SolveCoupledSystem`: It takes as input a coupled system of difference and/or differential equations, tries to uncouple it, and in the successful case finds all rational solutions by backwards substitution as we did before. So for finding a second telescoping relation we can do faster (for the principal part in the ansatz we now choose a differential equation of order 2 in z as indicated by the supports of `lhs`):

In[52]:= **SolveCoupledSystem[OrePolynomialListCoefficients[OreReduce[a_3Der[z]^2+a_2Der[z]+a_1+(S[n]−1)**(c_1[n]+c_2[n]**S[n]), smnd]], {c_1, c_2}, {n}, ExtraParameters → {a_1, a_2, a_3}]**

Out[52]=
$$\left\{\left\{c_1[n] \to \frac{\mathrm{C}[1]\left(2n^2t^2 + 3\mathbb{i}n^2tz - n^2z^2 - nt^2 - 3\mathbb{i}ntz + nz^2\right)}{2t^2 + 3\mathbb{i}tz - z^2},\right.\right.$$
$$a_1 \to \frac{\mathrm{C}[1]\left(\mathbb{i}t^3z - 3t^2z^2 + t^2 - 3\mathbb{i}tz^3 + z^4\right)}{2t^2 + 3\mathbb{i}tz - z^2},$$
$$a_2 \to \frac{\mathrm{C}[1]z\left(-5t^2 - 6\mathbb{i}tz + 2z^2\right)}{-2t^2 - 3\mathbb{i}tz + z^2},$$
$$\left.\left. a_3 \to \mathrm{C}[1]z^2,\ \ c_2[n] \to \frac{\mathbb{i}\left(\mathrm{C}[1]n^2z + \mathrm{C}[1]nz\right)}{2t + \mathbb{i}z}\right\}\right\}$$

After clearing the content in the coefficients of the principal part it is apparent that the solution coincides with the second annihilating operator of `lhs`:

In[53]:= **NormalizeVector[{a_3, a_2, a_1}] /. First[%]**

Out[53]= $\left\{-\mathbb{i}t^3z + 3t^2z^2 - t^2 + 3\mathbb{i}tz^3 - z^4, -5t^2z - 6\mathbb{i}tz^2 + 2z^3, -2t^2z^2 - 3\mathbb{i}tz^3 + z^4\right\}$

`SolveCoupledSystem` in principal can also handle mixed difference and differential systems of equations, and solve them in case that they can be uncoupled. Gaussian elimination over Ore polynomial rings is used for the uncoupling and can be easily performed using our `OreGroebnerBasis` implementation (we have to work over a module).

For the special case where we have a coupled either difference or differential system of order 1, there are other algorithms that are designed exactly for such problems. Some of them—like Zürcher's algorithm [97] or an uncoupling algorithm due to Abramov and Zima [5] —have been implemented at RISC by Stefan Gerhold who described them in his thesis [37]. From an early stage of development and for testing purposes the package `HolonomicFunctions` contains a function `SolveOreSys` that can find rational solutions of order 1 systems as they appear in Chyzak's algorithm, using Gerhold's package `OreSys` (but it has to be loaded separately). We also made some efforts to implement the algorithms by Abramov and Barkatou [3, 11]. The commands `DSystemSolvePolynomial` and `DSystemSolveRational` for example compute the polynomial resp. rational solutions of a linear first-order system of differential equations directly. So far only *simple systems* (for the definition see [3, 11]) can be treated; there are several algorithms to transform a system that not simple into a simple one [12, 13] but this seems to be a very cumbersome procedure.

So far we have shown that both sides of identity (4.1) share the same annihilating ideal. To make the proof rigorous we have to compare initial values, which in general can be a tricky and difficult task. So in this example: Since we have a pole for $z = 0$ we cannot just prove the identity on the formal power series level. Additionally we have to consider convergence issues due to condition 4.2. One strategy consists in taking initial values at a different point, e.g., at $t = 0$ and $z = 1$, but one has to argue carefully what exactly then has been proven (compare also our proof of identity (10.2.38) below). We consider such questions as a direction for future research. Another strategy consists in reducing the problem to a univariate one by considering only the ordinary differential equation in t that is contained in both annihilating ideals. Then initial values are compared by treating z as a parameter; by convergence considerations it can then be determined for which $t, z \in \mathbb{C}$ the identity actually holds. This discussion has been carried out explicitly in [39].

4.6 Summation and Integration

We turn to the implementation of algorithms that are designed for solving summation and integration problems: Takayama's and Chyzak's algorithm. Using these algorithms we are going to solve four more of Olver's problems (at least concerning the computer algebra part—we will allow ourselves to leave to the reader some tasks that are not in the scope of our package). All the remaining ones are variations of them and can be solved analogously. In case that you belong to the kind of impatient readers who jumped directly into this section, you may consider to learn the basics about how to use the command `Annihilator` described at the end of Section 4.3 (page 77).

Identity (10.1.52)

First we are going to prove the following identity that was among Olver's problems [6, 10.1.52] where $j_n(z)$ and $\mathrm{Si}(z)$ denote the spherical Bessel function of the first kind and the sine integral, respectively:

$$\sum_{n=0}^{\infty} j_n^2(z) = \frac{\mathrm{Si}(2z)}{2z}. \tag{4.3}$$

The command for executing Chyzak's algorithm is `CreativeTelescoping`. It computes a set of telescoping relations $P_i + EQ_i$ for a given ∂-finite function.

The first argument has to be a mathematical expression (like `Annihilator` it has the attribute `HoldFirst`) or a list of Ore polynomials that form a Gröbner basis of some annihilating ideal. The second argument is the factor E that appears left to the delta part in the telescoping relation. It describes the nature of the problem: For an integration problem w.r.t. x we set $E = D_x$, and if we are tackling a sum that runs over n we set either $E = S_n - 1$ or $E = \Delta_n$. In the third argument the Ore operators that are intended to constitute the principal parts P_i are given.

In[54]:= **ct = CreativeTelescoping[SphericalBesselJ[*n*, *z*]^2, S[*n*] − 1, Der[*z*]]**

Out[54]= $\left\{\{zD_z + 1\},\ \left\{\frac{z}{2}D_z + (n+1)\right\}\right\}$

The only variable that survives the summation is z and hence we get a single telescoping relation $P + (S_n - 1)Q$. The output consists of the principal part P and the delta part Q, and it is easily verified that this telescoping relation lies indeed in the annihilating ideal of the summand:

In[55]:= **OreReduce[ct[[1, 1]] + (S[*n*] − 1) ** ct[[2, 1]],
Annihilator[SphericalBesselJ[*n*, *z*]^2]]**

Out[55]= 0

Next we have to check whether, after summing over the telescoping relation, the delta part $Q \bullet j_n^2(z)$ vanishes.

In[56]:= **test = ApplyOreOperator [ct[[2, 1]], SphericalBesselJ[*n*, *z*]^2]**

Out[56]= $(n+1)\text{SphericalBesselJ}[n,z]^2 + z\left(\frac{1}{2}(\text{SphericalBesselJ}[n-1,z] - \text{SphericalBesselJ}[n+1,z]) - \frac{\text{SphericalBesselJ}[n,z]}{2z}\right)\text{SphericalBesselJ}[n,z]$

We have to evaluate this expression at the summation bounds. For $n \to \infty$ this is not so trivial and Mathematica is not able to compute the limit directly. But we can use the asymptotic expansion for large orders for $J_n(z)$, after rewriting $j_n(z)$ in terms of $J_n(z)$. For $n = 0$ the delta part is nonzero.

In[57]:= **Limit[test
/. SphericalBesselJ[*k*_, *z*] → Sqrt[Pi/(2*z*)] BesselJ[*k* + 1/2, *z*]
/. BesselJ[*k*_, *z*] → (*Ez*/(2*k*))^*k*/Sqrt[2 Pi *k*], *n* → Infinity]**

Out[57]= 0

In[58]:= **delta = −FullSimplify[test /. *n* → 0]**

Out[58]= $-\frac{\text{Sin}[z]\,\text{Cos}[z]}{z}$

This means that we end up with an inhomogeneous equation for the left-hand side. It is homogenized by multiplying (from the left) with an annihilating operator of the inhomogeneous part:

In[59]:= **lhs = First[Annihilator[delta, Der[z]]] ** ct[[1, 1]]**

Out[59]= $z^2 D_z^4 + 7z D_z^3 + (4z^2+9) D_z^2 + 12z D_z + 4$

For the right-hand side an annihilating operator is readily obtained.

In[60]:= **rhs = Annihilator[SinIntegral[2z]/(2z), Der[z]]**

Out[60]= $\{z^2 D_z^3 + 5z D_z^2 + (4z^2+4) D_z + 4z\}$

The operators are not identical but we find out that the order 4 operator for the left-hand side is a left multiple of the other one:

In[61]:= **OreReduce[lhs, rhs]**

Out[61]= 0

Thus by comparing 4 initial values we can complete the proof. We start with the left-hand side, investigating the derivatives of the summand:

In[62]:= **Table[D[SphericalBesselJ[n, z]^2, {z, k}], {k, 0, 3}] // Together**

Out[62]= $\{\text{SphericalBesselJ}[n,z]^2,$

$-\frac{1}{z}(\text{SphericalBesselJ}[n,z](-z\,\text{SphericalBesselJ}[n-1,z]+$

$\text{SphericalBesselJ}[n,z]+z\,\text{SphericalBesselJ}[n+1,z]),$

$\frac{1}{2z^2}(z^2\text{SphericalBesselJ}[n-1,z]^2-2z^2\text{SphericalBesselJ}[n,z]^2+$

⋮

(We have cut the output which takes about one page and continues in a similar fashion.) Since $j_n(0)$ is 0 for all integers $n > 0$ and the smallest shift of n in $j_n(z)$ that occurs is $n-3$, we can safely cut the summation at $n = 3$.

In[63]:= **Limit[Sum[%, {n, 0, 3}], z → 0]**

Out[63]= $\left\{1, 0, -\frac{4}{9}, 0\right\}$

The initial values of the right-hand side match:

In[64]:= **Limit[Table[D[SinIntegral[2z]/(2z), {z, k}], {k, 0, 3}], z → 0]**

Out[64]= $\left\{1, 0, -\frac{4}{9}, 0\right\}$

For reasons of fairness we shall mention that Mathematica evaluates the sum immediately to the expression on the right-hand side of (4.3).

Identity (10.2.38)

The next example is the duplication formula for $K_n(z)$, the modified Bessel function of the second kind, see [6, 10.2.38].

$$K_{n+\frac{1}{2}}(2z) = \frac{n!}{\sqrt{\pi}} z^{n+\frac{1}{2}} \sum_{k=0}^{n} \frac{(-1)^k(2n-2k+1)}{k!(2n-k+1)!} K^2_{n-k+\frac{1}{2}}(z) \qquad (4.4)$$

Mathematica leaves the sum unevaluated. Recapitulating the previous example we realize that still a lot of work had to be done: Evaluating the delta part at the boundaries, computing an annihilating ideal for it, and so on, until we ended up with a differential equation for the sum. In many examples we can save this time: The functionality of `Annihilator` has been extended so that it recognizes also expressions of the form `Sum[...]` or `Integrate[...]` and then internally calls `CreativeTelescoping` and takes care about the boundaries and possible inhomogeneous parts. Hence we just type both sides of identity (4.4) into the computer and get:

In[65]:= **Annihilator[BesselK[n + 1/2, 2z], {S[n], Der[z]}]**

Out[65]= $\{4zS_n + 2zD_z + (-2n-1),\ 4z^2D_z^2 + 4zD_z + (-4n^2 - 4n - 16z^2 - 1)\}$

In[66]:= **Annihilator[n! Pi^(−1/2) z^(n + 1/2) Sum[(−1)^k (2n − 2k + 1) /k!/(2n−k+1)! BesselK[n−k+1/2, z]^2, {k, 0, n}], {S[n], Der[z]}]**

Out[66]= $\{4zS_n + 2zD_z + (-2n-1),\ 4z^2D_z^2 + 4zD_z + (-4n^2 - 4n - 16z^2 - 1)\}$

The only thing that has to be done by hand is comparing initial values. The initial values that have to be taken into account can be read off from the monomials that lie under the stairs of the Gröbner basis of the annihilating ideal. For convenience, we can just take the operators that are returned by `UnderTheStaircase` and apply them to both sides. After that the substitution has to take place.

In[67]:= **uts = UnderTheStaircase[%]**

Out[67]= $\{1, D_z\}$

In[68]:= **ApplyOreOperator[uts, BesselK[n + 1/2, 2z]] /. n → 0 /. z → 1**

Out[68]= $\left\{\frac{\sqrt{\pi}}{2e^2}, -\frac{5\sqrt{\pi}}{4e^2}\right\}$

In[69]:= **ApplyOreOperator[uts, n! Pi^(−1/2) z^(n+1/2)(−1)^k (2n − 2k + 1) /k!/(2n − k + 1)! BesselK[n − k + 1/2, z]^2] /. n → 0 /. k → 0 /. z → 1 // Together**

Out[69]= $\left\{\frac{\sqrt{\pi}}{2e^2}, -\frac{5\sqrt{\pi}}{4e^2}\right\}$

Since $K_n(z)$ has a singularity at $z = 0$ we decided to compare initial values at the point $z = 1$. The ordinary differential equation in z together with the two initial values (for $n = 0$) guarantees a unique solution that is analytic at $z = 1$. From this solution the instances for $n > 0$ can be obtained via the first relation. Hence there exists a uniquely defined function that is equal to the expression on either side of (4.4) in a neighborhood around $z = 1$. The validity of the identity can be extended to a larger region by analytic continuation.

Identity (10.1.48)

We continue with a beautiful formula that connects different kinds of Bessel functions with Legendre polynomials where trigonometric functions appear in the arguments [6, 10.1.48]:

$$J_0(z \sin t) = \sum_{n=0}^{\infty} (4n+1) \frac{(2n)!}{2^{2n}(n!)^2} j_{2n}(z) P_{2n}(\cos t). \tag{4.5}$$

At first glance one might be tempted to say that neither side is ∂-finite. In Section 2.3 we have proven that the class of ∂-finite functions in continuous variables is closed under algebraic substitution. But $z \sin(t)$ is definitely not algebraic. As we already feared we do not get annihilating operators by typing:

In[70]:= **Annihilator[BesselJ[0, *z* Sin[*t*]], {Der[*t*], Der[*z*]}]**

DFiniteSubstitute::algsubs : The substitutions for continuous variables
{z Sin[t]} must be algebraic expressions.
Not all of them are recognized to be algebraic.

Annihilator::nondf : The expression (w.r.t. Der[t], Der[z]) is not recognized to be ∂-finite.
The result will not generate a zero-dimensional ideal.

Out[70]= {}

But in contrast to Wallis's integral where the integration took place with respect to the argument of the cosine, we here can treat t and therefore also $\sin(t)$ and $\cos(t)$ as parameters. This is done by omitting the derivative with respect to t.

In[71]:= **lhs = Annihilator[BesselJ[0, *z* Sin[*t*]], Der[*z*]]**

Out[71]= $\{zD_z^2 + D_z + z\,\mathrm{Sin}^2[t]\}$

This operator was not found in the first try because its coefficients are not rational functions in the given variables t and z, but they are if we consider only z.

