Non-commutative Elimination in Ore Algebras Proves Multivariate Identities

Frédéric Chyzak, Bruno Salvy

J. Symbolic Computation (1998) **11**, 1–000

Non-commutative Elimination in Ore Algebras Proves Multivariate Identities

FRÉDÉRIC CHYZAK AND BRUNO SALVY

*INRIA-Rocquencourt and École polytechnique (*France*)*
Frederic.Chyzak@inria.fr

*INRIA-Rocquencourt (*France*)*
Bruno.Salvy@inria.fr

(Received 7 January 1998)

Many computations involving special functions, combinatorial sequences or their q-analogues can be performed using linear operators and simple arguments on the dimension of related vector spaces. In this article, we develop a theory of ∂-finite sequences and functions which provides a unified framework to express algorithms for computing sums and integrals and for the proof or discovery of multivariate identities. This approach is vindicated by an implementation.

Introduction

Computer algebra consists in performing calculations on mathematical objects represented by a finite amount of information. A class of computer algebra objects is especially useful when it is possible to recognize whether two members of the class are identical or not. Zeilberger (1990b) showed that a large set of combinatorial identities can be proved using properties of the class of *P-finite* functions and sequences and the important subclass of *holonomic* functions.

A function is *P-finite* when the set of its partial derivatives spans a finite-dimensional vector space over the field of rational functions. Computationally, a P-finite function is specified by a system of linear differential equations (linear relations between the partial derivatives) and a finite number of initial conditions. Proving that a P-finite function is zero requires finding a linear system it satisfies and checking that sufficiently many of its initial conditions are zero. Given an algorithm for the difference, this provides an equality test.

Consider for instance the function

$$f(z,t) = \frac{\cos(zt)}{\sqrt{1-t^2}}. \tag{0.1}$$

This function is P-finite since the set of its derivatives $\{D_z^i D_t^j \cdot f\}$ for $(i,j) \in \mathbb{N}^2$ generates a finite-dimensional vector space over the field of rational functions $\mathbb{Q}(z,t)$. (In this article, we use D_x to denote the partial differential operator $\partial/\partial x$ with respect to x,

0747–7171/90/000000 + 00 $03.00/0

and a dot to denote the action of a linear operator on a function.) This vector space admits $\{f, D_z \cdot f\}$ as a basis, as follows from the following system of linear partial differential equations:

$$D_z^2 \cdot f + t^2 f = 0, \qquad t(t^2-1)D_t \cdot f + z(1-t^2)D_z \cdot f + t^2 f = 0. \tag{0.2}$$

From a small database of differential equations like those satisfied by the trigonometric functions sin and cos, the computation of such a system is made possible by algorithms making effective the numerous closure properties (sum, product, algebraic substitution) enjoyed by the class of P-finite functions. These algorithms will be described in a general setting in Section 2.

Now, consider the following specification for the Bessel function $J_0(z)$ of the first kind and order zero:

$$\mathcal{A} \cdot y = 0, \qquad \text{where} \qquad \mathcal{A} \cdot y = zD_z^2 \cdot y + D_z \cdot y + zy, \tag{0.3}$$

with initial conditions $y(0) = 1$, $y'(0) = 0$. Then, the identity

$$J_0(z) = \frac{2}{\pi} \int_0^1 \frac{\cos(zt)}{\sqrt{1-t^2}}\, dt, \tag{0.4}$$

follows from noticing that

$$\mathcal{A} \cdot f = D_t \cdot \left(0 \cdot f - \frac{1-t^2}{t} D_z \cdot f\right) = D_t \cdot \left(\sin(zt)\sqrt{1-t^2}\right), \tag{0.5}$$

(which can be checked directly) and integrating from 0 to 1: this shows that the right-hand side of (0.4) also satisfies (0.3). It is easily seen that the initial conditions are again $y(0) = 1$ and $y'(0) = 0$, therefore by the Cauchy-Lipschitz theorem we have proved (0.4). Equation (0.3) can be derived from the system (0.2) by an algorithm called *creative telescoping* which we describe in Section 3. One of the variants of this algorithm also produces the corresponding *certificate* (0.5). In general, this certificate consists in: rational functions that are the coefficients of the operator $\mathcal{A}$; rational functions that are the coordinates of an antiderivative of $\mathcal{A} \cdot f$ in a specified basis of the (finite-dimensional) vector space generated by f and its derivatives; and additional information from which the proof of the identity is reduced to manipulation of rational functions, as described in §3.1.

In a very similar way, *P-finite sequences* are defined as sequences such that the set of sequences obtained by shifting the indices spans a finite-dimensional vector space over the field of rational functions. Identities involving such sequences are proved by computing systems of recurrences and sufficiently many initial conditions. There again, the computation of these systems is made possible by the closure properties enjoyed by the class of P-finite sequences.

An example of a P-finite sequence with respect to n and m is

$$a_{n,m} = (-1)^m \frac{\Gamma(\alpha+n-m)}{m!(n-2m)!}(2x)^{n-2m}. \tag{0.6}$$

This sequence is P-finite since the set of its shifts $(a_{n+i,m+j})$ for $(i,j) \in \mathbb{N}^2$ generates a finite-dimensional vector space over the field of rational functions $\mathbb{K}(n,m)$ where $\mathbb{K} = \mathbb{Q}(x,\alpha)$. This one-dimensional vector space admits $\{a_{n,m}\}$ as a basis, as follows from the following system of linear recurrence equations:

$$\begin{cases} (n+1-2m)a_{n+1,m} - 2(\alpha+n-m)xa_{n,m} = 0, \\ 4(m+1)(\alpha+n-m-1)x^2 a_{n,m+1} + (n-2m)(n-2m-1)a_{n,m} = 0. \end{cases} \tag{0.7}$$

In this case and similar ones when the vector space is of dimension one, the sequence is called *hypergeometric*.

Again, a small database of recurrences like that satisfied by the factorial, adjoined to a few algorithms making effective the closure properties enjoyed by the class of P-finite sequences makes the computation of such a system very easy, even in not so trivial cases. Let $C_n^{(\alpha)}(x)$ denote the n-th ultraspherical (or Gegenbauer) polynomial, of which a possible specification is the recurrence equation $(\mathcal{A} \cdot u)(n) = 0$, where

$$(\mathcal{A} \cdot u)(n) = (n+2)u_{n+2} - 2(n+\alpha+1)xu_{n+1} + (n+2\alpha)u_n, \tag{0.8}$$

with initial conditions $u_0 = 1$, $u_1 = 2x\alpha$. Then, the identity

$$C_n^{(\alpha)}(x) = \frac{1}{\Gamma(\alpha)} \sum_{m=0}^{\lfloor n/2 \rfloor} (-1)^m \frac{\Gamma(\alpha+n-m)}{m!(n-2m)!} (2x)^{n-2m} \tag{0.9}$$

can be proved by summation over m after noting that $(\mathcal{A} \cdot a)(n,m) = b_{n,m+1} - b_{n,m}$, with

$$b_{n,m} = -4\frac{(n+2\alpha)(n-m+\alpha)}{(n+1-2m)(n+2-2m)} mx^2 a_{n,m}, \tag{0.10}$$

which can be checked directly. Again, algorithms to perform creative telescoping are able to compute the operator (0.8) as well as Eq. (0.10), directly from the system (0.7).

In the univariate case, it is well-known that P-finiteness of functions and P-finiteness of sequences are equivalent via generating functions. This gives rise to various closure properties by going back and forth between sequences and their generating functions. The experience gained from an implementation of these operations in the univariate case (Salvy and Zimmermann, 1994) shows that the algorithms used in the differential and in the difference case are essentially identical. The reason for this is that the algorithms use very few specific properties of the shift operator and differentiation operator.

In this article, we use the framework of *Ore polynomials* and *skew polynomial rings* to define a general notion of *∂-finiteness* which generalizes P-finiteness of functions and sequences. These Ore polynomials capture the properties of linear operators that are necessary to express our algorithms. The notion of ∂-finiteness makes it easy to describe mixed differential-difference systems which were studied by Zeilberger (1990b) and Takayama (1989) and linear q-equations which up to now have mainly been studied in the q-hypergeometric case (equations of order 1). Our generalization makes it possible to have a general program working at the level of Ore polynomials. New types of systems of operators can be defined by adding *very few lines* to the existing program.

For instance, Jacobi's triple product identity

$$\prod_{k=1}^{\infty} (1-q^k)(1+zq^{k-1})(1+q^k/z) = \sum_{k=-\infty}^{\infty} q^{\binom{k}{2}} z^k, \tag{0.11}$$

is reflected by simple identities in an algebra of operators containing the operators q, z and Q_k, of multiplication by q, z and q^k, respectively, as well as the operator S_k

corresponding to a shift of the index k and the operator $H_{q,z}$ of q-dilation, whose action is

$$(H_{q,z} \cdot f)(z) = f(qz).$$

In the Ore algebra framework, these operators are not defined by their actions but by the following *commutation rules* among polynomials in the algebra:

$$S_k P(q, z, Q_k) = P(q, z, qQ_k) S_k, \tag{0.12}$$

$$H_{q,z} P(q, z, Q_k) = P(q, qz, Q_k) H_{q,z}, \tag{0.13}$$

for any polynomial P in q, z and Q_k. Then the only information which will be needed about the summand $u_k(z) = q^{\binom{k}{2}} z^k$ in the right-hand side of (0.11) is an *annihilating system* such as

$$S_k - zQ_k, \qquad H_{q,z} - Q_k. \tag{0.14}$$

From this system it is apparent that the sequence of functions u is ∂-*finite* with respect to S_k and $H_{q,z}$, in the sense that the system $\{S_k^i H_{q,z}^j \cdot u, (i,j) \in \mathbb{N}^2\}$ generates a finite-dimensional vector space over the field of rational functions $\mathbb{Q}(q, z, q^k)$. It is indeed of dimension 1 and admits $\{u\}$ as a basis. We prove in Section 2 that the set of ∂-finite functions is closed under sum and product and we describe algorithms performing the corresponding operations. Thus, in this instance a small database of operators vanishing on q^k, q^{k^2}, z^k would make it possible to derive the system (0.14) automatically. Note that as opposed to the general algorithms we discuss here, this database approach depends on the specific algebra of functions under study.

Creative telescoping can also be generalized to some extent, and we provide several algorithms based on *non-commutative elimination* to do so. In the example above, one of these algorithms readily finds the obvious operator

$$\mathcal{A} - (S_k - 1), \qquad \text{where} \qquad \mathcal{A} = zH_{q,z} - 1$$

in the *left ideal* generated by the system (0.14) over the operator algebra in S_k, $H_{q,z}$ with coefficients in $\mathbb{Q}(q, z, Q_k)$, which will be formally defined in §1.1. Note that $\mathcal{A}$ does not involve Q_k.

Operators of this type give its name to the method of creative telescoping: one has

$$(\mathcal{A} \cdot u)(k, z) - (u_{k+1}(z) - u_k(z)) = 0,$$

so that summing with respect to k makes the rightmost summand "telescope". Now interchanging summation and $\mathcal{A}$ in this equation shows that $\mathcal{A}$ annihilates the right-hand side of Jacobi's triple product identity (0.11). It is easy to check that the left-hand side of (0.11) is also annihilated by $\mathcal{A}$. To conclude the proof of the identity, one needs to show that two "initial conditions" coincide. We do not address this final step: since our algorithms deal with Ore polynomials, initial conditions lie outside their scope. Indeed, for each Ore algebra, initial conditions require specific algorithms and a specific implementation. This should not be viewed as a defect of the method, but rather as a reflection of the common fact that constants are more difficult to handle than functions in symbolic computation. In this particular example, several methods are known to deal with this problem (Askey, 1992). (Another approach to the automatic proving of this identity consists in proving a finite version of it, which involves ordinary recurrences on the index of summation, whose initial conditions are more easily checked, see (Paule, 1994).)

Our elimination algorithms are based on a generalization of the theory of *Gröbner* *bases*. As Takayama (1989) noticed in the differential-difference case, and as was developed by Kandri-Rody and Weispfenning (1990) in the more general setting of polynomial rings of solvable type, Buchberger's algorithm for Gröbner bases can be adapted to our non-commutative context. These bases furnish normal forms and an algorithm for elimination, which we use for the creative telescoping described above. In the special case of the Weyl algebra (differential operators), creative telescoping is guaranteed to succeed for a subclass of systems of equations classically called *holonomic*. The precise definition of holonomy is technical and will not be needed in this article. It is related to the minimality of the Berstein dimension of modules over Weyl algebras. Moreover, if a function is P-finite, there exists a holonomic system annihilating it (Kashiwara, 1978). It follows that the termination of our algorithms is guaranteed in the differential case (see Section 4 for further details). This property is the only consequence of holonomy which we shall use in this article. Results obtained by holonomy can also be translated to results for sequences via generating functions (Zeilberger, 1990b). A similar theory exists for q-analogues (Sabbah, 1993), but we do not have a corresponding notion of holonomy in the general case of Ore polynomials. The algorithms we give for creative telescoping are therefore not guaranteed to succeed. We give two such algorithms. The first one is a Gröbner basis computation. It may be slow but will always terminate, either successfully or detecting that the algorithm has failed. The second one is a generalization of an algorithm due to Takayama (1990a, 1990b). It is faster in the cases where creative telescoping is possible. In other cases, it will fail to terminate. In practice, failure often means that not enough information was encapsulated in the input, which can often be detected in advance.

The algorithmic study of holonomic systems in the differential and difference cases was initiated by (Zeilberger, 1990b), building up on earlier work of M. C. Fasenmyer (1945, 1947, 1949). In that article, Zeilberger relies on a non-commutative version of Sylvester's dialytic elimination method to perform creative telescoping. This will be presented in a different form in Section 1. He also mentions the great advantage that would be obtained by the use of Gröbner bases for elimination instead of Sylvester's dialytic elimination. Our work takes this direction and extends this approach to other contexts than the mixed difference-differential equations. The first use of Gröbner bases to deal with holonomic systems appears in (Galligo, 1985) and was later elaborated by Takayama (1989) for mixed differential-difference systems. This approach makes it possible to eliminate several indeterminates simultaneously in an operator algebra built on more than two operators. Since (Zeilberger, 1990b), most of the work in this area has been focused on specialized algorithms in the hypergeometric case (Zeilberger, 1990a, 1991b); extensions to the q-hypergeometric case with particular emphasis on the discrete case (Wilf and Zeilberger, 1992a, 1992b); and even extensions to multibasic q-hypergeometric identities (Riese, 1996) and some Abel-type identities (Ekhad and Majewicz, 1996). A very nice account of most of these algorithms for hypergeometric identities is the recent book by Petkovšek, Wilf and Zeilberger (1996). The general holonomic case however has not received much attention since Zeilberger's first fundamental article (1990b).

Summarizing, our main contributions are: (i) the use of skew polynomial rings so as to encapsulate different types of linear operators in a single algebraic setting and as to unify existing algorithms for these different frameworks; (ii) the use of a general theory of Gröbner bases to develop algorithms for ∂-finite functions at a general level, thereby setting the emphasis back on general holonomic and ∂-finite functions, as opposed to hypergeometric and q-hypergeometric ones; (iii) the extension and improvement of an algorithm by Takayama for fast creative telescoping; (iv) a Maple implementation *Mgfun*, due to (Chyzak, 1994)[†], which makes it easy to work in various mixed contexts with a single program. All the operations described in this paper are illustrated by examples using this package.

The present article is organized as follows. In Section 1, Ore polynomials are introduced and the algorithmic tools to work with them are provided. In Section 2, we define ∂-finiteness and we use Gröbner bases to make some of the closure properties effective. When interpreted in terms of ∂-finite functions, these closure properties correspond to closure under addition, product and the action of operators. Section 3 is devoted to the generalization of creative telescoping, which makes it possible to compute definite sums and integrals. The algorithms we use for creative telescoping in Ore algebras are restricted to Ore algebras of a special type, called *polynomial* Ore algebras, analogous to the Weyl algebra. This is where an analogous notion of holonomy is still missing. We conclude in Section 4 with a more extensive discussion of holonomy, q-holonomy and the relation between polynomial Ore algebras and the general case.

1. Non-commutative algebras of operators

Introduction

The classical Leibniz rule states that for any two functions f and g of a differential algebra,

$$(fg)' = fg' + f'g.$$

In terms of operators, this reads as

$$D_x f = f D_x + f', \tag{1.1}$$

where f and f' now stand for the operators of multiplication by the functions f and f', respectively. In the case of finite differences, the following functional identity

$$(fg)(x+1) - (fg)(x) = f(x+1)\,(g(x+1) - g(x)) + (f(x+1) - f(x))\,g(x)$$

reads as

$$\Delta \cdot (fg) = (S \cdot f)(\Delta \cdot g) + (\Delta \cdot f)g,$$

where $\Delta = S - 1$ in terms of the shift operator defined by $(S \cdot f)(x) = f(x+1)$. Equivalently, in terms of operators one has the commutation

$$\Delta f = (S \cdot f)\Delta + (\Delta \cdot f), \tag{1.2}$$

where f, $S \cdot f$ and $\Delta \cdot f$ have to be regarded as the operators of multiplication by the corresponding functions. Similary, the shift operator satisfies the following commutation

$$Sf = (S \cdot f)S, \tag{1.3}$$

which reflects

$$[S \cdot (fg)](x) = f(x+1)g(x+1).$$

[†] The packages mentioned in this article are available
by anonymous ftp from `ftp.inria.fr:INRIA/Projects/algo/programs`
or at the URL `http://www-rocq.inria.fr/algo/libraries/libraries.html`.

Equations (1.1), (1.2), (1.3) suggest a general pattern for commutations:

$$\partial f = \sigma(f)\partial + \delta(f), \tag{1.4}$$

where, as indicated by the examples above, the operators σ and δ are closely related to ∂. Since we are considering *linear* operators ∂, this commutation implies that σ and δ should be linear. Other constraints on suitable σ and δ are obtained by considering each side of $(\partial f)g = \partial(fg)$, yielding the identity between operators

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

Equating coefficients of ∂ makes it natural to demand that σ be a ring homomophism and that δ be a σ-derivation, i.e., δ is linear and satisfies

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

for any functions f and g.

Equation (1.4) and these constraints on σ and δ form the basis of a general treatment of linear operators developed by Ore (1933) under the name *skew polynomial rings*. We give the basic definitions and properties in §1.1.

In order to compute definite integrals and sums by creative telescoping in Section 3, we need a way to eliminate the variable with respect to which summation or integration is performed. We therefore first need a way to consider this variable in the framework of skew polynomial rings. This is achieved by considering a special case of skew polynomial rings which we call Ore algebra. These are defined in §1.1. In these algebras, the operator ∂ has a commutation rule involving another variable X. This new variable will often correspond to multiplication by x. For instance, the commutation rule for differentiation with respect to x can be expressed as

$$D_x P(X) = P(X)D_x + P'(X), \tag{1.5}$$

where the action of X is $(X \cdot f)(x) = xf(x)$ and $P'(X)$ is the formal derivative of the polynomial $P(X)$. In the difference case, we have

$$\Delta_x P(X) = P(X+1)\Delta_x + P(X+1) - P(X), \tag{1.6}$$

where again the action of X is $(X \cdot f)(x) = xf(x)$. Another interesting example is the difference operator when applied to functions of q and q^x; then one has

$$\Delta_x P(X) = P(qX)\Delta_x + P(qX) - P(X), \tag{1.7}$$

where now the action of X is $(X \cdot f)(q, q^x) = q^x f(q, q^x)$. Table 2 lists examples of pairs (X, ∂) of variables that can be treated in this framework, while Table 1 shows the operator viewpoint on these examples.

In §1.1, we define skew polynomial rings and Ore algebras. The link between these algebras and operators is clarified in §1.2. As shown by Ore, the Euclidean algorithm works in skew polynomial rings, and it provides an algorithmic way to eliminate the indeterminate ∂ between two operators. This is described in §1.3. Several examples are considered in §1.4, including applications to contiguity relations of generalized hypergeometric functions and the link with Sylvester's dialytic elimination. In §1.5, we come to the main algorithmic tool of this article: non-commutative Gröbner bases. Using an extension of results in (Kandri-Rody and Weispfenning, 1990) due to Kredel (1993), we show that for a large class of Ore algebras, (left) Gröbner bases can be computed by a non-commutative version of Buchberger's algorithm, with possible restrictions on the term orders.

Operator	$(\partial \cdot f)(x)$	$(X \cdot f)(x)$	$(\partial \cdot fg)(x)$
Differentiation	$f'(x)$	$xf(x)$	$f(x)(\partial \cdot g)(x) + (\partial \cdot f)(x)g(x)$
Shift	$f(x+1)$	$xf(x)$	$f(x+1)(\partial \cdot g)(x)$
Difference	$f(x+1) - f(x)$	$xf(x)$	$f(x+1)(\partial \cdot g)(x) + (\partial \cdot f)(x)g(x)$
q-Dilation	$f(qx)$	$xf(x)$	$f(qx)(\partial \cdot g)(x)$
Continuous q-difference	$f(qx) - f(x)$	$xf(x)$	$f(qx)(\partial \cdot g)(x) + (\partial \cdot f)(x)g(x)$
q-Differentiation	$\frac{f(qx)-f(x)}{(q-1)x}$	$xf(x)$	$f(qx)(\partial \cdot g)(x) + (\partial \cdot f)(x)g(x)$
q-Shift	$f(x+1)$	$q^x f(x)$	$f(x+1)(\partial \cdot g)(x)$
Discrete q-difference	$f(x+1) - f(x)$	$q^x f(x)$	$f(x+1)(\partial \cdot g)(x) + (\partial \cdot f)(x)g(x)$
Eulerian operator	$xf'(x)$	$xf(x)$	$f(x)(\partial \cdot g)(x) + (\partial \cdot f)(x)g(x)$
e^x-Differentiation	$f'(x)$	$e^x f(x)$	$f(x)(\partial \cdot g)(x) + (\partial \cdot f)(x)g(x)$
Mahlerian operator	$f(x^p)$	$xf(x)$	$f(x^p)(\partial \cdot g)(x)$
Divided differences	$\frac{f(x)-f(a)}{x-a}$	$xf(x)$	$f(a)(\partial \cdot g)(x) + (\partial \cdot f)(x)g(x)$

Table 1. Ore operators and their Leibniz rules

Operator	$\sigma(P)(X)$	$\delta(P)(X)$	$\partial P(X)$	∂X
Differentiation	$P(X)$	$P'(X)$	$P(X)\partial + P'(X)$	$X\partial + 1$
Shift	$P(X+1)$	0	$P(X+1)\partial$	$(X+1)\partial$
Difference	$P(X+1)$	$(\Delta \cdot P)(X)$	$P(X+1)\partial + (\Delta \cdot P)(X)$	$(X+1)\partial + 1$
q-Dilation	$P(qX)$	0	$P(qX)\partial$	$qX\partial$
Cont. q-difference	$P(qX)$	$P(qX) - P(X)$	$P(qX)\partial + P(qX) - P(X)$	$qX\partial + (q-1)X$
q-Differentiation	$P(qX)$	$\frac{P(qX)-P(X)}{(q-1)X}$	$P(qX)\partial + \frac{P(qX)-P(X)}{(q-1)X}$	$qX\partial + 1$
q-Shift	$P(qX)$	0	$P(qX)\partial$	$qX\partial$
Discr. q-difference	$P(qX)$	$P(qX) - P(X)$	$P(qX)\partial + P(qX) - P(X)$	$qX\partial + (q-1)X$
Eulerian operator	$P(X)$	$XP'(X)$	$P(X)\partial + XP'(X)$	$X\partial + X$
e^x-Differentiation	$P(X)$	$XP'(X)$	$P(X)\partial + XP'(X)$	$X\partial + X$
Mahlerian operator	$P(X^p)$	0	$P(X^p)\partial$	$X^p\partial$
Divided differences	$P(a)$	$\frac{P(X)-P(a)}{X-a}$	$P(a)\partial + \frac{P(X)-P(a)}{X-a}$	$a\partial + 1$

Table 2. Corresponding Ore algebras and their commutation rules

1.1. Definitions

Since all algebras of interest to our study are skew algebras of operators, we adopt the convention that the words *ring* and *field* always refer to possibly skew rings and fields. We specify *commutative ring* or *commutative field* when necessary. Moreover, all rings under consideration in this paper are of characteristic 0.

Definition 1.1. *Let $\mathbb{A}$ be an integral domain, i.e., a ring without zero-divisors. The* skew polynomial ring $\mathbb{A}[\partial; \sigma, \delta]$ *is the set of polynomials in ∂ with coefficients in $\mathbb{A}$, with usual addition and a product defined by associativity from the following commutation rule*

$$\forall a \in \mathbb{A} \qquad \partial a = \sigma(a)\partial + \delta(a). \tag{1.8}$$

Here, σ is a ring endomorphism of $\mathbb{A}$ and δ is a σ-derivation operator, i.e., an additive *endomorphism of $\mathbb{A}$ which satisfies the following Leibniz rule:*

$$\forall a, b \in \mathbb{A} \qquad \delta(ab) = \sigma(a)\delta(b) + \delta(a)b. \tag{1.9}$$

Using the commutation rule (1.8), any element of $\mathbb{A}[\partial; \sigma, \delta]$ can be uniquely rewritten in the form $\sum_{i=0}^{d} a_i \partial^i$, i.e., with the ∂'s on the right. Degree in ∂ and coefficients are then defined as in the commutative case, the coefficients being on the left side of the monomials.

One reason for studying these skew polynomial rings is that operations which can be performed in them need only be implemented once and then apply equally to linear differential equations, linear difference equations or their q-analogues.

The following proposition is due to the existence of a degree function and leads to the multivariate case.

Proposition 1.1. *(Cohn, 1971, p. 35) The skew polynomial ring $\mathbb{A}[\partial; \sigma, \delta]$ is an integral domain.*

By choosing appropriate integral domains $\mathbb{A}$, we can use this proposition in conjunction with Definition 1.1 to construct various multivariate skew polynomial rings. Several of these choices will be useful in the sequel. In particular, we have the following important special cases.

Definition 1.2. *Let $\mathbb{K}$ be a field and $\mathbb{A} = \mathbb{K}[x_1, \ldots, x_s]$ be a commutative polynomial ring (with $\mathbb{A} = \mathbb{K}$ when $s = 0$). The skew polynomial ring $\mathbb{A}[\partial_1; \sigma_1, \delta_1] \cdots [\partial_r; \sigma_r, \delta_r]$ is called an* Ore algebra *when the σ_i's and δ_j's commute for $1 \le i, j \le r$ with $i \ne j$, and satisfy $\sigma_i(\partial_j) = \partial_j$, $\delta_i(\partial_j) = 0$ for $i > j$. When $s = 0$, it is denoted $\mathbb{K}[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$, while for $s > 0$ it is called a* polynomial Ore algebra *and is denoted $\mathbb{K}[\mathbf{x}][\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$.*

Note that this definition does not demand that the elements of $\mathbb{K}$ should commute with the ∂_i's. Thus the case of an Ore algebra $\mathbb{Q}(\mathbf{x})[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$ is accomodated by the definition, by taking $\mathbb{K} = \mathbb{Q}(\mathbf{x})$ as the field. The conditions on the σ_i's and δ_j's imply that the ∂_i's commute. This fact is crucial for our subsequent treatment by Gröbner bases in §1.5.

Examples of Ore algebras are given in Table 2. In all the cases under consideration in this table, the Ore algebra is of the form $\mathbb{K}[\partial; \sigma, \delta]$ where $\mathbb{K}$ contains $\mathbb{Q}(X)$. By associativity and with the additional assumption that σ and δ commute, relation (1.9) then induces

$$\forall p \ge 1 \qquad \delta(X^p) = \delta(X) \sum_{k=0}^{p-1} \sigma(X)^k X^{p-1-k}. \tag{1.10}$$

A similar but more complicated formula is easily derived when σ and δ do not commute. This shows that σ and δ are completely determined over $\mathbb{Q}[X]$ by their values on X. In other words, the last column of Table 2 is sufficient to determine the three preceding ones. Assuming additionally that σ is injective, σ and δ extend uniquely to $\mathbb{Q}(X)$ as follows from expanding $\partial = \partial f f^{-1}$, which yields

$$\partial f^{-1} = \sigma(f)^{-1} \partial - \sigma(f)^{-1} \delta(f) f^{-1}.$$

Note that distinct algebras of operators can share the same commutation rule: see for example the cases of the Eulerian operator and of the e^x-differentiation in Tables 1 and 2.

Since very often in practice the variable X under consideration represents either the operator of multiplication by x or of multiplication by q^x, we shall not refrain in the sequel from writing x or q^x in place of X.

Example. Weyl algebras $\mathbb{K}[x_1,\dots,x_n][D_{x_1};1,D_{x_1}]\cdots[D_{x_n};1,D_{x_n}]$ are a special case of polynomial Ore algebras, obtained when the D_{x_i}'s have the same commutation rules as the usual partial differentiation operators.