Now the summand of the right-hand side has to be investigated. The squared factorial in the denominator causes the summand to be 0 for negative n. Similarly when n goes to infinity, the summand as well as all its possible "derivations" (in the sense of application of an operator from the algebra) tend to zero. In other words, the sum has natural boundaries and we can apply Takayama's algorithm. The corresponding command takes a list of Ore polynomials that constitute a Gröbner basis for some annihilating ideal, and a list of variables that have to be eliminated. In standard applications these are just the summation and integration variables.

```
In[72]:= ann = Annihilator[(4n + 1)(2n)!/2^(2n)/(n!)^2 SphericalBesselJ[2n, z]
             LegendreP[2n, Cos[t]], {S[n], Der[z]}];
In[73]:= rhs = Takayama[ann, {n}]
```

Out[73]= $\{-16z^5D_z^6 - 144z^4D_z^5 + (16z^5\text{Cos}[t]^2 - 48z^5 - 216z^3)D_z^4 + (128z^4\text{Cos}[t]^2 - 288z^4 + 120z^2)D_z^3 + (32z^5\text{Cos}[t]^2 + 152z^3\text{Cos}[t]^2 - 48z^5 - 240z^3 - 45z)D_z^2 + (128z^4\text{Cos}[t]^2 - 80z^2\text{Cos}[t]^2 - 144z^4 + 120z^2 - 45)D_z + (16z^5\text{Cos}[t]^2 + 24z^3\text{Cos}[t]^2 + 45z\text{Cos}[t]^2 - 16z^5 - 24z^3 - 45z)\}$

As it often happens with Takayama's algorithm we do not find the differential equation of minimal order. We have to test whether it is compatible with the left-hand side. We could reduce it with the order 2 differential equation of the left-hand side. Instead, let's plug in the left-hand side and see whether it satisfies the order 6 differential equation:

```
In[74]:= ApplyOreOperator[First[rhs], BesselJ[0, z Sin[t]]] // FullSimplify
```

Out[74]= 0

Finally 6 initial values have to be computed for both sides.

```
In[75]:= Table[D[BesselJ[0, z Sin[t]], {z, k}], {k, 0, 5}] /. z -> 0
```

Out[75]= $\left\{1, 0, -\frac{1}{2}\text{Sin}[t]^2, 0, \frac{3\,\text{Sin}[t]^4}{8}, 0\right\}$

```
In[76]:= Limit[Table[Sum[D[(4n+1)(2n)!/2^(2n)/(n!)^2 SphericalBesselJ[2n, z]
             LegendreP[2n, Cos[t]], {z, k}], {n, 0, 2}], {k, 0, 5}], z -> 0]
```

Out[76]= $\left\{1, 0, -\frac{1}{2}\text{Sin}[t]^2, 0, \frac{3\,\text{Sin}[t]^4}{8}, 0\right\}$

Identity (10.1.41)

As a last example from Olver's problem set we want to mention a class of identities that express the first derivatives of spherical Bessel functions with respect to the discrete parameter n, evaluated at a certain point, in terms of the sine integral and the cosine integral. They are all very similar, so we content ourselves with displaying only one of them [6, 10.1.41] (note that there is a misprint in Abramowitz and Stegun's book: an additional factor $\pi/2$ appears on the right-hand side):

$$\left[\frac{\partial}{\partial\nu} j_\nu(x)\right]_{\nu=0} = \frac{1}{x}\left(\mathrm{Ci}(2x)\sin x - \mathrm{Si}(2x)\cos x\right)$$

First we have to notice that in our framework mixing the discrete or continuous nature of a variable is possible only in rare cases, e.g., for rational functions:

In[77]:= $\mathbf{Annihilator}\left[(a^2-1)/(a^2+1)\right]$

Out[77]= $\{(a^4-1)D_a - 4a,\ (a^4+2a^3+a^2-2a-2)S_a + (-a^4-2a^3-a^2-2a)\}$

But usually it is determined by the input which variables have to be treated as discrete and which ones as continuous. The spherical Bessel function $j_n(z)$ is only ∂-finite with respect to S_n and D_z. There is no other way to make it ∂-finite, or in other words there is no P-finite recurrence for $j_n(z)$ in n. So there is a problem when we want to consider the derivative with respect to a discrete variable: We cannot apply the closure property "application of an operator" since the Ore operator D_n does not play any rôle in the annihilating ideal of $j_n(z)$. Therefore we proceed by rewriting the left-hand side manually in order to fit it into the ∂-finite/holonomic framework. We replace $j_n(z)$ by $\sqrt{\pi/(2z)}J_{n+1/2}(z)$ and then apply formula (9.1.64) from [6]:

$$\frac{\partial}{\partial\nu} j_\nu(z) = \sqrt{\frac{\pi}{2z}}\left(\log\left(\frac{z}{2}\right)J_{\nu+\frac{1}{2}}(z) - \left(\frac{z}{2}\right)^{\nu+\frac{1}{2}}\sum_{k=0}^{\infty}\frac{\left(-\frac{z^2}{4}\right)^k\psi\left(k+\nu+\frac{3}{2}\right)}{k!\,\Gamma\left(k+\nu+\frac{3}{2}\right)}\right).$$

Now this expression, although it looks quite complicated, can be fed without further changes into `Annihilator`, and the right-hand side anyway is no big challenge for our package:

In[78]:= **lhs = Annihilator[Sqrt[Pi/(2*z*)] (Log[*z*/2] BesselJ[1/2, *z*] −
(*z*/2)^(1/2) Sum[(−*z*^2/4)^*k* PolyGamma[*k* + 3/2]
/*k*!/Gamma[*k* + 3/2], {*k*, 0, Infinity}]), Der[*z*]]**

Out[78]= $\{z^3D_z^6 + 14z^2D_z^5 + (3z^3+52z)D_z^4 + (28z^2+48)D_z^3 + (3z^3+64z)D_z^2 + (14z^2+32)D_z + (z^3+12z)\}$

```
In[79]:= rhs = Annihilator[1/z (CosIntegral[2z] Sin[z] −
             SinIntegral[2z] Cos[z]), Der[z]]
```

Out[79]= $\{(12z^5+5z^3)D_z^6+(144z^4+70z^2)D_z^5+(132z^5+475z^3+260z)D_z^4+(1056z^4+796z^2+240)D_z^3+(228z^5+1991z^3+1288z)D_z^2+(912z^4+1110z^2+560)D_z+(108z^5+753z^3+516z)\}$

Both differential equations are of order 6 and obviously not the same. This situation is different from the examples before where one annihilating ideal was contained in the other (when they did not even coincide). But they have nontrivial common factors, as a gcd computation via Gröbner bases reveals:

```
In[80]:= OreGroebnerBasis[Join[lhs, rhs]]
```

Out[80]= $\{z^3D_z^4+8z^2D_z^3+(2z^3+14z)D_z^2+(8z^2+4)D_z+(z^3+6z)\}$

This is a strong indication that the identity is indeed correct. What we have to do is to test whether the difference of both sides gives zero. We use the closure property sum to compute a differential equation that is satisfied by the difference of the two sides (in fact by any $\mathbb{K}$-linear combination of them):

```
In[81]:= DFinitePlus[lhs, rhs]
```

Out[81]= $\{(48z^5+95z^3)D_z^8+(864z^4+1900z^2)D_z^7+(576z^5+5436z^3+10830z)D_z^6+(7968z^4+23684z^2+17100)D_z^5+(1440z^5+32442z^3+77002z)D_z^4+(13344z^4+59332z^2+83448)D_z^3+(1344z^5+33596z^3+82858z)D_z^2+(6240z^4+31404z^2+46892)D_z+(432z^5+6495z^3+15150z)\}$

It remains to compare 8 initial values which we leave as an exercise to the reader.

Wallis's integral revisited

Let us return to our first example, Wallis's integral. In Section 4.1 we have performed Takayama's algorithm step by step. Of course, everything goes automatically when we use the corresponding command:

```
In[82]:= Takayama[ToOrePolynomial[{
             −Cos[x]^2 + S[m], Cos[x] Der[x] + 2m Sin[x], Sin[x]^2 + Cos[x]^2 − 1
           }], {Sin[x], Cos[x]}, Extended → True]
```

Out[82]= $\{\{(-2m-2)S_m+(2m+1)\},\ \{\{\mathrm{Sin}[x]\,\mathrm{Cos}[x]\}\}\}$

Note that in the second argument we have to give $\sin(x)$ and $\cos(x)$, the two "variables" that we wish to eliminate (usually we would just use x to indicate the integration with respect to that variable). The option `Extended` forces the computation of the delta part in the way it was done in Section 4.1 (which usually is not very efficient).

Alternatively we can use the powerful `Annihilator` command in order to get the recurrence at one stroke, after performing the substitution $u = \cos(x)$:

$$\int_0^{\frac{\pi}{2}} \cos(x)^{2m}\,\mathrm{d}x = \int_0^1 \frac{u^{2m}}{\sqrt{1-u^2}}\,\mathrm{d}u.$$

In[83]:= **Annihilator[Integrate[u^(2m)/Sqrt[1 − u^2], {u, 0, 1}], S[m]]**

Out[83]= $\{(-2m-2)S_m + (2m+1)\}$

Feynman integrals

Since all Bessel function identities sent by Frank Olver consisted of summation problems only, we want to turn to an integration problem now. In the following it is shown how recursive application of Chyzak's algorithm can solve multiple integrals. We want to tie in with a very fruitful cooperation that takes place between the Research Institute for Symbolic Computation (RISC) and Deutsches Elektronen-Synchrotron (DESY). In the latter institution, research on particle physics and big bang theory is done, for which purpose so-called Feynman integrals have to be evaluated. Computer algebra could already significantly contribute in the form of Carsten Schneider's `Sigma` package [78]. We study the double integral

$$\int_0^1\int_0^1 \frac{w^{-1-\varepsilon/2}(1-z)^{\varepsilon/2}z^{-\varepsilon/2}}{(z+w-wz)^{1-\varepsilon}}\left(1-w^{n+1}-(1-w)^{n+1}\right)\mathrm{d}w\,\mathrm{d}z \qquad (4.6)$$

than can be found in [52, (J.17)]. The task is to compute a recurrence in n where ε is just a parameter (that we replace by the symbol e for convenience). We are aware of the fact that (4.6) is not a hard challenge for the physicists, and we use it only as a proof of concept here.

For computing an annihilating ideal for the inner integral, we simply use the command `Annihilator` that takes care of the inhomogeneous part automatically:

In[84]:= **expr = w^(−1 − e/2) (1 − z)^(e/2) z^(−e/2)/(w + z − w z)^(1 − e)
(1 − w^(n + 1) − (1 − w)^(n + 1));**
In[85]:= **ann = Annihilator[Integrate[expr, {w, 0, 1}], {S[n], Der[z]}];**
In[86]:= **UnderTheStaircase[ann]**

Out[86]= $\{1, D_z, S_n, D_z^2, S_n D_z\}$

This result is so big that we do not want to display it here. But it can be used again as input to Chyzak's algorithm, in order to treat the outer integral.

In[87]:= **{{P}, {Q}} = CreativeTelescoping[ann, Der[z], S[n]];**

It is a little bit tricky to handle the inhomogeneous part of the outer integral since it involves an integral itself. In other words, with Q denoting the delta part that was obtained in the previous computation, we have to evaluate

$$\left[Q \bullet \int_0^1 \mathrm{expr}\,\mathrm{d}w\right]_{z=0}^{z=1} = \int_0^1 [Q \bullet \mathrm{expr}]_{z=0}^{z=1}\,\mathrm{d}w. \tag{4.7}$$

It turns out that the right-hand side of (4.7) is preferable. In the following, we first compute the integrand and investigate it further. It contains several expressions of the form 0^a which we can set to zero, provided that the symbolic parameter e is not in $\{-2, 0, 2\}$ (which is a realistic assumption).

In[88]:= **inh = With[{*d* = Together[ApplyOreOperator[*Q*, expr]]}, Together[(*d* /. *z* → 1) − (*d* /. *z* → 0)]];**

In[89]:= **Union[Cases[inh, 0^_, Infinity]]**

Out[89]= $\left\{0^{1-\frac{e}{2}},\ 0^{\frac{e}{2}+1},\ 0^{e/2}\right\}$

In[90]:= **inh /. (0^_) → 0**

Out[90]= 0

We find that the integrand of the inhomogeneous part is zero, hence this whole part vanishes. Therefore the operator P annihilates the double integral, and this is the desired recurrence in n (which is of order 3):

In[91]:= **Factor[*P*]**

Out[91]= $-(e-n-3)(e-n-2)(e+2n+4)(e+2n+6)S_n^3 +$
$(e-n-2)(e+2n+4)(e^2+2en+5e-6n^2-28n-34)S_n^2 -$
$(n+2)(e^3-3e^2n-6e^2-8en^2-30en-28e+12n^3+64n^2+116n+72)S_n -$
$2(n+1)(n+2)^2(e-2n-2)$

Non-holonomic functions

We have already seen in Out[31] that also functions that are not holonomic or ∂-finite can be treated to some extent with our package (although it is named `HolonomicFunctions`!). In Section 2.6 we have mentioned how the closure properties can be performed for functions that do not possess enough relations to be ∂-finite. Now we want to give two examples that are in the spirit of [27]. The handling of initial values is more involved in this setting than for ∂-finite functions, since we have to consider infinitely many initial values.

First, we look at the integral that is given as an example in [27]:

$$\int_0^\infty x^{a-1}\,\mathrm{Li}_n(-xy)\,\mathrm{d}x = \frac{\pi(-a)^n y^{-a}}{\sin(\pi a)}$$

where $\mathrm{Li}_n(x)$ denotes n-th polylogarithm. We start by computing an annihilating ideal for the integrand that is not ∂-finite but of dimension 1.

In[92]:= **ann = Annihilator[x^{a-1} PolyLog[n, $-x\,y$], {S[n], Der[x], Der[y]}]**

Annihilator::nondf : The expression PolyLog[n, -x y] is not recognized to be ∂-finite.
The result will not generate a zero-dimensional ideal.

Out[92]= $\{xD_x - yD_y + (1-a), yS_nD_y - 1\}$

In[93]:= **AnnihilatorDimension[ann]**

Out[93]= 1

Then Chyzak's algorithm, extended according to [27], delivers an ordinary differential equation in y for the whole integral (the delta part vanishes for $-1 < a < 0$, but this has to be established by hand) which is the same as we obtain for the right-hand side.

In[94]:= **CreativeTelescoping[ann, Der[x], {Der[y], S[n]}]**

Out[94]= $\{\{yD_y + a\}, \{-x\}\}$

In[95]:= **Annihilator[Pi $(-a)$^n y^$(-a)$/Sin[a Pi], Der[y]]**

Out[95]= $\{yD_y + a\}$

In this example the initial values are easily compared since both sides give 0 for $y = 0$. The second non-holonomic example is the summation identity

$$\sum_{k=0}^{n} \binom{k}{m} S(n,k)(x-k+m+1)_{k-m} = \sum_{k=0}^{n} \binom{n}{k} S(k,m)x^{n-k}$$

where $(x)_k = x(x+1)\cdots(x+k-1)$ denotes the Pochhammer symbol or rising factorial, and $S(n,k)$ the Stirling numbers of the second kind. The annihilating ideals for the two sums can be computed in a nearly completely automatic fashion. Only the additional information that Mathematica needs for evaluating the inhomogeneous parts, has to be figured out by hand: in the first case it suffices to state that n is a natural number, in the second instance we have to tell Mathematica about the Stirling recurrence—otherwise it does not simplify the inhomogeneous part (which would lead to a far too big annihilating ideal).

In[96]:= **lhs = Annihilator[Sum[Binomial[k, m]
Pochhammer[$x - k + m + 1, k - m$] StirlingS2[n, k], {$k, 0, n$}],
{S[m], S[n]}, Assumptions → Element[n, Integers] && $n \geq 0$]**

Annihilator::nondf : The expression StirlingS2[n, k] is not recognized to be ∂-finite.
The result will not generate a zero-dimensional ideal.

Out[96]= $\{S_mS_n + (-m-x-1)S_m - 1\}$

In[97]:= **rhs = Annihilator[Sum[Binomial[n, k] x^(n − k) StirlingS2[k, m], {k, 0, n}], {S[m], S[n]}, Assumptions → StirlingS2[n, m] + (m + 1) StirlingS2[n, m + 1] − StirlingS2[n + 1, m + 1] == 0]**

Annihilator::nondf : The expression StirlingS2[k, m] is not recognized to be ∂-finite.
The result will not generate a zero-dimensional ideal.

Out[97]= $\{S_m S_n + (-m - x - 1)S_m - 1\}$

The result indicates that we have to compare initial values for $m = 0$ and arbitrary n, as well as for $n = 0$ and arbitrary m. This means that infinitely many cases have to be checked, which can be achieved without problems in this instance. For $m = 0$ we obtain

$$\sum_{k=0}^{n}(x-k+1)_k S(n,k) = x^n = \sum_{k=0}^{n}\binom{n}{k}x^{n-k}$$

and for $n = 0$ the identity reduces to

$$\binom{0}{m}(x+m+1)_{-m} = S(0,m)$$

which is 1 for $m = 0$ and 0 otherwise. This concludes the proof.

4.7 q-Identities

To finish this series of examples we would like to demonstrate how q-identities can be treated with `HolonomicFunctions`. These kinds of identities contain an extra parameter q that is assumed to be a transcendental element, in particular no root of unity. For many classical combinatorial identities there exist q-analogs, i.e., identities that reduce to the classical one when q is set to 1. This aspect of `HolonomicFunctions` continues the tradition of the RISC combinatorics software of always having a strong emphasis on q-calculus. The following identity can be found in [64, Formula (56)]; it is one of the finite forms of the Göllnitz-Gordon identities:

$$\sum_{j=-\infty}^{\infty}(-1)^j q^{4j^2-3j}\begin{bmatrix}2l+1\\ j+l\end{bmatrix}_{q^2} = (q^{2l+2};q^2)_{l+1}\sum_{j=0}^{\infty}\frac{q^{2j^2+2j}}{(-q;q^2)_{j+1}}\begin{bmatrix}l\\ j\end{bmatrix}_{q^2}$$

where $(a;q)_m = (1-a)(1-aq)\cdots(1-aq^{m-1})$ denotes the q-Pochhammer symbol and

$$\begin{bmatrix}n\\ k\end{bmatrix}_q = \frac{(q;q)_n}{(q;q)_k(q;q)_{n-k}}$$

denotes the q-binomial coefficient.