Example. In $\mathbb{Q}(a,b)[n,x][S_n;S_n,0][D_x;1,D_x]$, where S_n denotes the shift operator with respect to n and D_x denotes differentiation with respect to x, the Jacobi polynomials $P_n^{(a,b)}(x)$ are annihilated by

$$\begin{aligned} G_1 = 2(n+2)(n+a+b+2)(2n+a+b+2)S_n^2 \\ -\left((2n+a+b+3)\left(a^2-b^2\right)\right. \\ \left.+(2n+a+b+2)(2n+a+b+3)(2n+a+b+4)x\right)S_n \\ +2(n+a+1)(n+b+1)(2n+a+b+4), \end{aligned} \tag{1.11}$$

$$\begin{aligned} G_2 = (2n+a+b+2)\left(1-x^2\right)S_nD_x - (n+1)\left(a-b-(2n+a+b+2)x\right)S_n \\ -2(n+a+1)(n+b+1), \end{aligned} \tag{1.12}$$

$$G_3 = (1-x^2)D_x^2 + (b-a-(a+b+2)x)D_x + n(n+a+b+1). \tag{1.13}$$

This is the only information our algorithms will use to deal with Jacobi polynomials. Initial conditions must be treated separately, if needed. Note that the information provided by the operators G_1, G_2, G_3 is actually redundant, and we give algorithms to deduce either G_1 or G_3 from both other ones below.

Example. The Ore algebra $\mathbb{Q}(q)[n,q^n][S_n;S_n,0]$ with the commutation rule

$$S_n n^k (q^n)^\ell = (n+1)^k q^\ell (q^n)^\ell S_n$$

is well-suited for certain q-computations. For instance, the sequence $u_n = n!q^{n^2}$ is annihilated by

$$S_n - (n+1)q\,(q^n)^2.$$

Example. In the Ore algebra $\mathbb{Q}(q)[z,Q_k][S_k;S_k,0][H_{q,z};\sigma,\delta]$ that is built on the relations (0.12–0.13) and on the applications σ and δ defined as in Table 2 by

$$\sigma(P)(z) = P(qz), \qquad \text{and} \qquad \delta(P)(z) = \frac{P(qz)-P(z)}{(q-1)z},$$

the function $q^{k^2}z^k$ is annihilated by Eq. (0.14).

1.2. Operators, ideals and modules

In this work, an Ore algebra $\mathbb{O}$ (resp. a polynomial Ore algebra) is interpreted as a ring of operators. This is achieved when ∂_i, σ_i and δ_i act as linear endomorphisms on a $\mathbb{K}$-algebra (resp. a $\mathbb{K}[x_1,\dots,x_s]$-algebra) $\mathcal{F}$ of functions, power series, sequences, distributions, etc. Then Eq. (1.8) extends to the following Leibniz rule for products

$$\forall f,g\in\mathcal{F} \qquad \partial_i\cdot(fg) = \sigma_i(f)\partial_i\cdot g + \delta_i(f)g. \tag{1.14}$$

This makes $\mathcal{F}$ an $\mathbb{O}$-module, the product in $\mathbb{O}$ acting as the composition of operators. The actions of the operators corresponding to important Ore algebras are given in Table 1. In the remainder of this article, we use the word "function" to denote any object on which the elements of an Ore algebra act.

This interpretation motivates the study of ideals of Ore algebras. Algebraically, an object of interest is the *left ideal* $\operatorname{Ann} f \subseteq \mathbb{O}$ of Ore polynomials which vanish on some $f \in \mathcal{F}$. It is called the *annihilating ideal* of f. Most of the operations we consider below consist in finding elements of this ideal which satisfy special properties, or in finding elements of an ideal of operators annihilating a function related to f.

Correspondingly, the $\mathbb{O}$-module $\mathbb{O} \cdot f \simeq \mathbb{O} / \operatorname{Ann} f$ encapsulates much of the structure of the *pseudo-derivatives* $\partial_1^{\alpha_1} \cdots \partial_r^{\alpha_r} \cdot f$. Computationally, all calculations take place in this module. Although all the algorithms we present below have an interpretation in terms of operators, the existence of a specific algebra $\mathcal{F}$ is not even needed. The algorithms can all be stated at the level of ideals $\mathfrak{I}$ of $\mathbb{O}$ and modules $\mathbb{O}/\mathfrak{I}$. The rôle of the function f is then taken over by the element 1 in $\mathbb{O}/\mathfrak{I}$.

1.3. Euclidean division

Two algorithms allow us to perform most of our computations. The first one is left Euclidean division which leads to an extended gcd algorithm for gcrd's. (We write gcrd for *greatest common right divisor*, and lclm for *least common left multiple*.) The second one generalizes the Euclidean algorithm and consists in a suitably modified version of Buchberger's algorithm for Gröbner bases. In this section and the next two ones, we detail both these algorithms, their constraints and some of their applications.

The main results in this section are due to Ore (1933). Our only contribution is to make explicit the natural recursive algorithm below, which we use in the case of multivariate skew polynomial rings.

Recall our convention that fields may be skew. Call an *effective field* a field in which the usual ring operations are computable, and where given two non-zero elements α and β, one can compute two non-zero elements α' and β' such that $\alpha'\alpha + \beta'\beta = 0$. In the commutative case, this can be done by taking $\alpha' = \beta$ and $\beta' = -\alpha$. Let $\mathbb{S} = \mathbb{K}[\partial; \sigma, \delta]$ be a skew polynomial ring over an effective field $\mathbb{K}$. Since the elements of $\mathbb{S}$ are polynomials in ∂, performing divisions on the left makes it possible to extend the usual Euclidean algorithm to compute gcrd's as follows. Let a and b be two polynomials in $\mathbb{S}$ for which we want to compute a gcrd. Assume that the degree d_a of a in ∂ is greater than the degree d_b of b. Left-multiplying b by $\partial^{d_a - d_b}$ yields a second polynomial c of degree d_a. Let α and γ be the leading coefficients of a and c respectively. Compute two non-zero cofactors α' and γ' such that $\alpha'\alpha + \gamma'\gamma = 0$. Then $d = \alpha' a + \gamma' c$ has degree less than d_a in ∂. The same process is now applied to b and d. Repeating this process eventually yields zero. It is not difficult to prove that the last polynomial obtained before 0 is a gcrd g of a and b. (Gcrd's are defined up to a non-zero constant in $\mathbb{K}$.) Collecting the successive factors yields the extended gcd algorithm which produces u and v such that

$$ua + vb = g.$$

Lclm's are also computed using this algorithm. This is achieved by considering the last identity produced by the algorithm:

$$Ua + Vb = 0.$$

Once again, it is not difficult to prove that the polynomial Ua is a lclm of a and b. This is summarized in the following theorem, which was proved by Ore (1933) in the case of a commutative field $\mathbb{K}$, but readily extends to skew fields.

THEOREM 1.1. (ORE) *Given two elements a and b in a skew polynomial ring $\mathbb{K}[\partial;\sigma,\delta]$ over an effective field $\mathbb{K}$, the Euclidean algorithm makes it possible to compute polynomials u, v, g, U, V, with U and V non-zero, such that*

$$ua + vb = g \qquad \text{and} \qquad Ua + Vb = 0, \tag{1.15}$$

where g is a gcrd of a and b and Ua is a lclm of a and b.

A *left Ore ring* is classically defined as a ring such that for any non-zero elements a and b there exist non-zero U and V in the ring which satisfy $Ua = Vb$. As shown by the theorem above, skew polynomial rings over a field are left Ore rings. The proof of the above theorem also yields the following corollary, after rewriting all Euclidean divisions in a fraction-free way.

COROLLARY 1.1. (ORE) *If $\mathbb{A}$ is a left Ore ring, so is $\mathbb{A}[\partial;\sigma,\delta]$.*

Call an *effective left Ore ring* a left Ore ring in which the usual ring operations are computable, as well as the pair (U, V) involved in Eq. (1.15). The previous corollary can also be interpreted as an elimination property as follows.

COROLLARY 1.2. *Given two elements a and b in a skew polynomial ring $\mathbb{S} = \mathbb{A}[\partial;\sigma,\delta]$ over an effective left Ore ring $\mathbb{A}$, if there exists $(u, v) \in \mathbb{S}^2$ and $\alpha \in \mathbb{A} \setminus \{0\}$ such that*

$$ua + vb = \alpha,$$

then (α, u, v) can be computed by the Euclidean algorithm.

The only case when no such triple (α, u, v) can be found is of course when a and b have a non-trivial gcrd in $\mathbb{S}$.

1.4. APPLICATIONS

Several non-trivial results can be obtained by the non-commutative Euclidean algorithm just described. After a simple application to the Jacobi polynomials, we show here how this algorithm can be used to get contiguity relations for hypergeometric series and we cast Sylvester's dialytic elimination in this framework.

1.4.1. JACOBI POLYNOMIALS

We apply the elimination of Corollary 1.2 on operators which annihilate the Jacobi polynomials. Starting from (1.11) and (1.12), we prove that Jacobi polynomials also satisfy (1.13) by eliminating the shift operator S_n between the polynomials G_1 and G_2 in the Ore algebra $\mathbb{Q}(a, b, n, x)[D_x; 1, D_x][S_n; S_n, 0]$.

The degrees of G_1 and G_2 in S_n are respectively 2 and 1. As a first step, we therefore multiply G_2 by S_n. Then we need to compute two polynomials α and β in the smaller Ore algebra $\mathbb{Q}(a, b, n, x)[D_x; 1, D_x]$ such that $\alpha G_1 + \beta S_n G_2$ has degree 1 in S_n. In general, this will be obtained by a recursive use of the algorithm. Here, since the leading coefficient of G_1 with respect to S_n does not depend on x or D_x, it is obviously sufficient to take the leading coefficient of G_1 for β and the leading coefficient of $-S_nG_2$ for α. Thus we get

$$\alpha G_1 + \beta S_n G_2 = AS_n + B,$$

where A and B are polynomials of degree 1 in D_x belonging to $\mathbb{Q}(a,b,n,x)[D_x;1,D_x]$. The next step consists in eliminating S_n between G_2 and $AS_n + B$. First, the same algorithm is applied recursively to compute polynomials α' and β' in $\mathbb{Q}(a,b,n,x)[D_x;1,D_x]$ such that $\alpha' A + \beta' g_2 = 0$, where g_2 is the leading coefficient of G_2 with respect to S_n. Then the polynomial

$$R = \alpha'(AS_n + B) + \beta' G_2 \tag{1.16}$$

does not involve S_n anymore.

These operations have been implemented in Chyzak's *Mgfun* package. Here follows the corresponding session. The first step is to load the package:

```
with(Mgfun):
```

Next, we create a suitable Ore algebra to accomodate both pairs of operators (n, S_n) and (x, D_x), with the commutation rules $S_n n = (n+1)S_n$ and $D_x x = xD_x + 1$:

```
A:=orealg(comm=[a,b],shift=[Sn,n],diff=[Dx,x]):
```

Using a philosophy reminiscent of Axiom's, an Ore algebra is represented internally as a table of procedures that perform its basic operations. Here `comm`, `diff` and `shift` are predefined types of Ore operators, but one could create Ore algebras with other operators.

Then we enter both polynomials corresponding to Eq. (1.11) and (1.12):

```
G:=[2*(n+2)*(n+a+b+2)*(2*n+a+b+2)*Sn^2
        -((2*n+a+b+3)*(a^2-b^2)+(2*n+a+b+2)
            *(2*n+a+b+3)*(2*n+a+b+4)*x)*Sn
        +2*(n+a+1)*(n+b+1)*(2*n+a+b+4),
    (2*n+a+b+2)*(1-x^2)*Dx*Sn-(n+1)
        *(a-b-(2*n+a+b+2)*x)*Sn-2*(n+a+1)*(n+b+1)]:
```

And we call the Euclidean algorithm to compute a non-zero polynomial free of S_n, if such a polynomial exists:

```
skewelim(G[1],G[2],Sn,A);
```

$$-an - bn - n - n^2 + axD_x + aD_x + bxD_x - bD_x + 2xD_x - D_x^2 + x^2D_x^2$$

This is the polynomial R in (1.16), which is precisely Eq. (1.13).

1.4.2. Gauss's hypergeometric function

Contiguity relations for hypergeometric series can also be computed by the non-commutative Euclidean algorithm (see (Takayama, 1989) for a generalization to multivariate hypergeometric functions based on Gröbner bases). We illustrate this computation on Gauss's hypergeometric function

$$F(a,b;c;z) = F\left(\begin{matrix} a,b \\ c \end{matrix} \middle| z\right) = \sum_{n=0}^{\infty} \frac{(a)_n(b)_n}{(c)_n n!} z^n,$$

where $(x)_n$ is the Pochhammer symbol $\Gamma(x+n)/\Gamma(x)$. The coefficient $u_{a,n}$ satisfies

$$\frac{u_{a,n+1}}{u_{a,n}} = \frac{(a+n)(b+n)}{(c+n)(n+1)}, \qquad \text{and} \qquad \frac{u_{a+1,n}}{u_{a,n}} = \frac{n}{a} + 1.$$

From the first identity, it is easy to see that the series F satisfies Gauss's hypergeometric equation (Erdélyi, 1981, p. 56), which is represented by the following differential linear operator

$$P = z(1-z)D_z^2 + (c-(a+b+1)z)D_z - ab. \tag{1.17}$$

From the second one follows the recurrence equation

$$F(a+1,b;c;z) = (H_a \cdot F)(a,b;c;z), \qquad \text{where} \qquad H_a = \frac{z}{a}D_z + 1. \tag{1.18}$$

The operator $H_a(D_z)$ is called a *step-up* operator.

From (1.17) and (1.18), we get the contiguity relation for F by the above skew Euclidean algorithm in the Ore algebra $\mathbb{Q}(a,b,c,z)[D_z;1,D_z][S_a;S_a,0]$. We first create the algebra

```
A:=orealg(comm=[b,c],diff=[Dz,z],shift=[Sa,a]):
```

Then we enter the operators

```
P:=z*(1-z)*Dz^2+(c-(a+b+1)*z)*Dz-a*b:
H:=a*Sa-(z*Dz+a):
```

And we compute the result of the elimination of D_z in this algebra:

```
skewelim(P,H,Dz,A);
```

$$aS_a^2 - 2aS_a + a + zaS_a - zaS_a^2 + zS_a - c - zbS_a + 1 + S_a^2 - zS_a^2 + S_a c - 2S_a$$

After some further cleaning up, this reads as follows:

$$(a+1)(1-z)F(a+2,b;c;z) + (c-zb+(z-2)(a+1))F(a+1,b;c;z) + (1+a-c)F(a,b;c;z) = 0,$$

which is the classical contiguity relation for the Gauss series (Erdélyi, 1981, p. 103).

More interestingly, the Euclidean algorithm can also be used to compute a *step-down* operator $B_a(D_z)$ such that

$$F(a-1,b;c;z) = (B_a \cdot F)(a,b;c;z)$$

from the knowledge of H_a. This is obtained by computing B_{a+1} as an inverse of H_a modulo the left ideal generated by P, or equivalently by computing L and B_{a+1} such that

$$LP + B_{a+1}H_a = 1,$$

which is exactly what the extended skew gcd algorithm does.

We begin as before by declaring a suitable Ore algebra, namely $\mathbb{Q}(a,b,c,z)[D_z;1,D_z]$:

```
A:=orealg(comm=[a,b,c],diff=[Dz,z]):
```

Then we use the operators P and $H_a = 1 + zD_z/a$ above and perform an extended gcd calculation (the current version of the code insists on being given the polynomial aH_a instead of H_a itself):

```
GCD:=skewgcdex(P,z*Dz+a,Dz,A):
```

The result is a list $[g,u,v,U,V]$ such that $g = uP + vaH_a$ and $0 = UP + VaH_a$. Then the result B_a is simply $(a-1)v(a-1)/g(a-1)$:

```
B:=collect((a-1)*subs(a=a-1,GCD[3]/GCD[1]),Dz,factor);
```

$$B := \frac{z(1-z)}{c-a}D_z + \frac{c-a-zb}{c-a}$$

Conversely, this type of computation is useful to get a step-up operator with respect to the parameter c, starting from the simpler step-down operator $B_c = 1 + zD_z/(c-1)$.

1.4.3. Partially hypergeometric series

The previous example can be generalized. This follows ideas from (Takayama, 1989, 1996) but avoids the use of Gröbner bases in a large class of summations.

We consider series of the form

$$f_n(z) = \sum_{k=0}^{\infty} u_{n,k} z^k, \tag{1.19}$$

where the sequence $u_{n,k}$ is *hypergeometric* with respect to n, i.e., $u_{n+1,k}/u_{n,k}$ is a rational function in $\mathbb{K}(n,k)$, for some field of coefficients $\mathbb{K}$. We also assume $u_{n,k}$ to satisfy a linear recurrence of the form

$$(L\cdot u)(n,k) = A_0(n,k)u_{n,k+p} + A_1(n,k)u_{n,k+p-1} + \cdots + A_p(n,k)u_{n,k} = 0,$$

where the A_i's belong to $\mathbb{K}(n,k)$.

Skew elimination in the Ore algebra $\mathbb{K}(n,z)[S_n;S_n,0][D_z;1,D_z]$ produces numerous results. First, it is well-known that from a linear operator like L above, one gets a linear operator $M \in \mathbb{K}(n,z)[D_z;1,D_z]$ vanishing at $f_n(z)$. We denote by K the degree of M in D_z.

Next, for any operator $H(n,S_n,S_n^{-1})$, the hypergeometric nature of $u_{n,k}$ with respect to n implies that $(H\cdot u)(n,k)/u_{n,k}$ is a rational function in $\mathbb{K}(n,k)$, i.e., there exist two polynomials P and Q in $\mathbb{K}[n,k]$ such that $Q(n,k)H - P(n,k)$ vanishes at $u_{n,k}$. It follows that $Q(n,zD_z)H - P(n,zD_z)$ vanishes at $f_n(z)$. Then eliminating D_z between this latter operator and M yields a linear dependency between $f, H\cdot f, H^2\cdot f, \ldots$ with coefficients in $\mathbb{K}(n,z)$. In particular, when $H = S_n$, this relation is a linear recurrence in n satisfied by $f_n(z)$, called a *contiguity relation*. Unfortunately, the relation obtained this way is not necessarily of the smallest possible order.

Following an idea of (Takayama, 1996), a smaller order contiguity relation can be obtained as follows. Using the Euclidean algorithm we first compute the inverse U of $Q(n, zD_z)$ modulo M, i.e., we compute U and V in $\mathbb{K}(n,z)[D_z;1,D_z]$ such that $UQ + VM = 1$. Then we divide $U(n,z,D_z)P(n,zD_z)$ by M in $\mathbb{K}(n,z)[D_z;1,D_z]$. If $R(n,z,D_z)$ is the remainder, we obtain that

$$(H \cdot f)(n,z) = [R(n,z,D_z) \cdot f](n,z),$$

where R has degree at most K in D_z. This relation relates a combination of $f_n(z)$ and its shifts to a combination of $f_n(z)$ and its derivatives. As before, an inverse of H could also be computed by the skew gcd algorithm, thus relating $(H^{-1} \cdot f)(n,z)$ to a combination of $f_n(z)$ and its derivatives.

In the case when $H = S_n$, which is of interest in the computation of a contiguity relation, a consequence of the above relation is

$$(S_n^p \cdot f)(n,z) = [R(n+p,z,D_z) \cdots R(n,z,D_z) \cdot f](n,z), \quad p \in \mathbb{N}. \tag{1.20}$$

Reducing the product in the right-hand side modulo M, we thus obtain a family of polynomials in D_z of degree at most K. Therefore by Gaussian elimination, we can obtain a linear dependency between $\{f, S_n \cdot f, \dots, S_n^K \cdot f\}$, which is the contiguity relation.

Example. The sum

$$f_n(z) = \sum_{k=0}^{\infty} \binom{n}{k}^2 \binom{n+k}{k}^2 (-1)^k z^k$$

can be treated by this algorithm. From the first order recurrence in S_k follows a fourth order differential equation M satisfied by $f_n(z)$. Direct elimination between this operator and that obtained from the first order recurrence in S_n yields a recurrence of order 7 satisfied by $f_n(z)$. However, the second method is guaranteed to yield a recurrence of order at most 4 (the degree of M in D_z). First, by recursive inversion of the coefficient of S_n modulo M, one obtains that modulo M,

$$\begin{aligned} S_n = 12\frac{z^3(z+1)}{(n+1)^3}D_z^3 + 4\frac{z^2(2n+2zn+11+14z)}{(n+1)^3}D_z^2 \\ - 4\frac{z(5zn^2-n^2-4n-6+2zn-9z)}{(n+1)^3}D_z - \frac{16zn-n-1+4z}{n+1}. \end{aligned}$$

This gives a relation between $f_{n+1}(z)$ and the derivatives of $f_n(z)$. Proceeding with further powers of S_n as in (1.20) and performing a Gaussian elimination eventually yields a fourth order recurrence satisfied by $f_n(z)$, whose coefficients are polynomials in n and z of degree at most 10.

This treatment also applies to q-analogues. We consider a sequence $(u_{n,k})$ assumed to be q-hypergeometric in n and which satisfies a linear recurrence in S_k in the Ore algebra $\mathbb{K}(q,q^n,q^k)[S_n;S_n,0][S_k;S_k,0]$. From this recurrence, it is again easy to derive an operator $M(q^n,z,H_{q,z})$ in the q-dilation operator $H_{q,z}$ (see Table 1) vanishing at the generating function f. For any operator $H(q^n,S_n,S_n^{-1})$, the rational function $(H \cdot u)/u$ has the form $P(q^n,q^k)/Q(q^n,q^k)$ for two polynomials P and Q in $\mathbb{K}[q,q^n,q^k]$. Then inverting Q modulo M in the Ore algebra $\mathbb{K}(q,q^n,z)[H_{q,z};H_{q,z},0]$ yields an operator R such that

$$H(q^n,S_n,S_n^{-1}) \cdot f = R(q^n,z,H_{q,z})P(q^n,H_{q,z}) \cdot f.$$

Proceeding as before when $H = S_n$ yields a linear recurrence satisfied by f.

1.4.4. Sylvester's dialytic elimination

Up to now, we have considered the application of the extended skew gcd algorithm in an Ore algebra $\mathbb{K}[\boldsymbol{\partial};\boldsymbol{\sigma},\boldsymbol{\delta}]$ or $\mathbb{K}[\mathbf{x}][\boldsymbol{\partial};\boldsymbol{\sigma},\boldsymbol{\delta}]$ to the elimination of one of the ∂_i's only. In the case when ∂_i commutes with $\mathbb{K}[\mathbf{x}\setminus x_i]$ and σ_i is an isomorphism, for instance when $\sigma_i(x_i)$ is a polynomial of degree 1 in x_i, the same algorithm applies to perform the elimination of x_i. This is obtained by rewriting the polynomials with ∂_i on the left of the monomials, which preserves the degrees in x_i and in ∂_i. Then the computation performed by the extended Euclidean algorithm and that of Sylvester's dialytic elimination (Zeilberger, 1990b) are equivalent, in the same way that a resultant computation is equivalent to computing the determinant of Sylvester's matrix in the classical commutative case.

Example. The identity (0.9) gives a summatory representation of the Gegenbauer polynomial. It can be proved in the polynomial Ore algebra $\mathbb{Q}(\alpha,x,n)[m][S_n;S_n,0][S_m;S_m,0]$. The elimination of m by the extended skew gcd algorithm between the operator specification (0.7) of the summand

$$a_{n,m} = (-1)^m \frac{\Gamma(\alpha+n-m)}{m!(n-2m)!}(2x)^{n-2m}$$

yields the operator

$$\begin{aligned}(S_m-1)&(2x(n+1+\alpha)S_n-(n+2)S_n^2)\\ &-((n+2)S_n^2-2x(n+1+\alpha)S_n+n+2\alpha).\end{aligned}$$

By construction, applying this operator on $a_{n,m}$ yields 0. It follows that

$$\left[\left((n+2)S_n^2-2x(n+1+\alpha)S_n+n+2\alpha\right)\cdot a\right](n,m) = b_{n,m+1}-b_{n,m},$$

where

$$b_{n,m} = \left[\left(2x(n+1+\alpha)S_n-(n+2)S_n^2\right)\cdot a\right](n,m).$$

Summation over m then proves Eq. (0.9).

1.4.5. Skew fractions

Another important application of the Euclidean algorithm is the construction of the field of fractions of a skew polynomial ring (Ore, 1933). Calculations with these fractions are not needed in this work although they are used implicitly when the effective left Ore ring is of the form $\mathbb{A} = \mathbb{K}[\boldsymbol{\partial};\boldsymbol{\sigma},\boldsymbol{\delta}]$ (i.e., a skew polynomial ring in several ∂'s).

1.5. Gröbner bases in Ore algebras

In the examples above, we use the Euclidean division in several ways. First, as a provider of *normal forms* by taking remainders of Euclidean divisions by the operator generating the ideal we are working with. Next, as an *elimination process* by the Euclidean algorithm. In commutative algebra, a generalization of Euclidean division to the multivariate case allowing the same computations is provided by *Gröbner* *bases*. A Gröbner basis is a system of generators of an ideal satisfying particular properties (see below), so that a reduction process analogous to the Euclidean division makes it possible to test ideal membership and to compute normal forms for elements of the residue class ring of the ideal. Besides, for special *term orders*, the computation of Gröbner bases makes it possible to eliminate variables (see (Cox *et al.*, 1992; Becker and Weispfenning, 1993) for tutorial introductions and (Buchberger, 1965, 1970, 1985) for the original articles on Buchberger's algorithm). In this section, we introduce non-commutative Gröbner bases for Ore algebras. The main result is Theorem 1.2, which gives sufficient conditions under which Gröbner bases can be computed in Ore algebras by a modification of Buchberger's algorithm. Early work in this area in the context of Weyl algebras is due to Galligo (1985). Takayama (1989) used an analogous technique for difference-differential algebras. While Gröbner bases are classical in a commutative context, the theory of Gröbner bases in non-commutative algebras is less well-known. We refer the reader to (Mora, 1994) for a survey.

In the non-commutative case, one distinguishes the one-sided ideals (left or right) and the two-sided ones. An algebra is *noetherian* when it does not contain any infinite strictly increasing chain of two-sided ideals; it is left-noetherian when it does not contain any such chain of left ideals; right-noetherianity is defined similarly. Left-noetherianity is a convenient condition for Gröbner bases of left ideals to be finite. Unfortunately, some Ore algebras are not left-noetherian. An example is given in (Weispfenning, 1992), with the polynomial Ore algebra $\mathbb{Q}[x][M; M, 0]$, where M is the Mahlerian operator with commutation rule $Mx = x^p M$ for an integer $p > 1$ (see Table 2). Let $\mathfrak{I}_n$ be the left ideal generated by $(x, xM, \dots, xM^n)$. Then $xM^{n+1} \notin \mathfrak{I}_n$, and $(\mathfrak{I}_n)_{n\in\mathbb{N}}$ is an infinite strictly increasing sequence of left ideals. Therefore, not all left ideals have a finite basis. (Surprisingly, this implies that Proposition 8.2 p. 35 in (Cohn, 1971) is wrong.)

The case of *polynomial rings of solvable type* studied by Kandri-Rody and Weispfenning (1990) is intermediate between the non-commutative case and the commutative one. Such a ring is defined as a ring $\mathbb{K}\langle x_1, \dots, x_n\rangle$ ruled by commutations $x_j x_i = c_{i,j} x_i x_j + p_{i,j}$ for $i < j$, with non-zero $c_{i,j} \in \mathbb{K}$ and polynomials $p_{i,j}$ of the ring smaller than $x_i x_j$ with respect to a fixed term order. These rings are left and right noetherian and hence noetherian. Even then, the finiteness of Gröbner bases and the termination of Buchberger's algorithm depend on the term order with respect to which the Gröbner basis is defined. In short, the termination of Buchberger's algorithm in polynomial rings of solvable type is guaranteed for all term orders $\preceq$ such that $p_{i,j} \prec x_i x_j$ for all i and j. To accomodate Ore algebras, we need a slightly more general framework, where the variables x_i have a commutation rule with elements a in the ground field $\mathbb{K}$ of the form $x_i a = \sigma_i(a) x_i + \delta_i(a)$. It is possible to extend slightly the ideas in (Kandri-Rody and Weispfenning, 1990) to this context and this was done by (Kredel, 1993).

As in the commutative case, Gröbner bases are defined with respect to admissible term orders, which makes it possible to generalize the leading term used in the Euclidean division. This is obtained by considering the set T of *terms* in the algebra. Each algebra for which we define Gröbner bases is canonically isomorphic as a $\mathbb{K}$-vector space to a polynomial ring $\mathbb{K}[\mathbf{u}]$ for a commutative tuple of indeterminates $\mathbf{u}$. The set of *terms* is the commutative monoid T generated by the u_i's.

Definition 1.3. *An* admissible term order *on the set T of terms is a total order $\preceq$ with 1 as least element and which is compatible with the product, i.e., such that $tu \preceq tv$ for all $t \in T$ whenever $u \preceq v$ for $u, v \in T$.*

From now on, the term orderings we consider are admissible.

The definition of Gröbner bases is in terms of *reductions*. A polynomial p involving a term s (with a non-zero coefficient) is *reducible* by a polynomial q of leading term t (with respect to a fixed term order) whenever t divides s. In this case, we write $p \underset{q}{\rightarrow} p'$ for $p' = p - at'q$, with t' such that $s = t't$ and a scalar a such that s appears with coefficient zero in p'. Similarly, we write $p \underset{Q}{\rightarrow} p'$ for a system Q of polynomials whenever $p \underset{q}{\rightarrow} p'$ for any $q \in Q$. We finally write $p \overset{+}{\underset{Q}{\rightarrow}} p'$ when there is a finite sequence of reductions leading from p to p' (including the case of no reduction, $p = p'$).