Note that the summations could as well be performed with qZeil [68], an implementation of a q-version of Zeilberger's fast algorithm. For univariate closure properties in the q-world, the package qGeneratingFunctions [47] is available. Since both sums have finite support we can perform creative telescoping without having to care about the delta part (recall that J and L replace the expressions q^j and q^l, respectively:

```
In[98]:= lhs = First[CreativeTelescoping[
              (-1)^j q^(4j^2 - 3j) QBinomial[2l + 1, l + j, q^2],
              QS[J, q^j] - 1, QS[L, q^l]]]
```

Out[98]= $\{S_{L,q}^4 + (L^8q^{33} + L^8q^{31} + L^4q^{20} + L^4q^{18} - q^6 - q^4 - q^2 - 1)S_{L,q}^3 + (L^{16}q^{56} + L^8q^{36} + L^8q^{34} + L^8q^{32} - L^4q^{24} - 2L^4q^{22} - 3L^4q^{20} - 2L^4q^{18} - L^4q^{16} - L^4q^{14} + q^{10} + q^8 + 2q^6 + q^4 + q^2)S_{L,q}^2 + (-L^{20}q^{64} - L^{20}q^{62} + L^{16}q^{54} + L^{16}q^{52} + L^{16}q^{50} + L^{16}q^{48} + L^{12}q^{48} + L^{12}q^{46} + L^{12}q^{44} + L^{12}q^{42} - L^8q^{38} - 2L^8q^{36} - 3L^8q^{34} - 3L^8q^{32} - 3L^8q^{30} - 2L^8q^{28} - L^8q^{26} - L^8q^{24} + L^4q^{26} + 2L^4q^{24} + 3L^4q^{22} + 3L^4q^{20} + 2L^4q^{18} + 2L^4q^{16} + L^4q^{14} - q^{12} - q^{10} - q^8 - q^6)S_{L,q} + (L^{24}q^{66} - L^{20}q^{62} - L^{20}q^{60} - L^{20}q^{58} - L^{20}q^{56} - L^{20}q^{54} - L^{20}q^{52} + L^{16}q^{56} + L^{16}q^{54} + 2L^{16}q^{52} + 2L^{16}q^{50} + 3L^{16}q^{48} + 2L^{16}q^{46} + 2L^{16}q^{44} + L^{16}q^{42} + L^{16}q^{40} - L^{12}q^{48} - L^{12}q^{46} - 2L^{12}q^{44} - 3L^{12}q^{42} - 3L^{12}q^{40} - 3L^{12}q^{38} - 3L^{12}q^{36} - 2L^{12}q^{34} - L^{12}q^{32} - L^{12}q^{30} + L^8q^{38} + L^8q^{36} + 2L^8q^{34} + 2L^8q^{32} + 3L^8q^{30} + 2L^8q^{28} + 2L^8q^{26} + L^8q^{24} + L^8q^{22} - L^4q^{26} - L^4q^{24} - L^4q^{22} - L^4q^{20} - L^4q^{18} - L^4q^{16} + q^{12})\}$

```
In[99]:= ct = First[CreativeTelescoping[q^(2j^2 + 2j)
             QBinomial[l, j, q^2]/QPochhammer[-q, q^2, j + 1],
             QS[J, q^j] - 1, QS[L, q^l]]]
```

Out[99]= $\{(-L^2q^5 - 1)S_{L,q}^2 + (L^4q^8 + L^2q^5 - L^2q^4 + q^2 + 1)S_{L,q} + (L^2q^4 - q^2)\}$

In order to obtain a recurrence for the right hand side, we compute an annihilating ideal of the extra factor $(q^{2l+2}; q^2)_{l+1}$ and use the closure property product:

```
In[100]:= ann = Annihilator[QPochhammer[q^(2l + 2), q^2, l + 1], QS[L, q^l]]
```

Out[100]= $\{S_{L,q} + (L^6q^8 + L^4q^6 - L^2q^2 - 1)\}$

```
In[101]:= rhs = DFiniteTimes[ann, ct]
```

Out[101]= $\{S_{L,q}^2 + (L^8q^{17} + L^6q^{14} - L^6q^{12} + L^4q^{11} + L^4q^{10} - L^4q^9 + L^2q^7 - L^2q^6 - q^2 - 1)S_{L,q} + (-L^{12}q^{21} - L^{10}q^{17} + L^{10}q^{16} + L^8q^{17} + L^8q^{15} + L^8q^{12} + L^6q^{13} - L^6q^{12} + L^6q^{11} - L^6q^{10} - L^4q^{11} - L^4q^8 - L^4q^6 - L^2q^7 + L^2q^6 + q^2)\}$

We end up with two different recurrences. We can now use the closure property sum to get an operator that annihilates either side; it turns out that it is the same as we computed already for the left-hand side. This means that it is a left multiple of the recurrence for the right-hand side.

```
In[102]:= GBEqual[DFinitePlus[lhs, rhs], lhs]
Out[102]= True
In[103]:= OreReduce[lhs, rhs]
Out[103]= {0}
```

Hence we have to compare 4 initial values:

```
In[104]:= Table[Sum[(−1)^j q^(4j^2 − 3j) QBinomial[2l + 1, l + j, q^2],
                    {j, −l, l + 1}] ==
                QPochhammer[q^(2l + 2), q^2, l + 1]
                Sum[q^(2j^2 + 2j) QBinomial[l, j, q^2]
                    /QPochhammer[−q, q^2, j + 1], {j, 0, l}],
                {l, 0, 3}] // FullSimplify
Out[104]= {True, True, True, True}
```

4.8 Interface to Singular/Plural

Computing Gröbner bases is a very costly task, especially in noncommutative polynomial rings. And since this task is only a subproblem in the algorithms that are contained in `HolonomicFunctions`, our main concern was not to make this implementation of Buchberger's algorithm the fastest in the world; using an interpreter language, in our case that of Mathematica, this anyway seems to be a hopeless undertaking. Ergo we have to admit that our `OreGroebnerBasis` cannot compete with special purpose computer algebra systems like Singular [43] and its noncommutative extension Plural [57, 55]. The idea therefore was to provide an interface between `HolonomicFunctions` and Singular/Plural that enables the user to switch to the special purpose system for heavy computations. Manuel Kauers and Viktor Levandovskyy have written such an interface for commutative Gröbner bases and related computations [50]. We have extended this piece of software such that it can also translate objects of type `OrePolynomial` into polynomials in the corresponding noncommutative ring in Plural. This works already fine for computations in purely polynomial algebras like the Weyl algebra.

But in contrast to the Weyl algebra, the noncommutativity in a rational Ore algebra $\mathbb{K}(\boldsymbol{x})[\boldsymbol{\partial}_{\boldsymbol{x}}; \boldsymbol{\sigma}_{\boldsymbol{x}}, \boldsymbol{\delta}_{\boldsymbol{x}}]$ is between the "variables" $\partial_1, \partial_2, \ldots$ of the skew polynomial ring and its coefficients. It was a basic prerequisite in the design of Singular from the very beginning, that the coefficients commute with the variables that generate the polynomial ring.

During two visits to RISC, Viktor Levandovskyy and Hans Schönemann worked hard on an extension of the Singular kernel that enables computations in rational Ore algebras. At the moment a prototype version is tested which will be included into one of the forthcoming Singular distributions. We also plan to include our interface between `HolonomicFunctions` and Singular/Plural into [50].

Chapter 5

Proof of Gessel's Lattice Path Conjecture

In this chapter we want to describe how the methods that were subject of the previous chapters could successfully be used to prove a conjecture about the enumeration of certain lattice paths that was open for several years (Doron Zeilberger named it "the holy grail for lattice-walk-counters" [51]). The results presented below evolved from a collaboration with Manuel Kauers and Doron Zeilberger and have recently been published in the prestigious journal PNAS [49].

5.1 Gessel walks

We start by introducing some basic notions. The objects that we are going to study are walks in the integer lattice $\mathbb{N}^2$ that start at the origin $(0,0)$ and use only unit steps from the step set

$$G := \left\{ \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\} = \{\leftarrow, \rightarrow, \swarrow, \nearrow\}.$$

Of course we demand that a walk does not leave the first quadrant $\mathbb{N}^2$. Such walks we want to call *Gessel walks*; an example is depicted in Figure 5.1.

Throughout this chapter we want to use the symbol $f(n;i,j)$ with $i,j,n \in \mathbb{N}$ to denote the number of Gessel walks that consist of exactly n steps, start at $(0,0)$ and end at the point (i,j).

The conjecture that we are going to prove is about the enumeration of such Gessel walks and has been formulated by Ira Gessel around the year 2001. Unfortunately he did not publish it anywhere, but nevertheless it was spread inside the combinatorial community and several people worked on it.

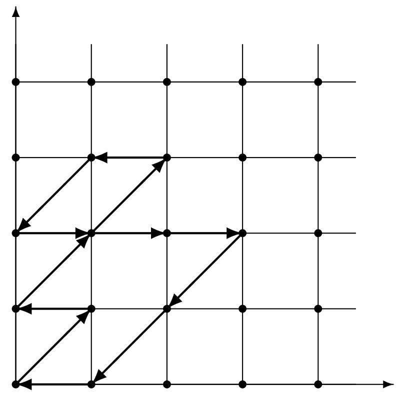

Figure 5.1: Example of a closed Gessel walk

In short Gessel conjectured that $f(n;0,0)$, the number of closed Gessel walks that return to the origin after n steps, can be computed by the innocent-looking formula

$$f(n;0,0) = \begin{cases} 16^k \frac{(5/6)_k(1/2)_k}{(2)_k(5/3)_k} & \text{if } n = 2k \\ 0 & \text{if } n \text{ is odd} \end{cases}$$

5.2 Transfer to the holonomic world

When just reading the statement of Gessel's conjecture it is not immediately clear how the techniques related to the holonomic systems approach can be applied. Before doing so we have to reformulate the problem and the key for that is the quantity $f(n;i,j)$ which is more general than needed to state the conjecture. But this generality makes it easier to characterize it via recurrence relations.

Looking at the steps $\{\leftarrow, \rightarrow, \nearrow, \swarrow\}$ that are allowed in Gessel walks, a very simple recurrence for $f(n;i,j)$ can be readily read off. Consider a certain point in $(i,j) \in \mathbb{N}^2$ and assume that this point was reached with exactly n steps. We can express the number of possible paths by considering all points that are candidates for the previous position (see Figure 5.2):

$$\begin{aligned} f(n;i,j) = \; & f(n-1;i+1,j) + f(n-1;i-1,j) + \\ & f(n-1;i+1,j+1) + f(n-1;i-1,j-1). \end{aligned} \tag{5.1}$$

Having only this single recurrence at hand we are still far away from a ∂-finite description of the function $f(n;i,j)$. Moreover it is not at all clear

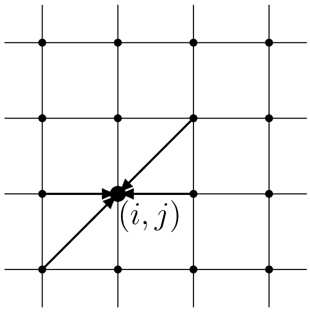

Figure 5.2: Four possibilities to reach the point (i, j) with a single step.

a priori whether such a description exists (in fact, this question was open as well and has been answered only very recently by Alin Bostan and Manuel Kauers [20] who proved that the trivariate generating function of $f(n; i, j)$ is even algebraic!). Also note that with a single operator we cannot do much (no recombination, elimination, summation, etc.). So we wish to have more recurrences and the question is how to find these. The answer is: with *guessing*!

What sounds as a joke at the first glance turns out to be a serious and very powerful technique. The basic ingredient is an ansatz with unspecified coefficients for a linear recurrence with polynomial coefficients. The prerequisite is that sufficiently many values of the sequence in question are known. Plugging them into the ansatz yields a linear system that has to be solved in the end. This method has deserved the name *guessing* for the reason that there is some uncertainty in it: There is no guarantee that guessed recurrences are indeed correct. We only know that they describe our sequence up to the limit to which we provided concrete values. But who knows how the sequence proceeds beyond them? The more values we provide the higher is the probability that a guessed recurrence is correct, but we can never be 100% sure. Summarizing, we have to prove separately that the guessed recurrences are indeed correct!

The guessing was conducted by Manuel Kauers with his powerful Mathematica package `Guess` [46]. The result is a set of 68 recurrences and each of them—except the one from above—fills several lines or even pages. For that reason and to save paper, we do not print them here.

Instead we present a simple procedure how to prove their correctness. It has to be noted that this method is very particular and is applicable only to a very restricted class of problems. Let

$$\mathbb{O} = \mathbb{Q}(n, i, j)[S_n; S_n, 0][S_i; S_i, 0][S_j; S_j, 0]$$

and let

$$T = S_n S_i S_j - S_i^2 S_j - S_j - S_i^2 S_j^2 - 1$$

be the simple recurrence (5.1). Given some operator $R \in \mathbb{O}$ we want to decide whether $R \bullet f(n; i, j) = 0$ or not.

We will argue that it suffices to show that $(TR) \bullet f = 0$, because then it can be algorithmically decided whether also $R \bullet f = 0$. Saying $(TR) \bullet f = 0$ is equivalent to the assertion that $R \bullet f$ satisfies the recurrence T. But $R \bullet f = 0$ if and only if all the initial values $(R \bullet f)(0; i, j)$ are zero. The crucial property with such problems coming from restricted lattice walks is that $f(n; i, j) = 0$ for $i > n$ or $j > n$, so we are left with only finitely many values to check!

By a division with remainder computation we obtain

$$TR = UT + V$$

where the remainder V is of smaller total degree with respect to the variables n, i, j. This can easily be seen by—without loss of generality—consider R to be a single term of the form $p(n, i, j) S_n^a S_i^b S_j^c$. Then

$$TR = p(n+1, i+1, j+1) S_n^{a+1} S_i^{b+1} S_j^{c+1} - p(n, i+2, j+2) S_n^a S_i^{b+2} S_j^{c+2} - \dots$$

Viewing the first displayed term as the leading one (this can be achieved by a lexicographic monomial order for example), the quotient will be

$$U = p(n+1, i+1, j+1) S_n^a S_i^b S_j^c$$

and it is now obvious that the degree in the coefficients of the remainder drops.

Since $(UT) \bullet f = 0$ for sure, we reduced the problem: $TR \bullet f = 0$ if and only if $V \bullet f = 0$. We continue recursively with TV knowing that after finitely many steps we will end up with remainder 0. This completes the recurrence proving algorithm; it is needless to say that we applied it to all 68 guessed candidate recurrences with affirmative result.

5.3 The quasi-holonomic ansatz

Let $I \subset \mathbb{O}$ be the left ideal that is generated by the 68 recurrences that we guessed in the previous section. The quasi-holonomic ansatz proposed in [51] consists in finding an operator $R \in I$ that has the special form

$$R(n, i, j, S_n, S_i, S_j) = P(n, S_n) + iQ_1(n, i, j, S_n, S_i, S_j) + jQ_2(n, i, j, S_n, S_i, S_j). \tag{5.2}$$

By construction, the operator R annihilates $f(n;i,j)$. If we now set $i=j=0$ the only part that will survive is $P(n,S_n)$, which then will be an annihilating operator of $f(n;0,0)$. As soon as we have such a univariate recurrence, it is an easy routine exercise to rigorously proving or disproving Gessel's conjecture.

My colleague Manuel Kauers tried his `Guess` package in order to guess this operator R. By means of modular computations he found out that if it really exists, then R must be bigger than what nowaday's computers could cope with.

5.4 Takayama's algorithm adapted

The fact that by setting $i=j=0$ in equation (5.2) the two parts with Q_1 and Q_2 will vanish suggests to tackle the problem with a Takayama-like approach. Recall that Takayama's algorithm (see Section 3.2) computes the principal part of a telescoping relation without caring about the delta parts. For that reason one has to ensure natural boundaries in order to make the inhomogeneous parts vanish. In the quasi-holonomic ansatz we are in an even more comfortable position: The "delta parts" *will* vanish and we do not have to care about any extra condition.

So instead of computing the full operator R and then setting $i=j=0$, we first substitute $i=j=0$ and then compute the "principal part" P by eliminating S_i and S_j. Hence the only difference to Takayama's original algorithm is that here somehow the rôles of the shift operators S_i and S_j and corresponding variables i and j are interchanged. Still we have to care about the noncommutativity and shall not allow multiplication by either S_i or S_j after the substitution $i=j=0$. As before this is achieved by translating the operators to elements in a module, the positions of whose elements being in correspondence with the power products $S_i^\alpha S_j^\beta$. For example, the simple recurrence (5.1)

$$S_nS_iS_j - S_i^2S_j - S_j - S_i^2S_j^2 - 1$$

translates to

$$(-1,-1,0,0,S_n,0,0,-1,-1)^T$$

when going up to degree 2 in both S_i and S_j. Finally we have to compute a Gröbner basis with position over term ordering to get and element with zero entries except in the first position, corresponding to an operator P free of S_i and S_j.

5.5 Results

Following the lines of the previous section we carried out the necessary computation with our package `HolonomicFunctions`. They are quite time-consuming and therefore we first experimented with modular Gröbner bases, i.e. using the option `Modulus -> p` in `OreGroebnerBasis` we calculate with the coefficient domain $\mathbb{Z}_p(n)$ instead of $\mathbb{Q}(n)$ and get information about which input (a selection of the 68 recurrences) and which pair selection strategy in Buchberger's algorithm deliver the desired $P(n, S_n)$ in shortest time (it turned out that our `elimination` strategy was superior here). Moreover we can record during the modular run, which pairs reduced to zero and omit them in the final computation over $\mathbb{Q}(n)$. Another means to shorten Buchberger's algorithm is to set the option `Incomplete -> True`. The effect is that Buchberger's algorithm is interrupted as soon as an element that matches the desired elimination property. Of course it is not guaranteed then that this element is the minimal one that lies ideal generated by the input. Sometimes however, like in the application that we have in mind here, this does not hurt much. If we have to compare 10 initial values in the end or 20, we do not really care, since we can easily compute the number of Gessel walks for moderate-sized number of steps (for other applications it may well by that this is not the case!).

Finally the computation took around 7 hours and delivered an operator $P(n, S_n)$ annihilating $f(n; 0, 0)$ of impressive size (order 32, polynomial coefficients of degree 172 and therein integers with up to 385 decimal digits). With the computer it is now easily verified that this monstrous recurrence operator also annihilates

$$g(n;0,0) := \begin{cases} 16^k \frac{(5/6)_k (1/2)_k}{(2)_k (5/3)_k} & \text{if } n = 2k \\ 0 & \text{if } n \text{ is odd} \end{cases}$$

We now have to compare initial values, in other words we have to inspect whether $f(n;0,0) = g(n;0,0)$ for $0 \leq n \leq 31$ (this is indeed the case). Finally we have to make sure that the leading coefficient of $P(n, S_n)$ (and all polynomial contents that have been cancelled out during the computation) do not introduce singularities, i.e., do not have positive integer roots. In order to get a handle on the contents that during the run of Buchberger's algorithm are tacitly canceled, the are stored in the variable `GlobalNormalizeFactors`. The fact that all these polynomials have only negative integer roots concludes our rigorous computer proof of Gessel's conjecture.

The natural question that arises is whether this computational effort is really necessary, or whether it should not be possible to prove this simplystated conjecture in a more direct way by hand. Since so far, our proof is the only one, we don't know. But Doron Zeilberger [49] offers a bet on it:

> I offer a prize of one hundred (100) US-dollars for a short, self-contained, human-generated (and computer-free) proof of Gessel's conjecture, not to exceed five standard pages typed in standard font. The longer that prize would remain unclaimed, the more (empirical) evidence we would have that a proof of Gessel's conjecture is indeed beyond the scope of humankind.

5.6 Related conjectures

In 2008 also Marko Petkovšek and Herb Wilf worked on Gessel's conjecture. In their report [71] they state a bunch of conjectures that are related to Gessel's and that we want to investigate in this section.

The first two conjectures of Petkovšek and Wilf concern other special cases of the general path counting function $f(n;i,j)$, namely $f(2n;0,1)$ and $f(2n+1;1,0)$, which count the number of Gessel walks that end at $(0,1)$ or $(1,0)$ respectively. For the first they conjecture the closed form formula

$$f(2n;0,1) = 16^n \frac{(1/2)_n}{(3)_n}\left(\frac{(111n^2+183n-50)\,(5/6)_n}{270(8/3)_n} + \frac{5(7/6)_n}{27(7/3)_n}\right)$$

and for the latter they conjecture that it satisfies the second order recurrence

$$\begin{aligned}&(n+3)(3n+7)(3n+8)\ g(n+1)\\ -8&(2n+3)(18n^2+54n+35)\ g(n)\\ &+256n(3n+1)(3n+2)\ g(n-1) = 0.\end{aligned}$$

We have proven both conjectures in the same way as we did Gessel's. Omitting the details here, we just want to mention that the computations did not become smaller compared to the case $f(n;0,0)$ but grew.

In the same article [71] Marko Petkovšek and Herbert Wilf conjectured that $h(n) := f(2n;2,0)$ is not holonomic. With our approach we were able to disprove this conjecture! In fact, we could prove that $h(n)$ satisfies the recurrence

$$\begin{aligned}0 = {}&4096(n+1)(2n+1)(2n+3)(3n+4)(3n+5)(6n+5)(6n+7)\,p_1(n)\,h(n)\\ &-128(2n+3)\,p_2(n)\,h(n+1) + 48(n+4)\,p_3(n)\,h(n+2)\\ &-8(n+4)(n+5)(3n+13)(3n+14)\,p_4(n)\,h(n+3) +\\ &(n+4)(n+5)(n+6)(3n+13)(3n+14)(3n+16)(3n+17)\,p_5(n)\,h(n+4)\end{aligned}$$

that was guessed by Manuel Kauers with help of the computer. In order to present a clear arrangement, we abbreviated some nasty irreducible factors of higher degree:

$$\begin{aligned}
p_1(n) = {} & 6144n^7 + 130560n^6 + 1169216n^5 + 5718720n^4 + 16490716n^3 + \\
& 28015035n^2 + 25933899n + 10077210, \\
p_2(n) = {} & 31850496n^{13} + 1043103744n^{12} + 15528112128n^{11} + \\
& 139066675200n^{10} + 835537836288n^9 + 3554184658752n^8 + \\
& 11003992594864n^7 + 25083927328960n^6 + 42052581871616n^5 + \\
& 51138759649954n^4 + 43770815405708n^3 + 24915467579665n^2 + \\
& 8429189779675n + 1274964941250, \\
p_3(n) = {} & 15925248n^{13} + 561364992n^{12} + 9001764864n^{11} + 86874808320n^{10} + \\
& 562452019584n^9 + 2576877461856n^8 + 8584177057392n^7 + \\
& 21020268432120n^6 + 37767656881868n^5 + 49065078284877n^4 + \\
& 44671143917844n^3 + 26891118085035n^2 + 9545234776900n + \\
& 1498120123500, \\
p_4(n) = {} & 442368n^{10} + 11612160n^9 + 133731840n^8 + 888142080n^7 + \\
& 3758533024n^6 + 10562908440n^5 + 19901273510n^4 + \\
& 24718969695n^3 + 19263730233n^2 + 8437822050n + 1558180800, \\
p_5(n) = {} & 6144n^7 + 87552n^6 + 514880n^5 + 1616000n^4 + 2911836n^3 + \\
& 2992423n^2 + 1606825n + 341550.
\end{aligned}$$

Chapter 6

Applications in Numerics

A domain which is quite far away from symbolic computation and computer algebra but in which our package `HolonomicFunctions` can be very fruitfully applied, are finite element methods (FEM) in numerical analysis. These methods serve to approximate the solution of a partial differential equation on a given domain $\Omega \subseteq \mathbb{R}^d$ subject to certain constraints, e.g., boundary conditions. The domain Ω is decomposed into a set of *finite elements*, that are just intervals when working in one dimension and that can be triangles or tetrahedra for example when working in dimension 2 or 3 respectively. Using the concept of *weak derivative*—that is a generalization of the derivative in the classical sense to nondifferentiable functions—the original problem is transformed to a *variational formulation*. In order to obtain this formulation the original equation is multiplied by a smooth *test function* v, then integrated over Ω and simplified by partial integration. Two main reasons for doing so are that the order of the partial differential equation is decreased by passing to the weak formulation and that one now can look for solutions in a *Sobolev space* V that admits also non-smooth solutions. After this reformulation usually the task consists in finding solutions $u \in V$ of $a(u, v) = F(v)$ for all $v \in V$, for some bilinear form a and some linear form F. This problem now is discretized by approximating the infinite-dimensional space V by some finite-dimensional space that is spanned by a finite set of *basis functions.* By that the variational formulation translates into a linear system that has to be solved. The basis functions are chosen in a way that they have local support, for example being nonzero only on a small patch of neighboring elements, and therefore they are typically defined to be piecewise polynomial functions. Furthermore they influence the sparsity of the linear system that in the end has to be solved. It turned out that orthogonal polynomials serve this goal perfectly in many instances. This is the point where computer algebra can contribute valuable results: Computing relations for the basis functions may help in building the system matrix, and providing recurrences for the matrix entries can show that most of them are zero. Being not at all an expert in finite element methods we would like to refer to [73] where the interested reader can find an excellent and easy-to-read introduction to the subject (this paragraph also followed this exposition).

6.1 Small examples

A Jacobi polynomial identity

This section contains some identities that are easily derived with our package `HolonomicFunctions` and which are useful for mathematicians working in numerics. Veronika Pillwein was so kind to communicate these formulas to us. We start with a relation for the Jacobi polynomials that can be found in her thesis [73, Formula (4.7)]:

$$(2n+a+b)P_n^{(a-1,b)}(x) = (n+a+b)P_n^{(a,b)}(x) - (n+b)P_{n-1}^{(a,b)}(x).$$

In order to find this relation automatically we have to formulate the problem first, for example: For $P_n^{(a,b)}(x)$ find a mixed recurrence with shifts in a and n, whose coefficients are free of x. This can be seen as an elimination problem: Find an operator in the annihilating ideal of $P_n^{(a,b)}(x)$ that is free of S_b, D_x and x. This problem can be solved by means of Gröbner bases. We have to include the variable x into the generators of the algebra, because otherwise we don't have a handle on it with the monomial order (but recall that because of extension/contraction we are not guaranteed to find the smallest operator or to succeed at all by such an approach):

In[105]:= **jac = Annihilator[JacobiP[n, a, b, x], {S[n], S[a], S[b], Der[x]}]**

Out[105]= $\{(a+b+n+1)S_b + (1-x)D_x + (-a-b-n-1),$
$(a+b+n+1)S_a + (-x-1)D_x + (-a-b-n-1),$
$(2an+2a+2bn+2b+2n^2+4n+2)S_n + (-ax^2+a-bx^2+b-2nx^2+2n-2x^2+2)D_x + (a^2(-x)-a^2-2abx-3anx-an-3ax-a-b^2x+b^2-3bnx+bn-3bx+b-2n^2x-4nx-2x),$
$(x^2-1)D_x^2 + (ax+a+bx-b+2x)D_x + (-an-bn-n^2-n)\}$

In[106]:= **First[OreGroebnerBasis[jac, OreAlgebra[x, S[b], Der[x], S[n], S[a]], MonomialOrder → EliminationOrder[3]]]**

Out[106]= $(-a-b-n-2)S_nS_a + (a+b+2n+3)S_n + (b+n+1)S_a$

In this simple example, elimination via Gröbner basis worked fine, but in general the command `FindRelation` is preferable for such problems:

In[107]:= **FindRelation[jac, Eliminate → {x}, Pattern → {_, _, 0, 0}]**

Out[107]= $\{(-a-b-n-2)S_nS_a+(a+b+2n+3)S_n+(b+n+1)S_a\}$

In the above input line the pattern $\{\_, \_, 0, 0\}$ tells that only exponent vectors are taken into account where the first two positions can contain anything but the last two, which correspond to S_b and D_x, have to be 0. Of course, we could also start differently by computing an annihilating ideal for $P_n^{(a,b)}(x)$ with respect to S_n and S_a only, and then eliminate x.

There is a different way to formulate the problem, namely to ask for an operator with a certain support. We have seen that the relation that we are looking for, involves the monomials S_a, S_n, and S_aS_n. We can specify this condition with the option `Support`:

In[108]:= **FindRelation[jac, Support → {S[a], S[n], S[a] S[n]}]**

Out[108]= $\{(a+b+n+2)S_nS_a+(-a-b-2n-3)S_n+(-b-n-1)S_a\}$

Integrated Jacobi polynomials

As a second example we want to turn to integrated Jacobi polynomials that were introduced in [18, 17] for constructing sparse shape functions for tetrahedral p-FEM. They are defined via

$$\hat{p}_n^a(x) := \int_{-1}^{x} P_{n-1}^{(a,0)}(s)\,\mathrm{d}s. \tag{6.1}$$

In order to find relations for $\hat{p}_n^a(x)$, we first need to compute an annihilating ideal for it. Unfortunately this cannot be achieved completely automatic (by typing `Annihilator` of the integral expression), because Mathematica spends too much time in simplifying the inhomogeneous part. So we have to work a little bit and first compute a creative telescoping relation for (6.1):

In[109]:= **ann = Annihilator[JacobiP[n − 1, a, 0, x], {Der[x], S[n], S[a]}];**
In[110]:= **ct = CreativeTelescoping[ann, Der[x], {S[n], S[a]}]**

Out[110]= $\left\{\{1\}, \left\{\frac{a+n-ax-nx}{n(-1+a+n)}\,S_a+\frac{-2a-n+nx}{n(-1+a+n)}\right\}\right\}$

Now we have to care about the inhomogeneous part. For the lower bound we can easily convince ourselves that $P_n^{(a,0)}(-1)$ does not depend on a. Hence S_a can be replaced by 1 in the delta part and we find that the lower bound does not contribute to the inhomogeneous part.

In[111]:= **Annihilator[JacobiP[n − 1, a, 0, −1]]**

Out[111]= $\{S_n+1, S_a-1\}$

In[112]:= **OrePolynomialSubstitute[ct[[2, 1]], {S[a] → 1, x → −1}]**

Out[112]= 0

Applying the delta part to the integrand (we bypassed the extra substitution $s \to x$) thus gives the inhomogeneous part, and an annihilating ideal for it can be obtained by using the closure property `DFiniteOreAction`. Since the principal part was 1 this ideal does already annihilate the whole integral.

In[113]:= **phat = DFiniteOreAction[ann, ct[[2, 1]]]**

Out[113]= $\{(2+2n)S_n + (a+2n-ax-2nx)S_a + (2-2a-2n),$
$(1+x)D_x + (-a-n)S_a + (-1+a+n),$
$(-1-a-n+x+ax+nx)S_a^2 + (3a+2n-ax-2nx)S_a + (2-2a-2n)\}$

A useful property of the integrated Jacobi polynomial—breakbefores is that they can be written in terms of the original Jacobi polynomials whereat this relation is free of the variable x and shifts are only taken with respect to n (see [17, Formula (12)]). With `FindRelation` this identity is immediately recovered, where we make use of the fact that $D_x \bullet \hat{p}_n^a(x) = P_{n-1}^{(a,0)}(x)$:

In[114]:= **Factor[FindRelation[phat, Eliminate → x, Pattern → {_, _, 0}]]**

Out[114]= $\{2(a+n+1)(a+2n)D_xS_n^2 + 2a(a+2n+1)D_xS_n - 2n(a+2n+2)D_x -$
$(a+2n)(a+2n+1)(a+2n+2)S_n\}$

As a last example of this type we want to derive automatically the recurrence relation for $\hat{p}_n^a(x)$ that is given in [17, Formula (14)], and we observe that the result matches exactly:

In[115]:= **ApplyOreOperator[First[FindRelation[phat, Pattern → {0, _, 0}]], ph[n]] /. n → n − 2 // FullSimplify**

Out[115]= $2(-2+n)(-3+a+n)(-2+a+2n)\,\mathrm{ph}[-2+n] -$
$(-3+a+2n)((-2+a)a + (-4+a+2n)(-2+a+2n)x)\,\mathrm{ph}[-1+n] +$
$2n(-1+a+n)(-4+a+2n)\,\mathrm{ph}[n]$

6.2 Simulation of electromagnetic waves

The rest of this chapter is dedicated to some results that arose from a joint work with Joachim Schöberl (RWTH Aachen). The goal is to simulate the propagation of electromagnetic waves for which he uses *high order discontinuous Galerkin finite elements*. The results of these simulations can be applied for example for constructing antennas of mobile phones, car radios, etc. Some inventions in this context that make use of our symbolically derived formulas, are considered to be registered as a patent.