The following theorem defines Gröbner bases by equivalent properties they satisfy, gives a sufficient condition for an Ore algebra to possess finite Gröbner bases, and for these bases to be computed by a non-commutative analogue of Buchberger's algorithm.

THEOREM 1.2. (KREDEL) *Let $\mathbb{O} = \mathbb{K}[\mathbf{x}][\boldsymbol{\partial};\boldsymbol{\sigma},\boldsymbol{\delta}]$ be a polynomial Ore algebra such that $\boldsymbol{\partial}$, $\boldsymbol{\sigma}$, $\boldsymbol{\delta}$ and $\mathbf{x}$ satisfy relations of the type*

$$\partial_i x_j = (a_{i,j} x_j + b_{i,j})\partial_i + c_{i,j}(\mathbf{x}), \qquad 1 \le i \le r, \quad 1 \le j \le s,$$

with $b_{i,j} \in \mathbb{K}$, $a_{i,j} \in \mathbb{K} \setminus \{0\}$, and $c_{i,j} \in \mathbb{K}[\mathbf{x}]$.

Let $\mathfrak{I}$ be a left ideal of $\mathbb{O}$, with basis G. Then, the following properties are equivalent:

1. *for all $f, f_1, f_2 \in \mathbb{O}$, if $f \overset{+}{\underset{G}{\rightarrow}} f_1$ and $f \overset{+}{\underset{G}{\rightarrow}} f_2$, there exists $f' \in \mathbb{O}$ such that $f_1 \overset{+}{\underset{G}{\rightarrow}} f'$ and $f_2 \overset{+}{\underset{G}{\rightarrow}} f'$;*
2. *for all $f, g \in \mathbb{O}$ with $f - g \in \mathfrak{I}$, there exists $h \in \mathbb{O}$ such that $f \overset{+}{\underset{G}{\rightarrow}} h$ and $g \overset{+}{\underset{G}{\rightarrow}} h$;*
3. *for all $f \in \mathfrak{I}$, $f \overset{+}{\underset{G}{\rightarrow}} 0$;*
4. *for all non-zero $f \in \mathfrak{I}$, f is reducible modulo G;*
5. *for all non-zero $f \in \mathfrak{I}$, there exists $g \in G$ such that the leading term of g divides the leading term of f.*

When a basis G of an ideal satisfies these properties, it is called a (left) Gröbner basis*.*

Moreover, $\mathbb{O}$ is left Noetherian and a non-commutative version of Buchberger's algorithm terminates for term orders with respect to which all the ∂_i are larger than the x_i's. When additionally all the $c_{i,j}$'s are of total degree at most 1 in the x_i's, Buchberger's algorithm terminates for any term order on $\mathbf{x}$ and $\boldsymbol{\partial}$. In all cases of termination, Buchberger's algorithm computes a Gröbner basis with respect to the term order.

PROOF. The main part of this theorem is the case treated by Kandri-Rody and Weispfenning (1990). The extension to the commutation rules obeyed by the polynomials in Ore algebras is due to (Kredel, 1993). □

When this theorem applies, efficiency can be improved by suitable generalizations of the so-called "normal strategy" (Cox *et al.*, 1992, Chap. 2), "sugar strategy" (Giovini *et al.*, 1991), by "trace lifting" (Takayama, 1995) and by generalizations of Buchberger's criteria (Kredel, 1993, Chap. 4). Further discussion of implementation and efficiency will be part of (Chyzak, 1998)—see also (Chyzak, 1994).

As can be seen from Table 2, this theorem applies to many useful Ore algebras. The only exception mentioned in this table is the algebra $\mathbb{K}[x][M;M,0]$ for a Mahlerian operator M. (Compare to $\mathbb{K}(x)[M;M,0]$, which is Euclidean and hence Noetherian.)

The special case $s=0$ of this theorem states that in non-polynomial Ore algebras, Gröbner bases for *any* order can be computed by a non-commutative version of Buchberger's algorithm.

Example. All the examples of §1.4 can also be treated by computation of Gröbner bases. For instance, we perform the same computation as before on the Jacobi polynomials. This is achieved by defining a lexicographic order on the variables, with $D_x \prec S_n$:

```
T:=termorder(A,plex=[Sn,Dx]):
```

Next, a Gröbner basis with respect to this order is computed. It contains two polynomials: the mixed difference-differential operator (1.12) and the differential operator (1.13). We display the latter by selecting only those terms without S_n (the call to `gbasis` computes the Gröbner basis; the call to `remove` performs the selection):

```
remove(has,gbasis(G,T),Sn);
```

$$\left[-n^2-n-na-nb+x^2D_x^2+2xD_x+bxD_x+acD_x+aD_x-bD_x-D_x^2\right]$$

Similarly, one could obtain (1.11) by eliminating D_x between the Ore polynomials (1.12) and (1.13).

Example. As an example of an Ore algebra for which not all term orders are allowed by Theorem 1.2, we deal with the case of $\mathbb{O}=\mathbb{K}[x,u][D_x;1,D_x]$, defined as an extension of the Weyl algebra $\mathbb{K}[x][D_x;1,D_x]$ by the additional commutation rule

$$D_x u = uD_x - u^2. \tag{1.21}$$

This algebra appears when one tries to localize the Weyl algebra $\mathbb{K}[x][D_x;1,D_x]$, extending it with an inverse for x: the skew polynomial ring $\mathbb{S}=\mathbb{K}[x,x^{-1}][D_x;1,D_x]$, with D_x defined on the ring $\mathbb{K}[x,x^{-1}]$ as the usual differentiation with respect to x, is not an Ore algebra for which Theorem 1.2 is applicable; however, Eq. (1.21) formally looks like the commutation $D_x x^{-1} = x^{-1}D_x - x^{-2}$ in $\mathbb{S}$.

Theorem 1.2 applies to $\mathbb{O}$, and Buchberger's algorithm provides us with Gröbner bases, but only for term orders such that $u \prec D_x$ (in this case, $u^2 \prec uD_x$). It is therefore not possible to eliminate u from an ideal in $\mathbb{O}$ by simply computing a Gröbner basis. (This elimination can be performed in another way (see §4.3 and (Chyzak, 1998)).

Here is how Buchberger's algorithm with a term order such that $D_x \prec u$ fails to terminate. Let $p=uD_x$ and $G=\{D_x-u\}$. Then the reduction of p by G yields an infinite sequence of successive polynomials p_n, with

$$p_{2n}=uD_x+nD_x^2, \qquad p_{2n+1}=u^2+(n+1)D_x^2 \qquad (n\in\mathbb{N}).$$

2. Ore algebras and ∂-finiteness

Solutions of linear recurrence or differential equations with polynomial coefficients are of particular interest to computer algebra and combinatorics, since they can be specified by a finite amount of information: the coefficients and a finite number of initial conditions.

This has led Zeilberger to generalize the notions of P-recursive sequences and D-finite functions studied by Stanley (1980) into a notion of P-finiteness (Zeilberger, 1990b). In several variables, a function is P-finite when the vector space generated by its derivatives has finite dimension over the field of rational functions. Similarly, a sequence is P-finite when the vector space generated by its shifts has finite dimension over the field of rational functions. This has a simple translation in terms of ideals, and this translation yields a very natural generalization in the context of Ore algebras.

Definition 2.1. *Let $\mathbb{O} = \mathbb{K}[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$ be an Ore algebra over a field $\mathbb{K}$. A left ideal $\mathfrak{I}$ of $\mathbb{O}$ is* ∂-finite *if $\mathbb{O}/\mathfrak{I}$ is finite-dimensional over $\mathbb{K}$.*

The "∂" in this definition is merely a symbol and has no relation with the actual ∂_i's in the algebra. Functions, series, distributions, sequences, etc which are annihilated by such an ideal will also be called ∂-finite. Example of ∂-finite functions and sequences are the Jacobi polynomials already discussed, or the functions appearing in (0.1) and (0.6).

When $\mathfrak{I}$ is the annihilating ideal $\operatorname{Ann} f$ of a function f, the quotient $\mathbb{O}/\operatorname{Ann} f$ is isomorphic to the $\mathbb{O}$-module $\mathbb{O} \cdot f$ and this quotient is finite-dimensional if and only if the successive pseudo-derivatives $\partial_1^{\alpha_1} \cdots \partial_r^{\alpha_r} \cdot f$ of f span a finite-dimensional vector space over $\mathbb{K}$.

Reciprocally, when $\mathfrak{I}$ is a ∂-finite ideal, then $\mathbb{O}/\mathfrak{I}$ is isomorphic to the module $\mathbb{O} \cdot f$ where f is the residue class of 1 in $\mathbb{O}/\mathfrak{I}$. This f corresponds to a generic function annihilated by $\mathfrak{I}$. Thus, ∂-finite ideals make it possible to express the properties and algorithms below without any reference to a specific algebra of functions. For any g annihilated by $\mathfrak{I}$, $\operatorname{Ann} g \supset \operatorname{Ann} f = \mathfrak{I}$. For instance, if $\mathfrak{I}$ is generated by $D_x^2 + 1$ in $\mathbb{O} = \mathbb{C}(x)[D_x; 1, D_x]$, then $\mathbb{O} \cdot f$ is isomorphic to either $\mathbb{O} \cdot \cos(x)$ or $\mathbb{O} \cdot \sin(x)$. Besides, $g = (-iD_x + 1) \cdot f$ corresponds to $e^{\pm ix}$; it is annihilated by $\mathfrak{I}$ and $\mathbb{O} \cdot g$ is a strict submodule of $\mathbb{O} \cdot f$.

The study of ∂-finite ideals is motivated by their nice closure properties and the relative simplicity of the corresponding algorithms. In the definition of Ore algebras, we demand that the σ_i's and δ_j's commute pairwise (except possibly when $i = j$). This constraint could be relaxed, leaving the closure properties of ∂-finite functions unchanged. However, this assumption becomes crucial when we want to compute an annihilating system for a sum or a product, and in particular when using Gröbner bases to do so.

2.1. Rectangular systems

To simplify the proofs, we first note that ∂-finite ideals contain systems of polynomials of a special shape which we call *rectangular.*

Definition 2.2. *A system of polynomials of an Ore algebra $\mathbb{K}[\partial_1; \sigma_1, \delta_1] \ldots [\partial_r; \sigma_r, \delta_r]$ is* rectangular *when it consists of r non-zero univariate polynomials $P_i(\partial_i)$, $i = 1, \ldots, r$.*

There is no loss of generality in considering systems of this special form, as follows from the next proposition.

Proposition 2.1. *An ideal of an Ore algebra $\mathbb{K}[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$ is ∂-finite if and only if it contains a rectangular system.*

Proof. Let $\mathbb{O} = \mathbb{K}[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$. If $\mathfrak{I}$ is a ∂-finite ideal, then for each i, $\left\{1, \partial_i, \partial_i^2, \ldots\right\}$ spans a finite-dimensional vector space over $\mathbb{K}$ in $\mathbb{O}/\mathfrak{I}$, from which follows the existence of a polynomial in ∂_i with coefficients in $\mathbb{K}$ which becomes zero in the quotient (i.e., belongs to the ideal). Conversely, if $\mathfrak{I}$ contains a rectangular system with k_i the degree of the polynomial in ∂_i, then $\mathbb{O}/\mathfrak{I}$ is generated by $\{\partial_1^{p_1}\cdots\partial_n^{p_n}\}_{0\le p_i<k_i}$ as a $\mathbb{K}$-vector space. □

A consequence of this proposition is that proving the ∂-finiteness of a "function" in an Ore algebra $\mathbb{O}$ reduces to proving that it is annihilated by a rectangular system of operators in $\mathbb{O}$. As an example of application, an important subclass of ∂-finite functions is often provided by *rational functions*.

Proposition 2.2. *Let $\mathbb{O} = \mathbb{K}[\boldsymbol{\partial};\boldsymbol{\sigma},\boldsymbol{\delta}]$ be an Ore algebra which acts on an algebra of functions $\mathcal{F}\supseteq\mathbb{K}$, making it an $\mathbb{O}$-module. If for all operators ∂_i of $\mathbb{O}$, the function $\gamma_i = \partial_i\cdot 1$ is in $\mathbb{K}$ then all functions of $\mathbb{K}$ are ∂-finite with respect to $\mathbb{O}$.*

Proof. Let r be any function of $\mathbb{K}$. Then $\partial_i\cdot r = \sigma_i(r)\gamma_i + \delta_i(r)$ is a function $f_i\in\mathbb{K}$. Thus $\partial_i - f_i r^{-1}$ is an operator which annihilates r. □

Another very simple example of ∂-finite "functions" is provided by hypergeometric sequences, i.e., sequences $u_{n_1,\dots,n_p}$ such that $u_{n_1,\dots,n_i+1,\dots,n_p}/u_{n_1,\dots,n_p}$ is rational (i.e., belongs to $\mathbb{K}$) for all i. The corresponding rectangular system consists only of shift operators of order one.

By Theorem 1.2 computations of Gröbner bases always terminate in non-polynomial Ore algebras. It is possible to compute a rectangular system included in a ∂-finite ideal $\mathfrak{I}$ from a Gröbner basis of $\mathfrak{I}$ (for any order) as follows. For each ∂_i in the algebra, ∂_i^k is reduced modulo this basis for $k = 0, 1, \dots$ This reduction rewrites the ∂_i^k in terms of a finite number of monomials $\partial_1^{i_1}\cdots\partial_r^{i_r}$ independent of k. The algorithm stops when a linear dependency between the remainders is detected by Gaussian elimination. Note however that in general, the ideal generated by this rectangular system is smaller than the original ideal. This may lead to calculations where the final equations have larger order than the minimal one, since inclusion is reversed on the corresponding modules.

2.2. Closure properties

Given two ∂-finite "functions" f and g (or equivalently two ∂-finite ideals $\mathfrak{I}$ and $\mathfrak{K}$ of an Ore algebra $\mathbb{O}$ and generators f and g of the $\mathbb{O}$-modules $\mathbb{O}/\mathfrak{I}$ and $\mathbb{O}/\mathfrak{K}$), we show in this section that $f+g$ is also ∂-finite, we determine sufficient conditions for fg to be ∂-finite and we show how to perform computations of expressions involving specializations of f and pseudo-derivatives of f.

In each case, the problem is first translated into the language of ideals and modules, then conditions on the Ore algebra for the resulting ideal to exist are derived. This is then made effective by providing algorithms which construct generators of the ideal under consideration. For each operation, we give two different algorithms. One inputs and outputs rectangular systems and relies on skew Euclidean division. The other one is based on Gröbner bases and returns generators of an ideal which is generally larger (hence better) than that produced from the rectangular systems. Our versions of the algorithms based on rectangular systems are natural extensions of both algorithms in (Takayama, 1992, Sec. 3) for the differential case. This generalization is straightforward in the case of the sum, while a restriction on the Ore algebras under consideration is needed in the case of the product.

2.2.1. Closure under addition

Lemma 2.1. *Let $\mathfrak{I}$ and $\mathfrak{K}$ be two ∂-finite ideals in an Ore algebra $\mathbb{O}$. The annihilating ideal for any sum $f+g$ where f is annihilated by $\mathfrak{I}$ and g is annihilated by $\mathfrak{K}$ is also ∂-finite.*

Proof. An operator $P \in \mathbb{O}$ is applied to $f+g$ by $P \cdot (f+g) = (P \cdot f) + (P \cdot g)$. The first summand can be reduced modulo $\mathfrak{I} = \operatorname{Ann} f$, while the second summand can be reduced modulo $\mathfrak{K} = \operatorname{Ann} g$. Thus the natural algebraic setting is the direct sum $\mathbb{T} = \mathbb{O}/\mathfrak{I} \oplus \mathbb{O}/\mathfrak{K} \simeq \mathbb{O} \cdot f \oplus \mathbb{O} \cdot g$ (over $\mathbb{K}$), which is of finite dimension, since both ideals are ∂-finite. □

A rectangular system for the sum can be computed using rectangular systems for $\operatorname{Ann} f$ and $\operatorname{Ann} g$. For each ∂ in the algebra, one reduces $\partial^k \cdot f$ and $\partial^k \cdot g$ for $k = 1, 2, 3, \ldots$ in the sequence $f+g, \partial \cdot f + \partial \cdot g, \partial^2 \cdot f + \partial^2 \cdot g, \ldots$ This eventually yields a rectangular system for $f+g$ by Gaussian elimination.

The ∂-finite ideal obtained in this way is not necessarily as large as possible: a rectangular system does not take possible mixed relations into account. An example is provided by the sum of Bessel functions $f = \sum_{i=1}^{n} J_{\mu_i}(xy)$ in the Ore algebra $\mathbb{O} = \mathbb{C}(x,y)[D_x; 1, D_x][D_y; 1, D_y]$. In this case, it is easily seen that a rectangular system for f is constituted of two differential polynomials of order 2^n. Therefore, the corresponding vector space is of dimension $(2^n)^2$. Noting that $xD_x - yD_y$ vanishes at f, we get that the dimension of $\mathbb{O} \cdot f$ is only 2^n. The complexity of further calculations with f is then dramatically different, depending on which description is used. This phenomenon shows the need for procedures to compute mixed relations. Two procedures are available.

If Gröbner bases are given for both $\operatorname{Ann} f$ and $\operatorname{Ann} g$, then a Gröbner basis of a subideal of the annihilating ideal of $f+g$ can be computed by noting that $\operatorname{Ann} f \cap \operatorname{Ann} g \subseteq \operatorname{Ann}(f+g)$. Thus, as in the commutative case, a basis for this ideal is obtained by eliminating a new commutative variable t in $t \operatorname{Ann} f + (1-t) \operatorname{Ann} g$. In the univariate case, this algorithm reduces to computing a lclm, for instance by the extended skew gcd algorithm. In the multivariate case, when the input Gröbner bases contain mixed polynomials, the output naturally takes this information into account.

Another procedure, which will also apply to other operations, consists in applying an extension of the fglm algorithm (Faugère *et al.*, 1993), and gives a basis of $\operatorname{Ann}(f \oplus g)$, which contains $\operatorname{Ann} f \cap \operatorname{Ann} g$ and is included in $\operatorname{Ann}(f+g)$.

2.2.2. The fglm algorithm

This algorithm was designed to compute *zero-dimensional* Gröbner bases by a change of ordering. It relies on the observation that given a term order and a zero-dimensional ideal $\mathfrak{I}$ of an algebra $\mathbb{A}$, a finite basis of $\mathbb{A}/\mathfrak{I}$ as a vector space is given by those terms smaller than all the leading terms of the polynomials in the Gröbner basis for that order. From a known Gröbner basis for any order, reduction yields the coordinates of elements of $\mathbb{A}/\mathfrak{I}$ in a finite-dimensional vector space (this is the *NormalForm* function in (Faugère *et al.*, 1993)). If $\preceq$ denotes the term order with respect to which one wants to compute a new Gröbner basis, the algorithm constructs a basis B of the vector space $\mathbb{A}/\mathfrak{I}$ and a set M of terms outside B, which are known to be expressible as a linear combination of elements of B. As long as there exists a term outside B which is not a multiple of any of the elements of M, the algorithm considers the term t which is the least of such terms with respect to $\preceq$ and computes its normal form by *NormalForm*. Then either there is a linear dependency between this normal form and the normal forms of the elements of B, in which case t is added to M, or there is none, in which case t is added to B. The loop terminates when sufficiently many terms have been introduced into B. This happens because $\mathbb{A}/\mathfrak{I}$ is finite-dimensional which is implied by the zero dimensionality of $\mathfrak{I}$. At the end, each term $m \in M$ is expressible as a linear combination p_m of elements of B, and the system of $m - p_m$ is a Gröbner basis for $\mathfrak{I}$ with respect to $\preceq$. This is the *NewBasis* algorithm of (Faugère *et al.*, 1993).

In our context, we use the same *NewBasis* algorithm, defining the *NormalForm* function in terms of the Gröbner bases for $\operatorname{Ann} f$ and $\operatorname{Ann} g$. The normal form of a term $t = \partial_1^{\alpha_1} \cdots \partial_n^{\alpha_n}$ is defined (and computed) as the pair $(t_1, t_2) \in \mathbb{O}/\mathfrak{I} \oplus \mathbb{O}/\mathfrak{K}$, where t_1 and t_2 are the normal forms of t with respect to each Gröbner bases available for $\operatorname{Ann} f$ and $\operatorname{Ann} g$ respectively. Thus this function inputs a term, and returns a normal form in a finite-dimensional vector space, which is the property required for *NewBasis* to terminate.

Example. We compute annihilators for the sum of the exponential function $f(x,y) = \exp(\mu x + \nu y)$ and of the product of Bessel functions $g(x,y) = J_\mu(x) J_\nu(y)$.

The functions f and g are specified by the rectangular systems

$$\begin{cases} f_x - \mu f = 0, \\ f_y - \nu f = 0 \end{cases} \qquad \text{and} \qquad \begin{cases} x^2 g_{x,x} + x g_x + (x^2 - \mu^2) g = 0, \\ y^2 g_{y,y} + y g_y + (y^2 - \nu^2) g = 0 \end{cases}$$

respectively (indices denoting differentiation). Using rectangular systems, one gets two Ore polynomials of order 3 for the sum, namely

$$\begin{aligned} p_1 = {} & x^2(x^2 - \mu^2 + \mu^2 x^2 + \mu x) D_x^3 - x(3\mu^2 - \mu^3 x + \mu^3 x^3 + \mu x^3 - 2\mu x - x^2) D_x^2 \\ & + (x^4 - x^2 + \mu^2 x^4 - \mu^2 - \mu^4 x^2 - \mu^3 x^3 + \mu^4 - 4\mu^2 x^2) D_x \\ & + \mu(\mu^2 + 3\mu^3 x - \mu^2 x^4 - \mu^4 - \mu x^3 - x^4 + \mu^4 x^2 + x^2 + 2\mu^2 x^2) \end{aligned}$$

and a similar polynomial (p_3 below) where the rôles of (μ, x, D_x) and (ν, y, D_y) have been exchanged.

We now show the use of the FGLM algorithm to compute a Gröbner basis of Ore polynomials vanishing on the sum $s(x,y)$. First, the algorithm reduces 1, D_y, D_x, D_y^2, $D_x D_y$ and detects that they are independent. Then D_x^2 is reduced and found to satisfy a linear relation with the previous ones, expressed by the following Ore polynomial:

$$\begin{aligned} p_2 = {} & -(x^2 - \mu^2 + x^2\mu^2 + \mu x) y^2 D_y^2 + x^2(y^2 - \nu^2 + y^2\nu^2 + \nu y) D_x^2 \\ & -(x^2 - \mu^2 + x^2\mu^2 + \mu x) y D_y + x(y^2 - \nu^2 + y^2\nu^2 + \nu y) D_x \\ & -\mu^2 y^2 \nu^2 + x^2 \nu y + x^2 y^2 \nu^2 - x^2 \mu^2 y^2 + x^2 \mu^2 \nu^2 - \mu x y^2 + \mu x \nu^2 - \mu^2 \nu y. \end{aligned}$$

Next, the algorithm continues by reducing D_y^3 and finds a new relation

$$\begin{aligned} p_3 = {} & y^2(y^2 - \nu^2 + \nu y + y^2\nu^2) D_y^3 - y(y^3\nu + y^3\nu^3 - y^2 - 2\nu y - \nu^3 y + 3\nu^2) D_y^2 \\ & + (y^4 + y^4\nu^2 - y^3\nu^3 - y^2 - y^2\nu^4 - 4y^2\nu^2 - \nu^2 + \nu^4) D_y \\ & + \nu(-y^4 - y^4\nu^2 + y^2 - \nu^4 + 2y^2\nu^2 + \nu^2 + y^2\nu^4 - y^3\nu + 3\nu^3 y). \end{aligned}$$

Finally, the reduction of $D_x D_y^2$ produces the Ore polynomial

$$p_4 = y^2 D_x D_y^2 - \mu y^2 D_y^2 + y D_x D_y - \mu y D_y + (y^2 - \nu^2) D_x - \mu(y^2 - \nu^2).$$

Thus, the computation with the FGLM algorithm returns more information than a simple rectangular system. On this example, the rectangular system $\{p_1, p_3\}$ makes it possible to rewrite any derivative of $s(x, y)$ as a linear combination of 9 derivatives, while the more accurate output $\{p_2, p_3, p_4\}$ of the FGLM algorithm yields a basis of 5 derivatives only.

The algorithm based on the intersection of ideals introduces a new commutative indeterminate t and starts from the following system in $\mathbb{K}(x, y)[t][D_x; 1, D_x][D_y; 1, D_y]$:

$$\{t(D_x - \mu), t(D_y - \nu), (1-t)(x^2 D_x^2 + x D_x + x^2 - \mu^2), (1-t)(y^2 D_y^2 + y D_y + y^2 - \nu^2)\}.$$

Eliminating the variable t yields the same basis $\{p_2, p_3, p_4\}$ as above.

2.2.3. Closure under product

In order to deal with the product, we need more information on σ_i and δ_i in (1.14). In case of an Ore operator ∂ and functions f and g, this relation implies

$$\begin{gathered}
\partial \cdot (fg) = (\sigma \cdot f)(\partial \cdot g) + (\delta \cdot f) g, \\
\partial^2 \cdot (fg) = (\sigma^2 \cdot f)(\partial^2 \cdot g) + 2(\sigma\delta \cdot f)(\partial \cdot g) + (\delta^2 \cdot f) g, \\
\partial^3 \cdot (fg) = (\sigma^3 \cdot f)(\partial^3 \cdot g) + 3(\sigma^2\delta \cdot f)(\partial^2 \cdot g) + 3(\sigma\delta^2 \cdot f)(\partial \cdot g) + (\delta^3 \cdot f) g, \\
\dots,
\end{gathered}$$

where we have assumed commutativity between σ and δ and denoted composition of those operators by a product. While g appears only in the $(\partial^i \cdot g)$'s in the successive pseudo-derivatives of fg, infinitely many new $\sigma^p \delta^q \cdot f$ are produced. In order to make use of potential ∂-finiteness of f, we need to relate those symbols to the successive pseudo-derivatives of f. The following sufficient condition is therefore natural.

LEMMA 2.2. *Let $\mathbb{O} = \mathbb{K}[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$ be an Ore algebra and $\mathfrak{I}$ and $\mathfrak{K}$ be two ∂-finite ideals of $\mathbb{O}$. Assume that for all $i \in \{1, \dots, r\}$ there are polynomials $A_i(u)$ and $B_i(u)$ over $\mathbb{K}$ such that $\sigma_i = A_i(\partial_i)$ and $\delta_i = B_i(\partial_i)$, where the products denote compositions. Then the annihilating ideal for any product fg where f is annihilated by $\mathfrak{I}$ and g is annihilated by $\mathfrak{K}$ is also ∂-finite.*

The hypothesis on the ∂_i's is satisfied by all the examples of Table 2.

Again, f and g in this lemma need not be interpreted as functions but as generators of the $\mathbb{O}$-modules $\mathbb{O}/\mathfrak{I}$ and $\mathbb{O}/\mathfrak{K}$.

PROOF. Let $\sigma_i = A_i(\partial_i)$ and $\delta_i = B_i(\partial_i)$ for $i \in \{1, \dots, r\}$ be as above. Instead of considering sums of the form $P \cdot f + Q \cdot g$, we need to consider linear combinations of monomials of the form $(P \cdot f)(Q \cdot g)$. The natural setting for this computation is the tensor product $\mathbb{T} = \mathbb{O}/\mathfrak{I} \otimes \mathbb{O}/\mathfrak{K} \simeq \mathbb{O} \cdot f \otimes \mathbb{O} \cdot g$ (over $\mathbb{K}$). The application of ∂_i to products of the above type,

$$\partial_i \cdot (P \cdot f)(Q \cdot g) = (\sigma_i \cdot (P \cdot f))(\partial_i \cdot (Q \cdot g)) + (\delta_i \cdot (P \cdot f))(Q \cdot g),$$

is translated into the following action which reflects (1.14):

$$\partial_i(P \otimes Q) = (A_i(\partial_i)P) \otimes (\partial_i Q) + (B_i(\partial_i)P) \otimes Q.$$

Computing an operator which annihilates the product fg reduces to computing a polynomial which annihilates $1 \otimes 1$. Such a polynomial exists since $\mathbb{T}$ is finite dimensional. □

The algorithm to get a rectangular system which annihilates the product works as for the sum above by expressing the $\partial^k \cdot (fg)$, $k = 1, 2, \dots$ in the finite basis $(\partial^i \cdot f) \otimes (\partial^j \cdot g)$ and using Gaussian elimination to get an operator for each ∂ in the algebra. Once again, if Gröbner bases are given for $\operatorname{Ann} f$ and $\operatorname{Ann} g$ then a Gröbner basis of the (generally larger) annihilating ideal for $f \otimes g$ is obtained by the extension of the FGLM algorithm described above. The ideal $\operatorname{Ann}(f \otimes g)$ is a ∂-finite subideal of $\operatorname{Ann}(fg)$.