We want to approximate the electric field $\boldsymbol{E} = (E_x(t), E_y(t), E_z(t))$ and the magnetic field $\boldsymbol{H} = (H_x(t), H_y(t), H_z(t))$ in 3D space. For that purpose we use Maxwell's equations in free space:

$$\begin{aligned}\frac{\partial \boldsymbol{H}}{\partial t} &= -\nabla \times \boldsymbol{E}\\ \frac{\partial \boldsymbol{E}}{\partial t} &= \frac{1}{\mu_0 \varepsilon_0}\nabla \times \boldsymbol{H}\end{aligned}$$

where ε_0 and μ_0 are the electric and the magnetic constant respectively, and where $\nabla \times \boldsymbol{F}$ denotes the *curl operator* (also known as *rotor*) which is defined to be

$$\begin{aligned}\nabla \times \boldsymbol{F} &= \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right) \times (F_x, F_y, F_z)\\ &= \left(\frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z}, \frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x}, \frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y}\right).\end{aligned}$$

As described above, the unknown quantities are approximated by polynomial basis functions using orthogonal polynomials. The basis functions are denoted by $\varphi_{i,j,k}(x, y, z)$, and in this application it turned out to be useful to define them as follows:

$$\begin{aligned}\varphi_{i,j,k}(x,y,z) &= u_i(x,y,z)v_{i,j}(x,y)w_{i,j,k}(x), \quad \text{where}\\ u_i(x,y,z) &= P_i\left(\frac{2z}{(1-x)(1-y)} - 1\right)(1-x)^i(1-y)^i,\\ v_{i,j}(x,y) &= P_j^{(2i+1,0)}\left(\frac{2y}{1-x} - 1\right)(1-x)^j,\\ w_{i,j,k}(x) &= P_k^{(2i+2j+2,0)}(2x-1),\end{aligned} \tag{6.2}$$

where $P_n(x)$ and $P_n^{(a,b)}(x)$ denote the Legendre and Jacobi polynomials, respectively. These basis functions have nice properties, for example being orthogonal on the reference tetrahedron

$$\{(x, y, z) \in \mathbb{R}^3 \mid x \geq 0 \wedge y \geq 0 \wedge z \geq 0 \wedge x + y + z \leq 1\}.$$

6.3 Relations for the basis functions

For setting up the linear system, integrals over products of basis functions and their partial derivatives in all combinations need to be computed, which then yield the entries for the system matrix. Certain rewritings of the basis functions have to be performed during this process. An important step is to express the partial derivatives (with respect to x, y, and z, respectively) in the original basis. In other words, we want to rewrite

$$\frac{\partial}{\partial x}\varphi_{i,j,k}(x,y,z) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}^3} c_{\boldsymbol{\alpha}} \cdot \varphi_{(i,j,k)+\boldsymbol{\alpha}}(x,y,z) \tag{6.3}$$

as a linear combination of the basis functions themselves. The same applies to the derivatives with respect to y and z, but for sake of simplicity we content ourselves with writing explicitly only the first case, the $\frac{\partial}{\partial x}$ derivative. Such relations are easily found by just computing an annihilating ideal for $\varphi_{i,j,k}(x,y,z)$ with our command `Annihilator`:

In[116]:= **phi = (1 − x)^(i + j) (1 − y)^i LegendreP[i, 2z/((1 − x)(1 − y)) − 1] JacobiP[j, 2i+1, 0, 2y/(1−x)−1] JacobiP[k, 2i+2j+2, 0, 2x−1];**

In[117]:= **Timing[ann = Annihilator[phi, {Der[x], S[i], S[j], S[k]}];]**

Out[117]= $\{820.147, \mathrm{Null}\}$

In[118]:= **Support[ann]**

Out[118]= $\{\{S_k^2, S_k, 1\}, \{S_j^2, S_jS_k, S_j, S_k, 1\}, \{S_i^2, S_iS_j, S_iS_k, S_jS_k, S_i, S_j, S_k, 1\},$
$\{D_xS_k, S_iS_j, S_iS_k, S_jS_k, D_x, S_i, S_j, S_k, 1\}, \{D_xS_j, S_iS_j, S_iS_k, S_jS_k, D_x, S_i, S_j, S_k, 1\},$
$\{D_xS_i, S_iS_j, S_iS_k, S_jS_k, D_x, S_i, S_j, S_k, 1\}, \{D_x^2, S_iS_j, S_iS_k, S_jS_k, D_x, S_i, S_j, S_k, 1\},$
$\{S_iS_jS_k, S_iS_j, S_iS_k, S_jS_k, D_x, S_i, S_j, S_k, 1\}\}$

In[119]:= **ByteCount[ann]**

Out[119]= 215677308

The last operator is exactly of the form (6.3). But there is another constraint that we did not mention so far, namely the coefficients $c_{\boldsymbol{\alpha}}$ have to be free of x, y, and z, and therefore may only depend on i, j, and k. It would be too much to expect a relation of the form (6.3) matching this additional condition. Hence we allow more freedom on the left-hand side, namely to have a linear combination of shifted $\frac{\partial}{\partial x}$ derivatives, their coefficients also being free of x, y, and z. More concretely we are now looking for a finitely-supported operator of the form

$$\sum_{(l,m,n)\in\mathbb{N}^3} c_{1,l,m,n}(i,j,k)D_xS_i^lS_j^mS_k^n + \sum_{(l,m,n)\in\mathbb{N}^3} c_{0,l,m,n}(i,j,k)S_i^lS_j^mS_k^n. \tag{6.4}$$

Having implemented noncommutative Gröbner bases, our first attempt was to use them to eliminate the variables x, y, and z. When we start with an annihilating ∂-finite ideal with respect to the Ore algebra

$$\mathbb{Q}(i,j,k,x,y,z)[D_x;1,D_x][S_i;S_i,0][S_j;S_j,0][S_k;S_k,0]$$

we can hope that an operator P with $\deg_{D_x} P = 1$ (matching the form of (6.4)) is found. But we can do better as we will see later. Before tackling this difficult problem we insert an intermezzo by looking at the simpler two-dimensional case.

2D case

Let's for a moment assume that our world is a flat disc (this can also be a realistic assumption if we consider plane waves where the propagation in the third dimension can be neglected). Then we would be interested in simulating the propagation of electromagnetic waves in two dimensions only. In this case the finite elements are just flat triangles and the basis functions are much simpler:

$$\varphi_{i,j}(x,y) = (1-x)^i P_j^{(2i+1,0)}(2x-1) P_i\left(\frac{2y}{1-x} - 1\right).$$

The task is the same as before, namely to find an $\{x,y\}$-free operator that is a 2D-analog of (6.4). The elimination of the two variables can be achieved by Gröbner bases and indeed we find an operator that involves only D_x and no higher derivative. But the computation takes quite long: we interrupted it after more than one day and took the smallest operator in the intermediate basis that fits our needs. Moreover this result seems to be far from being optimal: Its support involves 42 monomials, the shifts are up to order 9 and the coefficients have degrees up to 13 and 14 in i and j, respectively. This somehow was to be expected since with the Gröbner basis approach we suffer from the notorious extension/contraction problem (see Section 2.4, as well as the Examples 2.24 and 3.1).

At the end of Section 3.1 we described a method by ansatz that can deliver such $\{x,y\}$-free operators. Just use equation (6.4) as the ansatz (of course without k and S_k), reduce it with the ∂-finite annihilating ideal, set the coefficients to zero, do coefficient comparison with respect to x and y, and solve the resulting linear system over $\mathbb{Q}(i,j)$! Even more is possible: the degree condition on D_x can be incorporated from the very beginning when creating the ansatz. The last but very important point is that we can make use of the modular techniques that have been described in Section 3.4. Indeed, using our command `FindRelation` that automatically performs these steps, we can find $\{x,y\}$-free operators for both derivatives $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$ in a few seconds. We should mention that the Gröbner basis elimination approach works better if we consider the derivative with respect to y; in this case we get the Gröbner basis in less than a minute and the desired operator is exactly the same as we have found by ansatz.

3D case

Since the times of Copernicus we know that the earth is not a flat disc which leads us back to our original problem in three dimensions. The two-dimensional intermezzo has taught us that it may not be a good idea to try Gröbner basis elimination in the 3D case. Instead we immediately try to find (6.4) by ansatz. We primarily focus on the $\frac{\partial}{\partial x}$ derivative again (the other two give similar results). The modular computations already take about 10 hours and they tell us that the smallest operator of the form (6.4) has the following support (for that task the command `FindSupport` which performs only the modular computations, has been used):

$$\begin{aligned}
\{&S_jS_k^4, S_j^2S_k^3, S_j^3S_k^2, S_j^4S_k, D_xS_jS_k^3, D_xS_j^2S_k^2, D_xS_j^3S_k, D_xS_j^4, S_jS_k^5, S_j^2S_k^4, S_j^3S_k^3,\\
&S_j^4S_k^2, S_iS_k^5, S_iS_jS_k^4, S_iS_j^2S_k^3, S_iS_j^3S_k^2, D_xS_jS_k^4, D_xS_j^2S_k^3, D_xS_j^3S_k^2, D_xS_j^4S_k,\\
&D_xS_iS_k^4, D_xS_iS_jS_k^3, D_xS_iS_j^2S_k^2, D_xS_iS_j^3S_k, S_jS_k^6, S_j^2S_k^5, S_j^3S_k^4, S_j^4S_k^3, S_iS_k^6, S_iS_jS_k^5,\\
&S_iS_j^2S_k^4, S_iS_j^3S_k^3, D_xS_jS_k^5, D_xS_j^2S_k^4, D_xS_j^3S_k^3, D_xS_j^4S_k^2, D_xS_iS_k^5, D_xS_iS_jS_k^4,\\
&D_xS_iS_j^2S_k^3, D_xS_iS_j^3S_k^2, S_jS_k^7, S_j^2S_k^6, S_j^3S_k^5, S_j^4S_k^4, S_iS_k^7, S_iS_jS_k^6, S_iS_j^2S_k^5, S_iS_j^3S_k^4,\\
&D_xS_jS_k^6, D_xS_j^2S_k^5, D_xS_j^3S_k^4, D_xS_j^4S_k^3, D_xS_iS_k^6, D_xS_iS_jS_k^5, D_xS_iS_j^2S_k^4, D_xS_iS_j^3S_k^3,\\
&S_jS_k^8, S_j^2S_k^7, S_j^3S_k^6, S_j^4S_k^5, D_xS_jS_k^7, D_xS_j^2S_k^6, D_xS_j^3S_k^5, D_xS_j^4S_k^4, D_xS_iS_k^7,\\
&D_xS_iS_jS_k^6, D_xS_iS_j^2S_k^5, D_xS_iS_j^3S_k^4, D_xS_jS_k^8, D_xS_j^2S_k^7, D_xS_j^3S_k^6, D_xS_j^4S_k^5, D_xS_iS_k^8,\\
&D_xS_iS_jS_k^7, D_xS_iS_j^2S_k^6, D_xS_iS_j^3S_k^5, D_xS_jS_k^9, D_xS_j^2S_k^8, D_xS_j^3S_k^7, D_xS_j^4S_k^6\}.
\end{aligned}$$

This is impressive but unfortunately rather useless. Even if we succeeded in computing the coefficients (for this purpose homomorphic images and rational reconstruction can be used), the relation is of a size that might not help in speeding up the work of the numerists. But it also illustrates another advantage of the ansatz technique combined with modular computations: In certain instances as it is the case in the current example, we get a lot of information about the expected result at a very early stage. This usually suffices to decide whether it is worth to proceed or whether the computation should be aborted because the result will not match the individual needs.

6.4 Extension of the holonomic framework

As we have seen in the last section, the relations that we in principle could derive are not very adjuvant. Moreover, by having followed the proposed ansatz, we have a proof that no smaller relations (with respect to the total degree of the involved Ore operators) can exist. At this point, Veronika Pillwein and Joachim Schöberl came up with the following idea: Since the basis functions are composed of three factors $\varphi = u \cdot v \cdot w$ (see (6.2)), the product rule delivers

$$\frac{\partial \varphi}{\partial x} = \frac{\partial u}{\partial x} vw + u\frac{\partial v}{\partial x} w + uv\frac{\partial w}{\partial x}.$$

It is now manifest to split the right-hand side and to try to find smaller relations that connect for example uvw and $\frac{\partial u}{\partial x}vw$.

The question is how this fits into the holonomic framework where we handled only equations of the form $P \bullet f = 0$ for some operator P. A relation that contains $f_1 = uvw$ and $f_2 = \frac{\partial u}{\partial x}vw$ cannot be formulated as an operator applied to some function yielding zero (at least not for general u, v, and w). Instead it could be formulated as

$$P_1 \bullet f_1 + P_2 \bullet f_2 = 0. \tag{6.5}$$

More generally we can consider an arbitrary number of functions $f_1, \ldots, f_d$. The natural way to express a relation like (6.5) is by introducing the free left $\mathbb{O}$-module $M = \mathbb{O}^d$. The operations in this module are defined componentwise, e.g.,

$$R \cdot (\boldsymbol{P} + \boldsymbol{Q}) = R \cdot \begin{pmatrix} P_1 + Q_1 \\ \vdots \\ P_d + Q_d \end{pmatrix} = \begin{pmatrix} R \cdot (P_1 + Q_1) \\ \vdots \\ R \cdot (P_d + Q_d) \end{pmatrix} = \begin{pmatrix} RP_1 + RQ_1 \\ \vdots \\ RP_d + RQ_d \end{pmatrix}$$

for $R \in \mathbb{O}$ and $\boldsymbol{P}, \boldsymbol{Q} \in M$. Denoting with $\mathcal{F}$ the space of functions that we consider, we define the action of $\boldsymbol{P} \in M$ on $\boldsymbol{f} \in \mathcal{F}^d$ by

$$\boldsymbol{P} \bullet \boldsymbol{f} = \begin{pmatrix} P_1 \\ \vdots \\ P_d \end{pmatrix} \bullet \begin{pmatrix} f_1 \\ \vdots \\ f_d \end{pmatrix} := P_1 \bullet f_1 + \cdots + P_d \bullet f_d. \tag{6.6}$$

If the right-hand side of (6.6) evaluates to zero we say that the module element $\boldsymbol{P}$ annihilates the vector of functions $\boldsymbol{f}$. The set of all elements in M that annihilate $\boldsymbol{f}$ is a left submodule of M. So it makes sense to speak about *annihilating modules* rather than annihilating ideals. This construction becomes useful as soon as we are able to compute a basis for the submodule of relations between $f_1, \ldots, f_d$, i.e. the annihilator $\operatorname{Ann}_M(\boldsymbol{f})$ (as always we would be happy with a sufficiently large submodule, without claiming that it coincides with the full annihilator). Trivially we get that

$$\big\{(P_1, \ldots, P_d)^T \in M \mid P_i \bullet f_i = 0 \text{ for } 1 \le i \le d\big\} \subseteq \operatorname{Ann}_M(\boldsymbol{f}). \tag{6.7}$$

In many cases, when the f_i are not at all related with each other, we will have equality in (6.7). On the other side there are examples where more relations exist, in which case this construction makes sense in the first place.

Example 6.1. *Let $\mathbb{O} = \mathbb{Q}(x)[D_x; 1, D_x]$ and $M = \mathbb{O}^2$. First consider $\boldsymbol{f} = (f_1, f_2)^T$ with $f_1(x) = e^x$ and $f_2(x) = x$. Clearly whatever operator $P \in \mathbb{O}$ is applied to f_1, the result will be of the form $r(x)e^x$ for some rational function $r(x) \in \mathbb{Q}(x)$. So for any $(P_1, P_2)^T \in \operatorname{Ann}_M(\boldsymbol{f})$ we must have that $P_1 \bullet f_1 = 0$ because otherwise the surviving exponential cannot be canceled by $P_2 \bullet f_2$. Similarly $P_2 \bullet f_2 = 0$ must hold and we have equality in* (6.7)*.*

On the other hand let $\boldsymbol{g} = (g_1, g_2)^T$ with $g_1(x) = \log(x)$ and $g_2(x) = x$. Then clearly $(-D_x, x^{-2})^T \in \operatorname{Ann}_M(\boldsymbol{g})$ but does not have the property described in (6.7)*: both $(-D_x) \bullet g_1(x)$ and $x^{-2} \bullet g_2(x)$ are not zero.*

We have seen that in our work an essential ingredient is the ability to execute closure properties. Therefore we should have a short look on how to perform them for annihilating modules. It turns out that the algorithms of Section 2.3 can be extended in a natural way to the module setting. We want to exemplify that by discussing the closure property "multiplication by a scalar function", i.e., given an annihilating module $J \subseteq \operatorname{Ann}_M(\boldsymbol{f})$ for $\boldsymbol{f} \in \mathcal{F}^d$ and an annihilating ideal $I \subseteq \operatorname{Ann}_{\mathbb{O}}(g)$ for $g \in \mathcal{F}$, compute an annihilating module for $g \cdot \boldsymbol{f} = (gf_1, \ldots, gf_d)^T$ (this is also the only closure property that we will need in the following).

Once again, we use the basic ideas of the FGLM algorithm (see Figure 2.2). Since the result shall be an annihilating module in M, the "standard monomials" (that will be considered in the FGLM-systematic way) are elements of the form $\boldsymbol{P} = (0, \ldots, P_i, \ldots, 0)^T \in M$ where $P_i \in \mathbb{O}$ is a monomial. Consequently we do not start with $\{1\}$ as the set of test monomials, but with $\{(1, 0, \ldots, 0)^T, \ldots, (0, \ldots, 0, 1)^T\}$. The task is now to compute a normal form for such standard monomials. If we apply $\boldsymbol{P}$ to the product $g \cdot \boldsymbol{f}$ we obtain $P_i \bullet (gf_i)$, which can be rewritten to the form

$$\sum_k c_k \cdot (Q_k \bullet g) \cdot (R_k \bullet f_i), \quad Q_k, R_k \in \mathbb{O}$$

exactly as we did in the scalar case. Reducing the Q_k to normal form with respect to I and reducing the R_k to normal form with respect to J (for this purpose, R_k has to be identified with the vector that has R_k on its i-th position and that is zero elsewhere), yields a normal form for $\boldsymbol{P}$. The rest of the algorithm works exactly as depicted in Figure 2.2.

We turn back to our original problem: Compute an annihilating module for $\boldsymbol{f} = (f_1, f_2)^T$ where $f_1 = uvw$ and $f_2 = u'vw$ (u, v, w as defined in (6.2), and u' denoting the partial derivative of u with respect to x). Let from now on

$$\mathbb{O} = \mathbb{Q}(i, j, k, x, y, z)[S_i; S_i, 0][S_j; S_j, 0][S_k; S_k, 0][D_x; 1, D_x][D_y; 1, D_y][D_z; 1, D_z]$$

and $M = \mathbb{O}^2$. We expect the annihilating module for $\boldsymbol{f}$ to be strictly larger than (6.7), because f_1 and f_2 are closely related by the common factor vw. Hence we will start with $\boldsymbol{h} = (u, u')^T$ and compute an annihilating ideal for $\boldsymbol{f} = (vw) \cdot \boldsymbol{h}$ by the above mentioned closure property.