2.2.4. Closure under the action of Ore operators

LEMMA 2.3. *Let $\mathfrak{I}$ be a ∂-finite ideal of an Ore algebra $\mathbb{O} = \mathbb{K}[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$. Let P be any Ore polynomial in $\mathbb{O}$. Then for any f annihilated by $\mathfrak{I}$,* $\operatorname{Ann}(P \cdot f)$ *is also ∂-finite.*

PROOF. This follows from the inclusion $\mathbb{O}\cdot(P\cdot f) \subseteq \mathbb{O}\cdot f$ and the ∂-finiteness of $\operatorname{Ann} f \supset \mathfrak{I}$. □

The algorithm to find an operator vanishing on $P \cdot f$ consists in rewriting successive derivatives $\partial^\alpha P \cdot f$ in the finite basis formed with the pseudo-derivatives of f, and then finding a linear dependency by Gaussian elimination.

Putting all three lemmas together yields the following result for polynomial expressions in ∂-finite functions.

PROPOSITION 2.3. *Let $\mathbb{O} = \mathbb{K}[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$ be an Ore algebra and $\mathfrak{I}_1, \dots, \mathfrak{I}_n$ be ∂-finite ideals of $\mathbb{O}$. Assume that for all $i \in \{1, \dots, r\}$ there are polynomials $A_i(u)$ and $B_i(u)$ over $\mathbb{K}$ such that $\sigma_i = A_i(\partial_i)$ and $\delta_i = B_i(\partial_i)$, where the products denote compositions. Let P be an element of the polynomial ring $\mathbb{K}[u_1, \dots, u_p]$ and f_i be annihilated by $\mathfrak{I}_i$, $i = 1, \dots, n$. Then $P(\boldsymbol{\partial}^{r_1} \cdot f_{s_1}, \dots, \boldsymbol{\partial}^{r_p} \cdot f_{s_p})$ is ∂-finite with respect to $\mathbb{O}$ (when $r_i \in \mathbb{N}^r$, $s_i \in \{1, \dots, n\}$).*

In practice, one can apply the algorithms outlined above directly on $P(\boldsymbol{\partial}^{r_1} \cdot f_{s_1}, \dots, \boldsymbol{\partial}^{r_p} \cdot f_{s_p})$, instead of decomposing into sums of products. This has the nice property of often producing equations of a lower order (i.e., larger ideals).

EXAMPLE. Cassini's identity on the Fibonacci numbers reads

$$F_{n+2}F_n - F_{n+1}^2 = (-1)^n,$$

with $F_0 = F_1 = 1$ and $F_{n+2} = F_{n+1} + F_n$. In the Ore algebra $\mathbb{Q}[S_n; S_n, 0]$, the annihilating ideal $\mathfrak{I} = \operatorname{Ann} f$ of the Fibonacci numbers is generated by $S_n^2 - S_n - 1$. We consider the polynomial

$$P = (S_n^2 \cdot f)f - (S_n \cdot f)^2.$$

First each of the S_n^i is reduced modulo $\mathfrak{I}$, so that P is rewritten

$$P = (S_n \cdot f)f + f^2 - (S_n \cdot f)^2.$$

Then $S_n \cdot P$ is reduced similarly, and this yields

$$S_n \cdot P = -(S_n \cdot f)f - f^2 + (S_n \cdot f)^2.$$

Thus Gaussian elimination detects that S_n+1 annihilates P, whereas the decomposition into sums of products yields the less precise annihilator $S_n^3 - 2S_n^2 - 2S_n + 1$ (which is a multiple of $S_n + 1$).

2.2.5. Closure under specialization

Proposition 2.4. *Let $\mathbb{K}$ be a field, $\mathbb{O} = \mathbb{K}(\mathbf{x})(y_1, \dots, y_q)[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$ be an Ore algebra, $f(\mathbf{x}, \mathbf{y})$ be a ∂-finite function with respect to $\mathbb{O}$ and $a_1, \dots, a_q$ be elements of $\mathbb{K}$. Then $g(\mathbf{x}) = f(\mathbf{x}, a_1, \dots, a_q)$ is ∂-finite with respect to $\mathbb{O}' = \mathbb{K}(\mathbf{x})[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$, and a rectangular system of* $\operatorname{Ann} g$ *can be computed (in $\mathbb{O}'$) from a system of generators of* $\operatorname{Ann} f$.

Again, this proposition could also be stated at the level of ∂-finite ideals.

Proof. Starting from a rectangular system for f, the algorithm simply consists in replacing $y_1, \dots, y_q$ by $a_1, \dots, a_q$ in the polynomials involving those ∂_i's that commute with the y_j, $j = 1, \dots, q$ and discarding the other ones. This process does not yield trivial equations provided (left) polynomial factors are removed from the input polynomials before substitutions. $\square$

If a system of generators of the ideal $\operatorname{Ann} f$ is given, for instance as a Gröbner basis calculated by closure operations, a system for $\operatorname{Ann} g$ is obtained by eliminating (by a Gröbner basis computation) the ∂_j's that do not commute with $y_1, \dots, y_q$ and then replacing $y_1, \dots, y_q$ by $a_1, \dots, a_q$. This system is not necessarily rectangular.

3. Polynomial Ore algebras and creative telescoping

The main success of Zeilberger's theory of holonomic functions is *creative telescoping* (Almkvist and Zeilberger, 1990; Takayama, 1990b; Wilf and Zeilberger, 1992a; Zeilberger, 1990b, 1991a, 1991b). This is an algorithm to compute equations satisfied by definite sums or integrals. Examples of applications of this algorithm were given in the introduction. We now generalize this algorithm to Ore algebras.

3.1. Indefinite ∂^{-1} and definite $\partial^{-1}|_\Omega$

Let $\mathfrak{I}$ be a ∂-finite ideal of an Ore algebra $\mathbb{O}_\mathrm{r} = \mathbb{K}(\mathbf{x})[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$ and f be a generator of the module $\mathbb{O}_\mathrm{r}/\mathfrak{I}$ (for instance f can be the residue class of 1). We view f as an element of an algebra $\mathcal{F}$ of "functions" on which the action of $\mathbb{O}_\mathrm{r}$ is defined. Assume that all the x_j's commute with all the ∂_k's except a single one, ∂_i. (In practice, we often have $\mathbb{O}_\mathrm{r} = \mathbb{K}(x_i)[\boldsymbol{\partial}; \boldsymbol{\sigma}, \boldsymbol{\delta}]$ for a single indeterminate x_i). Note that $\mathbb{K}$ may contain other indeterminates, provided that they commute with ∂_i.

We assume an *indefinite* operator ∂_i^{-1} and a *definite* operator $\partial_i^{-1}|_\Omega$ exist, with the property that they commute with all the ∂_j's of the algebra such that $i \neq j$. In addition, we assume that they satisfy

$$\partial_i^{-1}\partial_i = \partial_i\partial_i^{-1} = 1 \qquad \text{and} \qquad \partial_i^{-1}|_\Omega\partial_i = \partial_i\partial_i^{-1}|_\Omega = 0. \tag{3.1}$$

The indefinite operator ∂_i^{-1} corresponds to the indefinite sum or integration operator when ∂_i is the difference or differentiation operator, provided the set $\mathcal{F}$ of functions satisfies some analytic conditions. For instance, D_x and $D_x^{-1} = \int_{-\infty}^x dt$ commute on $\mathbb{Q}(x)e^{-x^2}$. Similarly $\Delta_n = S_n - 1$ and $\Delta_n^{-1} = \sum_{-\infty}^{n-1}$ commute on many classes of expressions involving binomial coefficients; in the q-differential case, the operators of q-differentiation $H_{q,x}$ and q-integration $H_{q,x}^{-1} = \int_0^x d_q t$, defined at a function f by

$$(H_{q,x} \cdot f)(x) = \frac{f(qx) - f(x)}{(q-1)x} \qquad \text{and} \qquad \int_0^x f(t) d_q t = (1-q) \sum_{n=0}^{\infty} f(q^n x) q^n x,$$

commute on a large class of functions (see (Gasper and Rahman, 1990)). The analytic conditions required on $\mathcal{F}$ correspond to setting constants of integration or summation, so that the left-hand part of Eq. (3.1) is satisfied.

In the same cases as above, the definite operator $\partial_i^{-1}|_\Omega$ corresponds to the definite integration, sum or q-integration operator respectively, this latter being defined by

$$\int_0^{+\infty} f(t) d_q t = (1-q) \sum_{n=-\infty}^{+\infty} f(q^n) q^n.$$

The constraint expressed in the right-hand part of (3.1) is actually a constraint on $\mathcal{F}$. Frequently, it corresponds to summing or integrating over a domain $\partial\Omega$ on which all the $\boldsymbol{\partial} \cdot f$, including f, vanish.

To compute indefinite ∂_i^{-1} or definite $\partial_i^{-1}|_\Omega$, the first step of creative telescoping consists in finding a polynomial $P \in \mathfrak{I}$ which does not contain any x_j. Euclidean division by ∂_i (which is commutative in this case) can then be used to produce two polynomials A and B such that both of them do not contain any x_j, B does not contain ∂_i, and

$$P \cdot f = 0 = A \cdot (\partial_i \cdot f) + B \cdot f = \partial_i \cdot (A \cdot f) + B \cdot f. \tag{3.2}$$

Next, left multiplying by ∂_i^{-1} and $\partial_i^{-1}|_\Omega$ and using the commutation rules (3.1) yields

$$B \cdot (\partial_i^{-1} \cdot f) = -A \cdot f, \tag{3.3}$$

$$B \cdot (\partial_i^{-1}|_\Omega \cdot f) = 0, \tag{3.4}$$

since ∂_i^{-1} and $\partial_i^{-1}|_\Omega$ commute with B.

In the definite case, we have found an operator, namely B, that vanishes at $\partial_i^{-1}|_\Omega f$. When a Gröbner basis G is known for the ideal $\mathfrak{I}$, the *certificate* of the identity (3.4) consists in the coefficients of A and B, together with G. Given this certificate, the proof of the identity (3.4) reduces to checking (3.2), which is a routine reduction by G in the finite-dimensional vector space $\mathbb{O}_{\mathrm{r}}/\mathfrak{I}$. In the hypergeometric case, this reduction itself only involves rational function manipulations. In the indefinite case, one appeals to Lemma 2.3 in order to compute polynomials C annihilating $A \cdot f$. Then for such polynomials, CB is a polynomial annihilating $\partial_i^{-1} \cdot f$. It was known beforehand that $(\operatorname{Ann} f)\partial_i \subset \operatorname{Ann}(\partial_i^{-1} \cdot f)$. Usually however, some of the polynomials CB found by the previous algorithm lie outside $(\operatorname{Ann} f)\partial_i$, so that this algorithm increases the information on $\partial_i^{-1} \cdot f$. This is necessary for further computations because $(\operatorname{Ann} f)\partial_i$ is not ∂-finite in the multivariate case.

There are two difficulties with this technique, which both reside in the first step. The first one is to determine whether there exists a non-zero polynomial P in $\mathfrak{I}$ which does not contain any x_j. The second one is to find such a polynomial, or better yet a basis of them when they exist.

The first question can be addressed partially by computing the *dimension* of the ideal; this will be discussed further in §4.2.

Our approach to the second question consists in using a Gröbner basis computation to perform the elimination of $\mathbf{x}$. We are therefore led to work in the *polynomial* Ore algebra $\mathbb{O}_{\mathrm{p}} = \mathbb{K}[\mathbf{x}][\boldsymbol{\partial};\boldsymbol{\sigma},\boldsymbol{\delta}]$. We then just have to compute a Gröbner basis for an appropriate elimination order. This basis contains polynomials free of $\mathbf{x}$ if such polynomials exist in the ideal. (See however the comments in §4.3.) In the definite case, each of the polynomials in the basis provides the certificate of a corresponding identity.

Example. Let $f(x,y) = (1+xy+y^2)^{-2}$. We want to compute the indefinite integral $F(x,y) = \int_y^{+\infty} f(x,t)\,dt$, which converges for any x. We thus work in the algebra $\mathbb{O}_{\mathrm{r}} = \mathbb{K}(y)[D_x;1,D_x][D_y;1,D_y]$, where $\mathbb{K} = \mathbb{C}(x)$. The function f is annihilated by both operators

$$p_y = (1+xy+y^2)D_y + 2x + 4y \qquad \text{and} \qquad p_x = (1+xy+y^2)D_x + 2y,$$

from which trivially follows that the indefinite integral F satisfies $p_y D_y$ and $p_x D_y$. Our goal is to find other operators satisfied by F by using the previous algorithm.

Eliminating y between the polynomials p_x and p_y in $\mathbb{O}_{\mathrm{p}} = \mathbb{K}[y][D_x;1,D_x][D_y;1,D_y]$ yields

$$P = AD_y + B \qquad \text{where} \qquad \begin{cases} A = x(x-2)(x+2)D_x + xD_y + 2x^2 + 2, \\ B = -(x(x-2)(x+2)D_x^2 + 4(x^2+1)D_x). \end{cases}$$

To compute those C such that $CA = 0$ modulo the ideal generated by p_x and p_y in $\mathbb{O}_{\mathrm{r}}$, we introduce new commutative indeterminates t, u, v and w, and eliminate t between the polynomials

$$u - tA, \qquad v - tp_x \qquad \text{and} \qquad w - tp_y,$$

by computing a Gröbner basis in the algebra $\mathbb{C}(x,y,u,v,w)[t][D_x;1,D_x][D_y;1,D_y]$. In this Gröbner basis, those polynomials which do not involve t are of the form $uU + vV + wW$, where U, V and W are polynomials in $\mathbb{O}_{\mathrm{r}}$ such that $UA + Vp_x + Wp_y = 0$. Right multiplication of the U's that we obtain in this way by B yields new operators satisfied by F:

$$\begin{aligned}
&(y(x^3y + 4x^2 + 4 + 16xy + 4x^2y^2 + 4y^2) + (1+xy+y^2)(x^2y^2 + y^2 + 3xy + 1)D_y) \\
&\qquad \times (4(x^2+1)D_x + x(x-2)(x+2)D_x^2), \\
&\quad (32y^2 + 32xy + 8 + 48x^2y^2 + 36xy^3 + 12x^3y^3)D_x \\
&\qquad + x(15y^4 + 5x^2y^4 + 24xy^3 + 8x^3y^3 - 2y^2 + 32x^2y^2 + 28xy + 7)D_x^2 \\
&\qquad + (x-2)(x+2)(1+xy+y^2)(x^2y^2 + y^2 + 3xy + 1)D_x^3.
\end{aligned}$$

Computing a Gröbner basis from those polynomials adjoined to the ones known beforehand, $p_x D_y$ and $p_y D_y$, finally yields a basis of a subideal of $\operatorname{Ann} F$ constituted of $p_x D_y$, $p_y D_y$ and a third polynomial

$$x(x-2)(x+2)(1+xy+y^2)D_x^2 + 4(x^2+1)(1+xy+y^2)D_x - (2x^2y^2 + 2y^2 + 6xy + 2)D_y,$$

from which follows that $\operatorname{Ann} F$ is ∂-finite.