First we have to ask ourselves how to compute an annihilating module $J \subseteq \operatorname{Ann}_M \boldsymbol{h} \subseteq M$ for $(u, u')^T$. We know how to compute annihilating ideals for u and u', and clearly J should incorporate this knowledge (this gives rise to elements in J that have only a single nonzero entry annihilating the corresponding component). Additionally we are aware of the fact that the second component is the derivative of the first one. Putting all this together, we let J be the submodule that is generated by

$$\left\{(P, 0)^T \mid P \in \operatorname{Ann}_{\mathbb{O}} u\right\} \cup \left\{(0, P)^T \mid P \in \operatorname{Ann}_{\mathbb{O}} u'\right\} \cup \left\{(D_x, -1)^T\right\}.$$

At this point we should not forget to compute a module Gröbner basis of the above, because this will usually not be the case on its own. Now the closure property "component-wise multiplication by vw" can be performed, yielding an annihilating module for $\boldsymbol{f}$.

Finally we can use the ansatz technique as before in order to find an $\{x, y, z\}$-free element in the annihilating module of $\boldsymbol{f}$. As a result we get the following relation:

$$\begin{gathered}
-2(1+2i)(2+j)(3+2i+j)(7+2i+2j)(5+i+j+k)(7+i+j+k) \\
(8+i+j+k)(8+2i+2j+k)(9+2i+2j+k)(11+2i+2j+2k) \\
(15+2i+2j+2k)\, f_1(i, j+1, k+3) \\
\vdots \\
+\ \langle 31 \text{ similar terms} \rangle\ + \\
\vdots \\
-2(4+2i+j)(5+2i+j)(5+2i+2j)(5+i+j+k)(6+i+j+k) \\
(8+i+j+k)(10+2i+2j+k)(11+2i+2j+k)(11+2i+2j+2k) \\
(15+2i+2j+2k)\, f_2(i+1, j+2, k+3) = 0
\end{gathered}$$

(most other cases are smaller, i.e., when taking the partial derivatives of v and w). But why restrict to vectors of dimension 2? We could as well try to find relations for

$$\boldsymbol{f} = \left(uvw,\ \frac{\partial u}{\partial x} vw,\ \frac{\partial u}{\partial y} vw,\ \frac{\partial u}{\partial z} vw\right)^T$$

and obtain an identity that connects uvw, $\frac{\partial u}{\partial x} vw$, and $\frac{\partial u}{\partial z} vw$, and that has the following support:

$$\begin{aligned}
\{&(0,0,0,S_j^3S_k^4)^T,(0,0,0,S_j^2S_k^5)^T,(0,0,0,S_jS_k^6)^T,(0,S_j^2S_k^4,0,0)^T,\\
&(0,0,0,S_iS_j^2S_k^3)^T,(0,0,0,S_iS_jS_k^4)^T,(0,0,0,S_iS_k^5)^T,(0,0,0,S_j^3S_k^3)^T,\\
&(0,0,0,S_j^2S_k^4)^T,(0,0,0,S_jS_k^5)^T,(S_j^2S_k^3,0,0,0)^T,(0,S_j^2S_k^3,0,0)^T,\\
&(0,0,0,S_iS_j^2S_k^2)^T,(0,0,0,S_iS_jS_k^3)^T,(0,0,0,S_iS_k^4)^T,(0,0,0,S_j^3S_k^2)^T,\\
&(0,0,0,S_j^2S_k^3)^T,(0,0,0,S_jS_k^4)^T,(0,S_j^2S_k^2,0,0)^T,(0,0,0,S_iS_j^2S_k)^T,\\
&(0,0,0,S_iS_jS_k^2)^T,(0,0,0,S_iS_k^3)^T,(0,0,0,S_j^3S_k)^T,(0,0,0,S_j^2S_k^2)^T,\\
&(0,0,0,S_jS_k^3)^T,(0,0,0,S_j^3)^T,(0,0,0,S_j^2S_k)^T,(0,0,0,S_jS_k^2)^T\}.
\end{aligned}$$

Chapter 7

A fully algorithmic proof of Stembridge's TSPP theorem

Chapter 3 contained four algorithms, all of them serving more or less the same goal: to find a creative telescoping relation in a given annihilating ideal. To the three existing classical algorithms we added what we called the "polynomial ansatz". In this chapter an application is presented that justifies its invention. In fact, it was the TSPP problem that lead us to come up with this kind of ansatz.

The theorem (see Theorem 7.3 below) that we are going to prove is about the enumeration of *totally symmetric plane partitions* (TSPP); it was first proven by John Stembridge [81]. We will reprove the statement using only computer algebra; this means that basically no human ingenuity (from the mathematical point of view) is needed any more—once the algorithmic method has been invented (in this case, by Doron Zeilberger). But it is not as simple (otherwise our contribution would be trivial): The summation problems that have to be solved are very much involved and we were not able to do them with the known methods that have been presented in Sections 3.1, 3.2, and 3.3. One option would be to wait for 20 years hoping that Moore's law equips us with computers that are thousands of times faster than the ones of nowadays and that can do the job easily. But we prefer a second option, namely to think about how to make the problem feasible for today's computers.

Somehow, the results of this chapter are a byproduct of a joint work with Doron Zeilberger and Manuel Kauers where the long term goal is to apply the methodology described below to the Andrews-Robbins q-TSPP conjecture [80] (which is a q-analogue of Theorem 7.3). Some first but remarkable steps into this direction have been done [48], but at some point the limitations of our computers forced us to stop. Therefore the ordinary $q = 1$

case serves as a proof-of-concept and to get a feeling for the complexity of the underlying computations; it delivers valuable information that go further than just giving yet another proof of Stembridge's theorem.

It should be mentioned that this is not the first proof of Stembridge's theorem that uses computer algebra. In 2005 George Andrews, Peter Paule, and Carsten Schneider [10] came up with a computer-assisted proof. By means of a matrix decomposition into triangular matrices they transformed the problem of evaluating the determinant into the task to verify the correctness of the decomposition which resulted in a couple of hypergeometric multiple-sum identities. These were solved with help of the Mathematica package `Sigma` [78]. Finding the matrix decomposition however required human insight (and so far it is not clear how to do it in the q-case). We claim to have the first "human-free" computer proof of Stembridge's theorem that is completely algorithmic and does not require any human insight into the problem. Moreover our method generalizes immediately to the q-case and it is only the computational complexity that prevented us from proving it.

7.1 Totally symmetric plane partitions

We first want to motivate the topic of this chapter by giving a short explanation of the underlying combinatorial objects.

Definition 7.1. *A* plane partition π *of some integer* n *is a two-dimensional array*

$$\pi = (\pi_{i,j}), \quad \pi_{i,j} \in \mathbb{N} \textit{ for integers } i,j \geq 1, \quad \sum_{i,j\geq 1} \pi_{i,j} = n,$$

which is weakly decreasing in rows and columns, or more precisely

$$\pi_{i+1,j} \leq \pi_{i,j} \quad \textit{and} \quad \pi_{i,j+1} \leq \pi_{i,j} \quad \textit{for all } i,j \geq 1.$$

Note that this definition implies that only finitely many entries $\pi_{i,j}$ can be nonzero. To each plane partition we can draw its 3D Ferrers diagram by stacking $\pi_{i,j}$ unit cubes on top of the location (i,j). Each unit cube can be addressed by its location (i,j,k) in 3D coordinates. The conditions on the entries of $\pi_{i,j}$ imply that the 3D Ferrers diagram is an origin-justified structure, in the sense that if the position (i,j,k) is occupied then so are all positions (i',j',k') with $i' \leq i$, $j' \leq j$, and $k' \leq k$. Figure 7.1 shows an example of a plane partition together with its 3D Ferrers diagram. We are now going to define TSPPs, the objects of interest.

5	4	1
3	2	1
1		

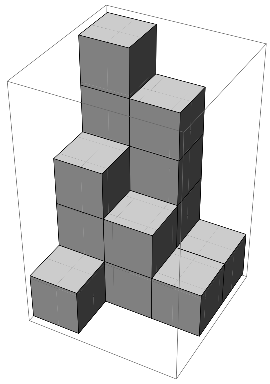

Figure 7.1: A plane partition of $n = 17$

Definition 7.2. *A plane partition is* totally symmetric *iff whenever the position (i, j, k) in its 3D Ferrers diagram is occupied (in other words $\pi_{i,j} \geq k$), it follows that all its permutations $\{(i, k, j), (j, i, k), (j, k, i), (k, i, j), (k, j, i)\}$ are also occupied.*

Stembridge's theorem [81] now tells us how many TSPPs of a certain size exist.

Theorem 7.3. *The number of totally symmetric plane partitions whose 3D Ferrers diagram is contained in the cube $[0, n]^3$ is given by the product-formula*

$$\prod_{1 \leq i \leq j \leq k \leq n} \frac{i + j + k - 1}{i + j + k - 2}. \tag{7.1}$$

Example 7.4. *We are considering the case $n = 2$: Formula (7.1) tells us that there should be*

$$\prod_{1 \leq i \leq j \leq k \leq 2} \frac{i + j + k - 1}{i + j + k - 2} = \frac{2}{1} \cdot \frac{3}{2} \cdot \frac{4}{3} \cdot \frac{5}{4} = 5$$

TSPPs that fit into the cube $[0, 2]^3$. This result is confirmed by the enumeration given in Figure 7.2.

As others that proved the TSPP formula before us we will make use of a result by Soichi Okada [61] that reduces the proof of Theorem 7.3 to a determinant evaluation:

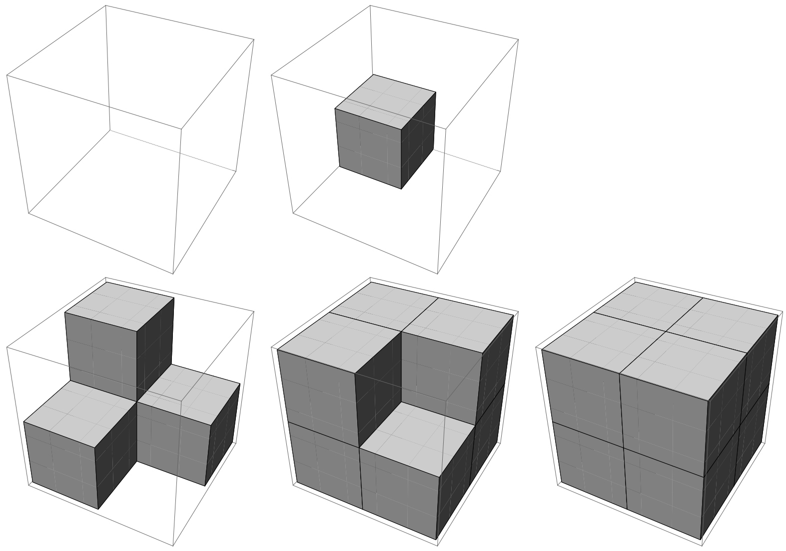

Figure 7.2: All TSPPs that fit into the cube $[0, 2]^2$

Theorem 7.5. *The enumeration formula* (7.1) *for TSPPs is correct if and only if the determinant evaluation*

$$\det\left(a(i,j)\right)_{1\le i,j\le n} = \prod_{1\le i\le j\le k\le n} \left(\frac{i+j+k-1}{i+j+k-2}\right)^2 \tag{7.2}$$

holds, where the entries in the matrix are given by

$$a(i,j) = \binom{i+j-2}{i-1} + \binom{i+j-1}{i} + 2\delta(i,j) - \delta(i,j+1). \tag{7.3}$$

In the above, $\delta(i,j)$ denotes the Kronecker delta.

7.2 How to prove determinant evaluations

Doron Zeilberger [95] proposes a method for completely automatic and rigorous proofs of determinant evaluations that fit into a certain class. For the sake of self-containedness this section gives a short summary how this method works. It concerns the following problem: For all $n \geq 1$ prove that

$$\det(a(i,j))_{1\le i,j\le n} = \mathrm{Nice}(n),$$

for some explicitly given expressions $a(i,j)$ and $\mathrm{Nice}(n)$. We assume that the matrix has full rank. What you have to do is the following: Pull out of the hat another discrete function $B(n,j)$ (this looks a little bit like magic for now—we will make this step more explicit later) and check the identities

$$\sum_{j=1}^{n} B(n,j)a(i,j)=0 \quad \text{for } 1 \le i < n, \quad i,n \in \mathbb{N}, \tag{7.4}$$

$$B(n,n)=1 \quad \text{for all } n \ge 1, \quad n \in \mathbb{N}. \tag{7.5}$$

If they hold, then by uniqueness it follows that $B(n,j)$ equals the cofactor of the (n,j)-entry of the $n \times n$ determinant divided by the $(n-1) \times (n-1)$ determinant. This fact becomes more obvious when we expand the determinant with respect to the last row using Laplace's formula (we divided the whole equation by $M_{n,n}$ already):

$$\frac{1}{M_{n,n}} \det(a(i,j))_{1\le i,j\le n} = \sum_{j=1}^{n} \underbrace{\frac{(-1)^{n+j} M_{n,j}}{M_{n,n}}}_{B(n,j)} a(n,j)$$

where $M_{n,j}$ denotes the (n,j)-minor, i.e., the determinant of the matrix with the last row and the jth column removed. Changing the term $a(n,j)$ in the sum to $a(i,j), i < n$ corresponds to replace the last row of the original matrix by its i-th row. The determinant then is of course 0 and we have equation (7.4). Since there are $n-1$ choices for i we obtain as many linear independent equations for the $B(n,j), 1 \le j \le n$ and they determine the solution up to a constant multiple. This ambiguity is disposed by the normalization condition (7.5).

If one finally succeeds to verify the identity

$$\sum_{j=1}^{n} B(n,j)a(n,j) = \frac{\text{Nice}(n)}{\text{Nice}(n-1)} \quad \text{for all } n \ge 1, \quad n \in \mathbb{N}, \tag{7.6}$$

the determinant evaluation Nice(n) is established.

7.3 The computer proof

The annihilating ideal

The first thing we have to do according to Zeilberger's algorithmic proof technique is to resolve the magic step that we have left as a black box so far, namely "to pull out of the hat" the sequence $B(n,j)$ for which we have to verify the identities (7.4) – (7.6). Note that we are able, using the definition of what $B(n, j)$ is supposed to be (namely a certain cofactor in a determinant expansion), to compute the values of $B(n, j)$ for small concrete integers n and j. This data allows us (by plugging it into an appropriate ansatz and solving the resulting linear system) to find recurrence relations for $B(n, j)$ that will hold for all values of n and j with a very high probability. We call this method *guessing*; it has been executed by Manuel Kauers who used his highly optimized software `Guess` [46]. More details about this part of the proof can be found in [48]. The result of the guessing were 65 recurrences, their total size being about 5MB.

Many of these recurrences are redundant and it is desirable to have a unique description of the object in question that additionally is as small as possible (in a certain metric). To this end we compute a Gröbner basis of the ∂-finite left ideal that is generated by the 65 recurrences. The computation was executed by the noncommutative Gröbner basis implementation which is part of the package `HolonomicFunctions`. We later found out that it can as well be obtained directly from the guessing procedure. One just has to choose the structure set of the ansatz in an appropriate way. The Gröbner basis consists of 5 polynomials (their total size being about 1.6MB). Their leading monomials $S_j^4, S_j^3 S_n, S_j^2 S_n^2, S_j S_n^3, S_n^4$ form a staircase of regular shape. This means that we should take 10 initial values into account which correspond to the monomials under the staircase.

In addition, we have now verified that all the 65 recurrences are consistent. Hence they are all valid for the same object. If this is not the case then we would have expected to get $\{1\}$ as the Gröbner basis. The next question is whether the 5 elements of the Gröbner basis together with the 10 initial values define a unique bivariate sequence, or in other words whether they give already a complete description of the sequence that we want to identify as $B(n, j)$.

The singularities

Before we start to prove the relevant identities there is one subtle point that, aiming at a fully rigorous proof, we should not omit: the question of singularities in the ∂-finite description of $B'(n, j)$. Recall that in the univariate case when we deal with a P-finite recurrence, we have to regard the zeros of the leading coefficient and in case that they introduce singularities in the range where we would like to apply the recurrence, we have to separately specify the values of the sequence at these points. Similarly in the bivariate case: We have to check whether there are points in $\mathbb{N}^2$ where none of the recurrences can be applied because the leading term vanishes.

For all points that lie in the area $(4, 4) + \mathbb{N}^2$ we may apply any of the recurrences, hence we have to look for common nonnegative integer solutions of all their leading coefficients (being bivariate polynomials in n and j). In order to determine the common solutions we compute a (commutative) Gröbner basis of the leading coefficients (the variables shifted according to the exponents of the leading monomial). It reveals that everything goes well in the $(4,4)$-shifted quadrant:

$$\begin{aligned}\big\{&(n-3)^2(n-2)(n-1)^2(2n-3)^2(2n-1)(j+n-1)(j+n),\\ &(n-2)(n-1)(2n-3)(2n-1)(j+n-1)(j+n)(1608j+154n^3-847n^2\\ &\quad -222n-693),\\ &(j-3)(n-1)(2n-1)(j-n)(j+n-1)(j+n),\\ &(2n-1)(j+n-1)(j+n)(308n^6-3080n^5+12397n^4+9648jn^2+\dots),\\ &(j+n-1)(j+n)(1053818880n^{14}-983564288jn^{13}+\dots)\big\}\end{aligned}$$

where two irreducible polynomials were cut off for better readability. First one observes that all Gröbner basis elements contain the factors $(j+n)$ and $(j+n-1)$ from which we can read off the solutions $(0,0)$, $(1,0)$, and $(0,1)$. But since these points are lying under the stairs anyway they are of no interest. The remaining factors of the first polynomial involve only the variable n and they tell us that we have to address the cases $n=1,2,3$. The first three polynomials contain the factor $(n-1)$ and hence are zero for all j when $n=1$. Plugging $n=1$ into the remaining two polynomials and factoring delivers

$$\begin{aligned}&3216(j-2)^2(j-1)j(j+1),\\ &1072(j-2)^2(j-1)j(j+1)\,(65536j^{11}-2686976j^{10}+\dots)\end{aligned}$$

which luckily restricts j to a finite set of values: For $n=1$ we obtain the additional solutions $j=1$ and $j=2$. Doing the same reasoning for $n=2$ and $n=3$ we can write down the complete set of common nonnegative integer solutions:

$$\{(0,0),(0,1),(1,0),(1,1),(2,1),(2,2),(3,2),(3,3)\}.$$

But all of them are outside of $(4,4)+\mathbb{N}^2$ so we need not to care.

It remains to look at the lines $j=0,1,2,3$ and the lines $n=0,1,2,3$. They are simpler to treat because we can substitute the value of the line for j resp. n and are left with finding common nonnegative integer roots of univariate polynomials. As an example consider the line $j=0$: On this line we may only apply the last recurrence that has leading monomial S_n^4, since all others would require values outside the first quadrant. So take the leading coefficient of this recurrence, shift n to $n-4$, and substitute $j=0$, obtaining

$$576(n-4)(n-3)^2(n-2)^2(n-1)^3n^2 \\ \times(2n-7)(2n-5)(2n-3)(2n-1)(3n-8)(3n-4) \\ \times(81n^6-1389n^5+10009n^4-38819n^3+85522n^2-101620n+51016)\,.$$

We can read off the singularities $(0,0)$, $(0,1)$, $(0,2)$, $(0,3)$, and $(0,4)$. In a similar way the other seven cases can be treated (we omit the details here). Summarizing, the points for which initial values have to be given (either because they are under the stairs or because of singularities) are

$$\{(j,n)\mid 0\le j\le 6\wedge 0\le n\le 1\}\cup\{(j,2)\mid 0\le j\le 4\}\cup \\ \{(j,3)\mid 0\le j\le 3\}\cup\{(j,4)\mid 0\le j\le 2\}\cup\{(1,5)\}.$$

They are depicted in Figure 7.3.

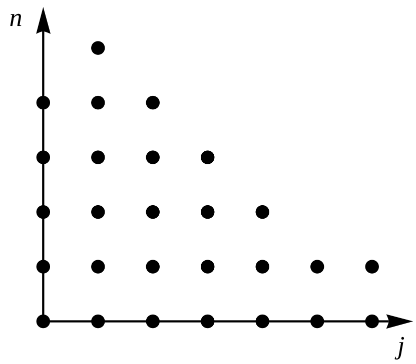

Figure 7.3: The points for which the initial values of the sequence $B(n,j)$ have to be given because the recurrences do not apply.

Aiming at a rigorous proof we have to admit at this point that what we have found so far does not prove anything yet. Supplying the ∂-finite description with these 27 initial values uniquely and completely defines a bivariate sequence—let's call it $B'(n,j)$. We still have to show that this $B'(n,j)$ is identical to the sequence $B(n,j)$ defined by (7.4) and (7.5). We anticipate the correctness of these two identities and write always $B(n,j)$ instead of $B'(n,j)$. Finally we have to verify that identity (7.6) indeed holds. In the following we prove the three identities in the order of their difficulty.

The second identity

The simplest of the three identities to prove is (7.5). From the ∂-finite description of $B'(n,j)$ we can compute a recurrence for the diagonal $B'(n,n)$ by the closure property substitution. Our command `DFiniteSubstitute` delivers a recurrence of order 7 in a couple of minutes. Reducing this recurrence with the ideal generated by $S_n - 1$ (which annihilates 1) gives 0; hence it is a left multiple of the recurrence for the right-hand side. We should not forget to have a look on the leading coefficient in order to make sure that we don't run into singularities:

$$256(2n+3)(2n+5)(2n+7)(2n+9)(2n+11)^2(2n+13)^2 p_1 p_2$$

where p_1 and p_2 are irreducible polynomials in n of degree 4 and 12 respectively. Comparing initial values (which of course match due to our definition) establishes identity (7.5).

The third identity

In order to prove (7.6) we first rewrite it slightly. Using the definition of the matrix entries $a(n,j)$ we obtain for the left-hand side

$$\sum_{j=1}^{n} B(n,j) \underbrace{\left(\binom{n+j-2}{n-1} + \binom{n+j-1}{n}\right)}_{=:a'(n,j)} + 2B(n,n) - B(n,n-1)$$

and the right-hand side simplifies to

$$\frac{\mathrm{Nice}(n)}{\mathrm{Nice}(n-1)} = \frac{\prod_{1\le i\le j\le k\le n}\left(\frac{i+j+k-1}{i+j+k-2}\right)^2}{\prod_{1\le i\le j\le k\le n-1}\left(\frac{i+j+k-1}{i+j+k-2}\right)^2} = \frac{4^{1-n}(3n-1)^2(2n)_{n-1}^2}{(3n-2)^2(n/2)_{n-1}^2}.$$

Note that $a'(n,j) = \frac{2n+j-1}{n+j-1}\binom{n+j-1}{j-1}$ is a hypergeometric expression in both variables j and n. A ∂-finite description of the summand can be computed with our implementation `DFiniteTimes` from the annihilating ideal of $B(n,j)$ using the closure property product. The task is now to find a creative telescoping relation for the summand. We tried to accomplish this in various ways.

Some unsuccessful tries

Zeilberger's slow algorithm We tried this approach but it seems to be hopeless: The summation variable j that we would like to eliminate occurs in the annihilating relations for the summand $B(n,j)a'(n,j)$ with degrees between 24 and 30. When we follow the intermediate results of the Gröbner basis computation we observe that none of the elements that were added to the basis because some S-polynomial did not reduce to zero has a degree in j lower than 23 (we aborted the computation after more than 48 hours). Additionally the coefficients grow rapidly and it seems very likely that we run out of memory before coming to an end.

Takayama's algorithm First we note that our summand has finite support and hence natural boundaries which allows us to apply Takayama's algorithm. Although this algorithm is much superior to the elimination algorithm proposed by Zeilberger, we were not able to complete the necessary computations. The underlying elimination problem, as before, seems to be unsolvable with today's computers: We now can lower the degree of j to 18, but the intermediate results consume already about 12GB of memory (after 48 hours).

Chyzak's algorithm In the run of Chyzak's algorithm (see Section 3.3) we have to solve a system of linear first-order difference equations. Due to the size of the input, we did not succeed in uncoupling this system (neither with our implementation nor with Stefan Gerhold's `OreSys`), and even if we can do this step, it would remain to solve a presumably huge (concerning the size of the coefficients as well as the order) scalar difference equation.

A successful approach

At this point we came up with the idea to treat this problem with the technique described in Section 3.4. The advantage of this technique is that the computations are better controllable for big inputs, because the building steps are only normal form computation with respect to a given Gröbner basis and linear algebra. No run of Buchberger's algorithm or expensive uncoupling is needed.

We found by means of modular computations that the ansatz

$$\underbrace{\sum_{i=0}^{I} p_i(n) S_n^i}_{=P(n,S_n)} \quad + \quad (S_j - 1) \cdot \underbrace{\sum_{k=0}^{K} \sum_{l=0}^{L} \sum_{m=0}^{M} q_{k,l,m}(n) j^k S_j^l S_n^m}_{=Q(j,n,S_j,S_n)} . \tag{7.7}$$

with $I = 7$, $K = 5$, and the support of Q being the power products $S_j^l S_n^m$ with $l + m \leq 7$ delivers a solution with nontrivial principal part. After omitting the 0-components of this solution, we ended up with an ansatz containing 126 unknowns. For computing the final solution we used again homomorphic images and rational reconstruction. Still it was quite some effort to compute the solution (it consists of rational functions in n with degrees up to 382 in the numerators and denominators). The total size of the telescoping relation becomes smaller when we reduce the delta part to normal form (then obtaining an operator of the form that Chyzak's algorithm would deliver). In contrast to Chyzak's algorithm we do not necessarily find the telescoping relation with minimal-order principal part. Finally the result takes about 5 MB of memory. We counterchecked its correctness by reducing the relation with the annihilating ideal of $B(n, j)a'(n, j)$ and obtained 0 as expected.

We have now a recurrence for the sum but we need to to cover the whole left-hand side. A recurrence for $B(n, n-1)$ is easily obtained with our package performing the substitution $j \to n-1$, and $B(n, n) = 1$ as shown before. Using our command `DFinitePlus`, the closure property sum delivers a recurrence of order 10. On the right-hand side we have a ∂-finite expression for which our package automatically computes an annihilating operator. This operator is a right divisor of the one that annihilates the left-hand side. By comparing 10 initial values and verifying that the leading coefficients of the recurrences do not have singularities among the positive integers, we have established identity (7.6).

The first identity

With the same notation as before we reformulate identity (7.4) as

$$\sum_{j=1}^{n} B(n, j)a'(i, j) = B(n, i-1) - 2B(n, i).$$

The hard part again is to do the sum on the left-hand side. Since two parameters i and n are involved and remain after the summation, one annihilating operator does not suffice. We decided to search for two operators with leading monomials being pure powers of S_i and S_n respectively. Although this is far away from being a Gröbner basis, it is nevertheless a complete description of the object (together with sufficiently (but still finitely) many initial values). We obtained these two relations in a similar way as in the previous section, but the computational effort was even bigger (more than 500 hours of computation time were needed). The first telescoping operator is about 200 MB big and the support of its principal part is

$$\{S_i^5, S_i^4 S_n, S_i^3 S_n^2, S_i^2 S_n^3, S_i S_n^4, S_i^4, S_i^3 S_n, S_i^2 S_n^2, S_i S_n^3,\\ S_i^3, S_i^2 S_n, S_i S_n^2, S_n^3, S_i^2, S_i S_n, S_n^2, S_i, S_n, 1\}.$$

The second one is of size 700 MB and the support of its principal part is

$$\{S_n^5, S_i^4, S_i^3 S_n, S_i^2 S_n^2, S_i S_n^3, S_n^4, S_i^3, S_i^2 S_n, S_i S_n^2, S_n^3, S_i^2, S_i S_n, S_n^2, S_i, S_n, 1\}.$$

Again we can independently from their derivation check their correctness by reducing them with the annihilating ideal of $B(n,j)a'(i,j)$: both give 0.

Let's now examine the right-hand side: From the Gröbner basis of recurrences for $B(n,j)$ that we computed in Section 7.3 one immediately gets an annihilating ideal for $B(n,i-1)$ by replacing S_j by S_i and by substituting $j \to i-1$ in the coefficients. We now could apply the closure property sum but we can do better: Since the right-hand side can be written as $(1-2S_i) \bullet B(n,i-1)$ we can use the closure property operator application and obtain a Gröbner basis which has even less monomials under the stairs than the input, namely 8. The opposite we expect to happen when using the closure property sum: usually there the dimension grows but never can shrink. It is now a relatively simple task to verify that the two principal parts that were computed for the left-hand side are contained in the annihilating ideal of the right-hand side (both reductions give 0).

The initial value question needs some special attention here since we want the identity to hold only for $i < n$; hence we cannot simply look at the initial values in the square $[0,4]^2$. Instead we compare the initial values in a trapezoid-shaped area which allows us to compute all values below the diagonal. Since all these initial values match for the left-hand and right-hand side we have the proof that the identity holds for all $i < n$. Looking at the leading coefficients of the two principal parts we find that they contain the factors $5+i-n$ and $5-i+n$ respectively. This means that both operators cannot be used to compute values on the diagonal which is a strong indication that the identity does not hold there: Indeed, identity (7.4) is wrong for $n = i$ because in this case we get (7.6).

The prize

It might have become manifest that our proof did not go so smooth and we had to overcome many difficulties, in particular we had to recover from many disappointing unsuccessful attempts. Doron Zeilberger was so generous to stimulate my motivation by offering two prizes of \$100 and \$200 for proving (7.6) and (7.4), respectively! And he really pushed towards the complete rigorous proof as the following e-mail [96] evidences:

> I was about to write you a \$100 check, when I realized that you don't deserve it (yet) The stipulation of the prize was that you FIRST do (Soichi) for \$200, and then if you can also do (Okada) then you get an additional \$100.
>
> [...]

> P.S. This is like Jacob and Laban, before Jacob could marry Rachel, working for seven years, he had to marry Leah, also by working seven years, so the price of Rachel was 14 years of labor, plus an extra, unwanted wife. I am sure that it would take you less than 14 years.

7.4 Outlook

As we have demonstrated, Zeilberger's methodology is completely algorithmic and does not need human intervention. This fact makes it possible to apply it to other problems (of the same class) without further thinking. Just feed the data into the computer! The q-TSPP enumeration formula

$$\prod_{1\le i\le j\le k\le n} \frac{1-q^{i+j+k-1}}{1-q^{i+j+k-2}}$$

has been conjectured independently by George Andrews and Dave Robbins in the early 1980s [80]. This conjecture is still open and one of the most intriguing problems in enumerative combinatorics. The tantalizing aspect of this chapter is that the methodology can be applied one-to-one to that problem (also a q-analogue of Okada's result exists). Unfortunately, due to the additional indeterminate q the complexity of the computations is increased considerably which prevents us from proving it right away. But we are working on that...

Bibliography

[1] Sergej Abramov. Rational solutions of linear differential and difference equations with polynomial coefficients. *USSR Computational Mathematics and Mathematical Physics*, 29(6):7–12, 1989.

[2] Sergej Abramov. Rational solutions of linear difference and q-difference equations with polynomial coefficients. In *ISSAC'95: Proceedings of the 1995 International Symposium on Symbolic and Algebraic Computation*, pages 285–289, New York, NY, USA, 1995. ACM.

[3] Sergej Abramov and Moulay Barkatou. Rational solutions of first order linear difference systems. In *ISSAC'98: Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation*, pages 124–131, New York, NY, USA, 1998. ACM.

[4] Sergej Abramov, Manuel Bronstein, and Marko Petkovšek. On polynomial solutions of linear operator equations. In *ISSAC '95: Proceedings of the 1995 International Symposium on Symbolic and Algebraic Computation*, pages 290–296, New York, NY, USA, 1995. ACM.

[5] Sergej Abramov and Eugene Zima. A universal program to uncouple linear systems. In *Proceedings of the International Conference on Computational Modelling and Computing in Physics, Dubna, Russia*, pages 16–26, September 1996.

[6] Milton Abramowitz and Irene Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* Dover, New York, 1964.

[7] Gert Almkvist and Doron Zeilberger. The method of differentiating under the integral sign. *Journal of Symbolic Computation*, 10(6):571–591, 1990.

[8] Tewodros Amdeberhan and Victor H. Moll. A formula for a quartic integral: a survey of old proofs and some new ones. *Ramanujan Journal*, 18(1):91–102, 2009.

[9] George Andrews, Richard Askey, and Ranjan Roy. *Special functions*, volume 71 of *Encyclopedia of mathematics and its applications*. Cambridge University Press, Cambridge, UK, 1999.

[10] George E. Andrews, Peter Paule, and Carsten Schneider. Plane Partitions VI. Stembridge's TSPP theorem. *Advances in Applied Mathematics*, 34:709–739, 2005.

[11] Moulay Barkatou. On rational solutions of systems of linear differential equations. *Journal of Symbolic Computation*, 28:547–567, 1999.

[12] Moulay Barkatou. On super-irreducible forms of linear differential systems with rational function coefficients. *Journal of Computational and Applied Mathematics*, 162(1):1–15, 2004.

[13] Moulay Barkatou and Eckhard Pflügel. Computing super-irreducible forms of systems of linear differential equations via Moser-reduction: a new approach. In *ISSAC'07: Proceedings of the 2007 International Symposium on Symbolic and Algebraic Computation*, pages 1–8, New York, NY, USA, 2007. ACM.

[14] Thomas Becker, Heinz Kredel, and Volker Weispfenning. *Gröbner bases: a computational approach to commutative algebra*. Springer-Verlag, London, UK, 1993.

[15] George M. Bergman. The diamond lemma for ring theory. *Advances in Mathematics*, 29:178–218, 1978.

[16] Joseph N. Bernstein. The analytic continuation of generalized functions with respect to a parameter. *Functional Analysis and its Applications*, 6(4):273–285, 1972.

[17] Sven Beuchler and Veronika Pillwein. Sparse shape functions for tetrahedral p-FEM using integrated Jacobi polynomials. *Computing*, 80(4):345–375, 2007.

[18] Sven Beuchler and Joachim Schöberl. New shape functions for triangular p-FEM using integrated Jacobi polynomials. *Numerische Mathematik*, 103(3):339–366, 2006.

[19] George Boros and Victor Moll. *Irresistible Integrals. Symbolics, Analysis and Experiments in the Evaluation of Integrals*. Cambridge University Press, 2004.

[20] Alin Bostan and Manuel Kauers. The full counting function for Gessel walks is algebraic. Technical report, INRIA-Rocquencourt, 2008. In preparation.

[21] Manuel Bronstein and Marko Petkovšek. An introduction to pseudo-linear algebra. *Theoretical Computer Science*, 157(1):3–33, 1996.

[22] Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal*. PhD thesis, University of Innsbruck, Innsbruck, Austria, 1965.

[23] Pierre Cartier. Démonstration "automatique" d'identités et fonctions hypergéometriques [d'après D. Zeilberger]. *Astérisque*, 206:41–91, 1991. Séminaire Bourbaki, 44ème année, 1991–1992, n°746.

[24] Frédéric Chyzak. *Fonctions holonomes en calcul formel*. PhD thesis, École polytechnique, 1998.

[25] Frédéric Chyzak. Gröbner bases, symbolic summation and symbolic integration. In Bruno Buchberger and Franz Winkler, editors, *Gröbner Bases and Applications*, volume 251 of *London Mathematical Society Lecture Notes Series*, pages 32–60. Cambridge University Press, 1998. Proceedings of the Conference 33 Years of Gröbner Bases.

[26] Frédéric Chyzak. An extension of Zeilberger's fast algorithm to general holonomic functions. *Discrete Mathematics*, 217(1-3):115–134, 2000.

[27] Frédéric Chyzak, Manuel Kauers, and Bruno Salvy. A non-holonomic systems approach to special function identities. In *ISSAC'09: Proceedings of the 2009 International Symposium on Symbolic and Algebraic Computation*, pages 111–118, New York, NY, USA, 2009. ACM.

[28] Frédéric Chyzak and Peter Paule. Computer algebra. In D. Lozier, editor, *The Digital Library of Mathematical Functions (DLMF)*. National Institute of Standards and Technology (NIST), Gaithersburg, U.S.A., 2007. In preparation.

[29] Frédéric Chyzak and Bruno Salvy. Non-commutative elimination in Ore algebras proves multivariate identities. *Journal of Symbolic Computation*, 26:187–227, 1998.

[30] Severino C. Coutinho. *A primer of algebraic D-modules*, volume 33 of *London Mathematical Society Student Texts*. Cambridge University Press, 1995.

[31] David Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms*. Springer-Verlag, New York, 1992.

[32] Sister Mary Celine Fasenmyer. *Some generalized hypergeometric polynomials*. PhD thesis, University of Michigan, 1945.

[33] Jean Charles Faugère, Patrizia Gianni, Daniel Lazard, and Teo Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *Journal of Symbolic Computation*, 16(4):329–344, 1993.

[34] Richard P. Feynman and Ralph Leighton (ed.). *Surely You're Joking, Mr. Feynman!: Adventures of a Curious Character*. W. W. Norton & Co, 1985.

[35] Philippe Flajolet, Stefan Gerhold, and Bruno Salvy. On the non-holonomic character of logarithms, powers and the nth prime function. *Electronic Journal of Combinatorics*, 11(2):1–16, 2005.

[36] Carl Friedrich Gauß. Disquisitiones generales circa series infinitam $1 + \frac{\alpha\beta}{1\cdot\gamma}x + \frac{\alpha(\alpha+1)\beta(\beta+1)}{1\cdot2\cdot\gamma(\gamma+1)}xx + \frac{\alpha(\alpha+1)(\alpha+2)\beta(\beta+1)(\beta+2)}{1\cdot2\cdot3\cdot\gamma(\gamma+1)(\gamma+2)}x^3 +$ etc. In *Commentationes societatis regiae scientiarum Gottingensis recentiores*, volume II, pages 3–46, 1813. English and German translation by Christoph Koutschan.

[37] Stefan Gerhold. Uncoupling systems of linear Ore operator equations. Master's thesis, RISC, Johannes Kepler University Linz, 2002. http://www.risc.uni-linz.ac.at/research/combinat/software/OreSys/.

[38] Stefan Gerhold. *Combinatorial Sequences: Non-Holonomicity and Inequalities*. PhD thesis, RISC, Johannes Kepler University Linz, 2005.

[39] Stefan Gerhold, Manuel Kauers, Frank W. J. Olver, Peter Paule, Carsten Schneider, and Burkhard Zimmermann. Computer proofs of some identities for Bessel functions of fractional order. In preparation, 2005.

[40] Alessandro Giovini, Teo Mora, Gianfranco Niesi, Lorenzo Robbiano, and Carlo Traverso. "one sugar cube, please" or selection strategies in the Buchberger algorithm. In *ISSAC'91: Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation*, pages 49–54, New York, NY, USA, 1991. ACM.

[41] William Gosper. Decision procedure for indefinite hypergeometric summation. *Proceedings of the US National Academy of Sciences*, 75:40–42, 1978.

[42] Israil S. Gradshteyn and Josif M. Ryzhik. *Table of Integrals, Series, and Products*. Academic Press, Elsevier, seventh edition, 2007. Alan Jeffrey and Daniel Zwillinger (eds.).

[43] Gert-Martin Greuel, Gerhard Pfister, and Hans Schönemann. Singular 3.1. A computer algebra system for polynomial computations, 2009. Centre for Computer Algebra, University of Kaiserslautern, http://www.singular.uni-kl.de.

[44] Abdelilah Kandri-Rody and Volker Weispfenning. Non-commutative Gröbner bases in algebras of solvable type. *Journal of Symbolic Computation*, 9(1):1–26, 1990.

[45] Masaki Kashiwara. On the holonomic systems of linear differential equations, II. *Inventiones Mathematicae*, 49(2):121–135, 1978.

[46] Manuel Kauers. Guessing handbook. Technical Report 09-07, RISC Report Series, Johannes Kepler University Linz, 2009. http://www.risc.uni-linz.ac.at/research/combinat/software/Guess/.

[47] Manuel Kauers and Christoph Koutschan. A Mathematica package for q-holonomic sequences and power series. *The Ramanujan Journal*, 19(2):137–150, 2009. http://www.risc.uni-linz.ac.at/research/combinat/software/qGeneratingFunctions/.

[48] Manuel Kauers, Christoph Koutschan, and Doron Zeilberger. A proof of George Andrews' and Dave Robbins' q-TSPP conjecture (modulo a finite amount of routine calculations). *The personal journal of Shalosh B. Ekhad and Doron Zeilberger*, pages 1–8, 2009. http://www.math.rutgers.edu/~zeilberg/pj.html.

[49] Manuel Kauers, Christoph Koutschan, and Doron Zeilberger. Proof of Ira Gessel's lattice path conjecture. *Proceedings of the US National Academy of Sciences*, 106(28):11502–11505, 2009.

[50] Manuel Kauers and Viktor Levandovskyy. An interface between Singular and Mathematica. http://www.kauers.de/software.html, 2006.

[51] Manuel Kauers and Doron Zeilberger. The quasi-holonomic ansatz and restricted lattice walks. *Journal of Difference Equations and Applications*, 14(10–11):1119–1126, 2008.

[52] Sebastian Klein. Heavy flavor coefficient functions in deep-inelastic scattering at $O(a_s^2)$ and large virtualities. Diplomarbeit, Universität Potsdam, Germany, 2006.

[53] Wolfram Koepf. *Hypergeometric Summation. An Algorithmic Approach to Summation and Special Function Identities.* Advanced Lectures in Mathematics. Vieweg Verlag, Braunschweig/Wiesbaden, 1998.

[54] Christoph Koutschan. HolonomicFunctions (user's guide). Technical Report 10-01, RISC Report Series, Johannes Kepler University, Linz, Austria, 2010. http://www.risc.jku.at/research/combinat/software/HolonomicFunctions/.

[55] Viktor Levandovskyy. *Non-commutative computer algebra for polynomial algebras: Gröbner bases, applications and implementation.* PhD thesis, Universität Kaiserslautern, 2005.

[56] Viktor Levandovskyy and Jorge Martin Morales. Computational D-module theory with SINGULAR, comparison with other systems and two new algorithms. In *ISSAC'08: Proceedings of the 2008 International Symposium on Symbolic and Algebraic Computation*, pages 173–180, New York, NY, USA, 2008. ACM.

[57] Viktor Levandovskyy and Hans Schönemann. Plural: a computer algebra system for noncommutative polynomial algebras. In *ISSAC'03: Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*, pages 176–183, New York, NY, USA, 2003. ACM.

[58] Leonard Lipshitz. D-finite power series. *Journal of Algebra*, 122:353–373, 1989.

[59] Christian Mallinger. Algorithmic Manipulations and Transformations of Univariate Holonomic Functions and Sequences. Master's thesis, RISC, Johannes Kepler University, August 1996. http://www.risc.uni-linz.ac.at/research/combinat/software/GeneratingFunctions/.

[60] Toshinori Oaku. Algorithms for the b-function and D-modules associated with a polynomial. *Journal of Pure and Applied Algebra*, 117-118:495–518, 1997.

[61] Soichi Okada. On the generating functions for certain classes of plane partitions. *Journal of Combinatorial Theory, Series A*, 53:1–23, 1989.

[62] Frank W. J. Olver. DLMF chapter BS. Personal e-mail communication with Peter Paule, 2006.

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

[64] Peter Paule. On identities of the Rogers-Ramanujan type. *Journal of Mathematical Analysis and Applications*, 107:255–284, 1985.

[65] Peter Paule. Parallel algorithms in computer algebra—Zeilberger's holonomic systems approach. Technical Report 90-49.0, RISC Report Series, Johannes Kepler University Linz, 1990.

[66] Peter Paule. Greatest factorial factorization and symbolic summation. *Journal of Symbolic Computation*, 20(3):235–268, 1995.

[67] Peter Paule and Veronika Pillwein. Automatic improvements of Wallis' inequality. Technical Report 08–18, RISC Report Series, University of Linz, Austria, 2008.

[68] Peter Paule and Axel Riese. A Mathematica q-analogue of Zeilberger's algorithm based on an algebraically motivated approach to q-hypergeometric telescoping. In Mourad E. H. Ismail, David R. Masson, and Mizan Rahman, editors, *Special Functions, q-Series and Related Topics*, volume 14 of *Fields Institute Communications*, pages 179–210. American Mathematical Society, 1997.

[69] Peter Paule and Markus Schorn. A Mathematica version of Zeilberger's algorithm for proving binomial coefficient identities. *Journal of Symbolic Computation*, 20(5/6):673–698, 1995. http://www.risc.uni-linz.ac.at/research/combinat/software/PauleSchorn/.

[70] Peter Paule and Volker Strehl. Symbolic summation—some recent developments. In *Computer Algebra in Science and Engineering—Algorithms, Systems, and Applications*, pages 138–162. World Scientific, Singapore, 1995.

[71] Marko Petkovšek and Herbert S. Wilf. On a conjecture of Ira Gessel. Technical Report 0807.3202, arXiv, July 2008.

[72] Marko Petkovšek, Herbert S. Wilf, and Doron Zeilberger. $A = B$. A. K. Peters, Ltd., Wellesley MA, 1996.

[73] Veronika Pillwein. *Computer Algebra Tools for Special Functions in High Order Finite Element Methods*. PhD thesis, Johannes Kepler University Linz, 2008.

[74] Earl D. Rainville. *Special Functions*. Macmillan, New York, 1960.

[75] Claude Sabbah. Systèmes holonomes d'équations aux q-différences. In Masaki Kashiwara, Pierre Schapira, and Teresa M. Fernandes, editors, *Proceedings of the International Conference on D-Modules and Microlocal Geometry, 1990, University of Lisbon*, pages 125–147. Walter de Gruyter & Co., Berlin, 1993.

[76] Mutsumi Saito, Bernd Sturmfels, and Nobuki Takayama. *Gröbner Deformations of Hypergeometric Differential Equations*, volume 6 of *Algorithms and Computation in Mathematics*. Springer-Verlag, Berlin, 2000.

[77] Bruno Salvy and Paul Zimmermann. GFUN: A Maple package for the manipulation of generating and holonomic functions in one variable. Rapport Technique 143, Institut National de Recherche en Informatique et en Automatique (INRIA), 78153 Le Chesnay Cedex, France, 1992.

[78] Carsten Schneider. Symbolic summation assists combinatorics. *Séminaire Lotharingien de Combinatoire*, 56:1–36, April 2007. Article B56b, http://www.risc.uni-linz.ac.at/research/combinat/software/Sigma/.

[79] Richard Stanley. Differentiably finite power series. *European Journal of Combinatorics*, 1:175–188, 1980.

[80] Richard Stanley. A baker's dozen of conjectures concerning plane partitions. In Gilbert Labelle and Pierre Leroux, editors, *Combinatoire Énumérative*, number 1234 in Lecture Notes in Mathematics, pages 285–293. Springer-Verlag Berlin/Heidelberg/New York, 1986.

[81] John Stembridge. The enumeration of totally symmetric plane partitions. *Advances in Mathematics*, 111:227–243, 1995.

[82] Thomas Stoll and Jiang Zeng. On the total weight of weighted matchings of segment graphs. *The Electronic Journal of Combinatorics*, 16(1):R56, 2009.

[83] Gilbert Strang. A Legendre identity. Personal e-mail to Peter Paule, 19 April 2002.

[84] Nobuki Takayama. An algorithm of constructing the integral of a module—an infinite dimensional analog of Gröbner basis. In *ISSAC'90: Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 206–211, New York, NY, USA, 1990. ACM.

[85] Nobuki Takayama. Gröbner basis, integration and transcendental functions. In *ISSAC'90: Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 152–156, New York, NY, USA, 1990. ACM.

[86] Nobuki Takayama. An approach to the zero recognition problem by Buchberger algorithm. *Journal of Symbolic Computation*, 14(2-3):265–282, 1992.

[87] Harrison Tsai. *Algorithms for algebraic analysis.* PhD thesis, University of California, Berkeley, 2000.

[88] Harrison Tsai. Weyl closure of a linear differential operator. *Journal of Symbolic Computation*, 29(4-5):747–775, 2000.

[89] Alfred van der Poorten. A proof that Euler missed—Apéry's proof of the irrationality of $\zeta(3)$. An informal report. *Mathematical Intelligencer*, 1:195–203, 1979.

[90] Kurt Wegschaider. Computer generated proofs of binomial multi-sum identities. Master's thesis, RISC, Johannes Kepler University Linz, May 1997. http://www.risc.uni-linz.ac.at/research/combinat/software/MultiSum/.

[91] Herbert S. Wilf and Doron Zeilberger. An algorithmic proof theory for hypergeometric (ordinary and "q") multisum/integral identities. *Inventiones Mathematicae*, 108(1):575–633, 1992.

[92] Doron Zeiberger. A fast algorithm for proving terminating hypergeometric identities. *Discrete Mathematics*, 80(2):207–211, 1990.

[93] Doron Zeilberger. A holonomic systems approach to special functions identities. *Journal of Computational and Applied Mathematics*, 32(3):321–368, 1990.

[94] Doron Zeilberger. The method of creative telescoping. *Journal of Symbolic Computation*, 11:195–204, 1991.

[95] Doron Zeilberger. The HOLONOMIC ANSATZ II. Automatic DISCOVERY(!) and PROOF(!!) of Holonomic Determinant Evaluations. *Annals of Combinatorics*, 11:241–247, 2007.

[96] Doron Zeilberger. (Okada) is nice, but to get the promised $100, you need to do (Soichi) first. Personal e-mail communication with Christoph Koutschan, 2009.

[97] B. Zürcher. Rationale Normalformen von pseudo-linearen Abbildungen. Diplomarbeit, ETH Zürich, 1994.

Index

Curriculum Vitae

Personal Data

Name	Christoph Bernhard Koutschan
Date of birth	December 12, 1978
Place of birth	Dillingen an der Donau, Germany
Nationality	German

Contact

Affiliation	Research Institute for Symbolic Computation (RISC) Johannes Kepler University Linz Altenberger Straße 69 4040 Linz, Austria http://www.risc.uni-linz.ac.at
E-mail	Christoph@Koutschan.de
Homepage	http://www.risc.uni-linz.ac.at/people/ckoutsch/

Education

1998	Abitur (high school graduation) at Gymnasium Casimirianum Coburg
1999–2005	Studies in Computer Science at Friedrich-Alexander University Erlangen-Nürnberg
2005	Diploma degree (Dipl.-Inf.) Diploma thesis: *Regular Languages and Their Generating Functions: The Inverse Problem.* (advised by Prof. Dr. Volker Strehl)
2005–2009	Doctorate studies in Symbolic Computation at RISC-Linz