Rather than obtaining the C's by a Gröbner basis computation, as above, we could have used the extended Euclidean algorithm. In this instance, this would have yielded the same final description of F.

Finally, note that this approach based on an elimination by Gröbner bases makes it possible to compute multiple summations and/or integrations by a single elimination. It directly extends the corresponding algorithms for the hypergeometric case (Wilf and Zeilberger, 1992b).

3.2. Example of creative telescoping by Gröbner bases

We illustrate the use of Gröbner bases in Ore algebras to compute annihilators of the generating function of the Jacobi polynomials

$$\sum_{n=0}^{\infty} P_n^{(a,b)}(x) y^n. \tag{3.5}$$

We refer the reader to (Parnes and Ekhad, 1992) for another automatic treatment of this generating function. However, in this reference, a closed form for the generating function and a recurrence for the Jacobi polynomials are known beforehand. The equality between the closed form and the sum (3.5) is then *checked* by extracting coefficients. Here, we *compute* a closed form for the generating function starting from equations defining the Jacobi polynomials.

We load the package

```
with(Mgfun):
```

We create the polynomial Ore algebra $\mathbb{Q}(a,b,x,y)[n][S_n;S_n,0][D_x;1,D_x][D_y;1,D_y]$,

```
A:=orealg(comm=[a,b],shift=[Sn,n],diff=[Dx,x],diff=[Dy,y]):
```

To get the equations for $P_n^{(a,b)}(x)y^n$ one could use the Ore polynomials (1.11–1.13) annihilating $P_n^{(a,b)}(x)$, define y^n as a solution of $\{S_n - y, yD_y - n\}$ and appeal to closure under product described in §2.2. A more direct way consists in noting that the differential equation (1.13) is also satisfied by $P_n^{(a,b)}(x)y^n$, while a recurrence is obtained by changing S_n into $y^{-1}S_n$ in recurrence (1.11). This yields four operators:

$$c_2 S_n^2 + c_1 S_n + c_0, \quad c_1' D_x S_n + c_0' S_n, \quad c_2'' D_x^2 + c_1'' D_x + c_0'', \quad yD_y - n,$$

with coefficients that are polynomials in n, x, y, a, b. In Maple syntax, the system of generators of Ann $P_n^{(a,b)}(x)y^n$ is thus

```
G:=[2*(n+2)*(n+a+b+2)*(2*n+a+b+2)*Sn^2
        -y*(2*n+a+b+3)*(a^2-b^2+4*x*n^2+4*x*n*a+4*x*n*b
            +12*x*n+x*a^2+2*x*a*b+6*x*a+x*b^2+6*x*b+8*x)*Sn
        +2*(n+a+1)*(n+b+1)*(2*n+a+b+4)*y^2,
    -2*(n+a+1)*(n+b+1)*y+(n+1)*(-a+b+2*x*n+x*a+x*b+2*x)*Sn
        -(x-1)*(x+1)*(2*n+a+b+2)*Dx*Sn,
    n*(n+a+b+1)+(b-a-x*a-x*b-2*x)*Dx-(x-1)*(x+1)*Dx^2,
    y*Dy-n]:
```

To compute the sum for non-negative n, we start by eliminating n. We therefore define an appropriate term order (i.e., such that $n \succ S_n$, $n \succ D_x$ and $n \succ D_y$) by using the *Mgfun* command `termorder`:

```
T:=termorder(A,lexdeg=[[n],[Sn,Dx,Dy]]):
```

The elimination is then obtained by a simple Gröbner basis computation (`gbasis` command, with basis and term order as inputs):

```
GB:=gbasis(G,T):
```

This basis consists of six polynomials which vanish on $P_n^{(a,b)}(x)y^n$, only the first one of which contains n. As usual in such calculations, the intermediate result of the Gröbner basis is rather complicated. It consists of six polynomials:

$$\begin{gathered} yD_y - n, \qquad\qquad p_xD_x + p_yD_y + p_{xx}D_x^2 + p_{yy}D_y^2, \\ q_{yn}D_yS_n + q_yD_y + q_{yy}D_y^2 + q_{xyn}D_xD_yS_n + q_{xn}D_xS_n + q_{yyn}D_y^2S_n, \\ r_1 + r_{yyyn}nD_y^3S_n + r_{yn}D_yS_n + r_xD_x + r_yD_y + r_{yy}D_y^2 + r_{yyy}D_y^3 \\ \qquad + r_{xn}D_xS_n + r_{xyy}D_xD_y^2 + r_{xy}D_xD_y + r_{yyn}D_y^2S_n, \\ s_1 + s_{yn}D_yS_n + s_nS_n + s_{ynn}D_yS_n^2 + s_{yy}D_y^2 + s_{xn}D_xS_n + s_{yyn}D_y^2S_n + s_{yynn}D_y^2S_n^2, \\ t_{yn}D_yS_n + t_{ynn}D_yS_n^2 + t_{xnn}D_xS_n^2 + t_{xn}D_xS_n + t_nS_n, \end{gathered}$$

with polynomial coefficients which we do not display.

The next step of creative telescoping consists in substituting S_n by 1 in these operators (i.e., in computing the remainder of the Euclidean division by the difference operator $S_n - 1$). This substitution is performed by the Maple `subs` command; the result is presented in a readable way by the `collect` command:

```
CT:=collect(subs(Sn=1,[GB[2..6]]),[Dx,Dy],distributed,factor);
```

$$\begin{aligned} CT := [&-2y(1+b)(1+a)(a+2xa+2xb-b+2x) - 4y^3(yx-1)D_y^3 \\ &\quad + (x-1)(x+1)(-4yb-4y-4ya-b^2-4yab+a^2)D_x \\ &\quad - y(-2yb^2+4yb^2x+12yxab+24yxb+6ya-6yb+28yx+4ya^2x \\ &\qquad + 24yxa+2ya^2-6b-a^2+xa^2-b^2-6a-6ab-xb^2-4)D_y \\ &\quad - 4y^2(x-1)(x+1)(a+b+3)D_xD_y - 4y^3(x-1)(x+1)D_xD_y^2 \\ &\quad - 2y^2(-yb+ya+4yxa+4yxb+12yx-3a-6-3b)D_y^2, \\ & y(a^2+b^2+a+b-xb^2+xa-xb+xa^2) + (x-1)(x+1)(a+b-yb+ya)D_x \\ &\quad + y(ya+yxa-yxb+yb-xb-xa+b-a)D_y, \\ & (b-a-xa-xb-2x)D_x + y(2+a+b)D_y - (x-1)(x+1)D_x^2 + y^2D_y^2, \\ & -2y(1+b)(1+a) - (x-1)(x+1)(b+a)D_x \\ &\quad + y(-6y-2ya-2yb-a+b+xa+2x+xb)D_y \\ &\quad - 2y(x-1)(x+1)D_xD_y + 2y^2(-y+x)D_y^2, \\ & -2y-2yab+3xb+3xa-b^2+a-b+xa^2+2x-2yb-2ya+a^2+xb^2 \end{aligned}$$

$$
\begin{aligned}
&+ (-2b - 2 - 2a - 2y^2b - 6y^2 + 4yxa - 2y^2a + 8yx + 4yxb)D_y\\
&+ 2xab + 2(x-1)(x+1)D_x + 2y(-1 + 2yx - y^2)D_y^2]
\end{aligned}
$$

The whole computation takes 17 seconds[†]. It is then possible to compute a rectangular system out of these equations: it is obvious from the second and fifth polynomials in CT that the ideal generated by CT is ∂-finite, so that the method of §2.2 applies. This yields the two second order operators

$$
\begin{aligned}
&2y(1 + y^2 - 2yx)(ya - yb + a + b)D_y^2\\
&+ (2a + 2b - 4yb^2x + 2y^3a^2 - 2y^3b^2 + 4y^2ab - 4ya^2x + 2a^2 + 2b^2 + 4ab\\
&\quad + 8y^2a + 6ay^3 - 6y^3b + 8y^2b + 2y^2a^2 + 2y^2b^2 - 10yxa - 10yxb - 8yxab\\
&\quad - 6y^2xa + 6y^2xb + 2ya^2 - 2yb^2 - 4y^2a^2x + 4y^2b^2x)D_y\\
&+ ab^2 - a^2b + yb^2x + 3ya^2b - 3xb^2 - 3xa^2 - 6xab + xb^3y - xa^3y - a^3y\\
&\quad - b^3y - ya^2x - 2xb + 3yab^2 - 2xa - a^2 + b^2 + 6yab + 4ya + 4yb + 2y^2a\\
&\quad - 2y^2b + 2y^2a^2 - 2y^2b^2 + 2y^2a^2b - 2y^2ab^2 - 3xa^2b - 3xb^2a\\
&\quad - xa^3 - xb^3 + 3ya^2 + 3yb^2 - ya^2bx + yab^2x - a^3 + b^3
\end{aligned}
$$

and

$$
\begin{aligned}
&2(x-1)(x+1)(1 + y^2 - 2yx)(yb - yxb + ya + yxa - xa - xb - a + b)D_x^2\\
&+ (-16y^2ax^2 - 16y^2bx^2 - 2a - 2b + 8yb^2x - 4y^2x^2ab + 4y^3a^2x\\
&\quad + 4y^3b^2x + 2y^3a^2 - 2y^3b^2 + 4y^2ab + 4xb^2 - 4xa^2 + 4a^2x^3y\\
&\quad - 4a^2y^2x^3 + 8ax^3yb - 2ax^2 + 10y^2x^3b + 8ya^2x + 10yx^3b\\
&\quad - 2x^2b^2 - 2y^3x^2b - 2b^2y^3x^2 + 4xb + 4x^3b^2y^2 - 10y^2b^2x^2\\
&\quad - 4x^2ab + 4x^3b^2y - 4xa - 10yb^2x^2 - 2a^2 - 2b^2 + 2a^2x^2y^3 + 4ab\\
&\quad - 2x^2a^2 - 4ya + 4yb + 4xy^3b + 10a^2x^2y + 4y^2a + 2ay^3 - 2y^3b\\
&\quad - 2x^2b + 4y^2b - 2y^2a^2 - 2y^2b^2 - 10a^2y^2x^2 + 2yxa + 2yxb\\
&\quad - 8yxab - 2y^2xa + 2y^2xb + 2ya^2 - 2yb^2 + 2ay^3x^2 + 10ayx^3\\
&\quad - 8y^2a^2x - 10ay^2x^3 + 8y^2b^2x + 4ay^3x + 16yax^2 - 16ybx^2)D_x\\
&- y(ab^2 + a^2b - b^3x^2 - a^3x^2 - 2a - 2b + 6yb^2x + 3ya^2b - 4y^2ab + 6xb^2\\
&\quad - 6xa^2 - 2ax^2 + 2xb^3y + 2xa^3y + a^3y - b^3y + 6ya^2x - 3x^2b^2\\
&\quad + 4xb - 3yab^2 - 6x^2ab - 4xa - 3yb^2x^2 - 3a^2 - 3b^2 + 2ab - 3x^2a^2\\
&\quad + 2ya - 2yb + 3a^2x^2y - 3a^2x^2b - 2x^2b - 2y^2a^2b - 2y^2ab^2\\
&\quad - 3b^2x^2a - 2xa^2b + 2xb^2a + 4yxa + 4yxb - 2xa^3 + 2xb^3 + 8yxab\\
&\quad + 2y^2b^2xa - yb^2x^2a + ya^3x^2 - yb^3x^2 + 3ya^2 - 3yb^2\\
&\quad + 4ya^2bx + 4yab^2x + ya^2x^2b - 2y^2a^2xb - a^3 - b^3 + 2yax^2 - 2ybx^2).
\end{aligned}
$$

From this system and initial conditions, a differential equation solver can find the

[†] All our timings are obtained on a DecStation 3000 300X (Alpha).

generating function of the Jacobi polynomials

$$\frac{1}{\sqrt{1-2yx+y^2}\left(1-y+\sqrt{1-2yx+y^2}\right)^a\left(1+y+\sqrt{1-2yx+y^2}\right)^b}.$$

Even when solving is not possible, these equations can be used to check such a conjectured right-hand side, or more importantly to proceed with further computations when no closed-form exists or is available. The verification would be as follows. We first define the generating function P, and then we apply each operator that we have calculated to P (the command is `applyopr`, and `simplify` simplifies the result):

```
R:=sqrt(1-2*x*y+y^2): P:=1/(R*(1-y+R)^a*(1+y+R)^b):
map(simplify,map(applyopr,CT,P,A));
```

$$[0,0,0,0,0]$$

Checking the initial conditions at 0 then proves that this solution is the generating function that we were looking for.

3.3. Extension of Takayama's algorithm for definite $\int_\Omega$ to definite $\partial^{-1}|_\Omega$

The elimination of the variables in $\mathbf{x}$ to perform creative telescoping when summing or integrating a function f is much stronger than what is necessary (Almkvist and Zeilberger, 1990) and can result in operators of order larger than necessary, or in a failure to compute the definite $\partial^{-1}|_\Omega$. It is actually often sufficient to determine an element of the ideal $\operatorname{Ann} f$ which can be written $\partial_i A + B$, where only B needs to commute with ∂_i and to be computed. In other words, the only polynomial which needs to be computed has to be searched for in $\operatorname{Ann} f + \partial_i \mathbb{O}_{\mathrm{r}}$. This set is the sum of a left and of a right ideal of $\mathbb{O}_{\mathrm{r}}$. As such, is it generally not an ideal, but only a left $\mathbb{K}[\boldsymbol{\partial};\boldsymbol{\sigma},\boldsymbol{\delta}]$-module.

An elimination algorithm based on Gröbner bases for modules was developed by Takayama (1990a,1990b) to solve this problem in the context of the Weyl algebra. This algorithm is based on a generalization to the non-commutative case of a classical technique to compute Gröbner bases of polynomial modules (see (Becker and Weispfenning, 1993, §10.4)). Takayama's algorithm is readily adapted to the context of Ore algebras and results in faster computations than when using the algorithm of the previous section. We now present an optimized version of Takayama's algorithm, extended to Ore algebras.

Since the aim is to compute B, during the intermediate computations one can replace all the polynomials which can be rewritten $\partial_i C$ for some C by zero, provided these polynomials will not be multiplied by any x_j in later computations. If the Ore algebra satisfies the hypothesis of Theorem 1.2 (which is necessary if we want to compute Gröbner bases), the idea is that this simplification can be achieved by computing Gröbner bases of (not finitely generated) $\mathbb{O}_{\mathrm{p}}$-modules.

The hypothesis of Theorem 1.2 on the a_{ij}'s implies that all the $x_j^p\partial_i^k$ can be rewritten as polynomials of lower degree in ∂_i using

$$\partial_i^k x_j^p = c x_j^p \partial_i^k + \text{lower order terms} \tag{3.6}$$

provided the left hand-side can be replaced by zero. The algorithm then considers the $\mathbb{O}_{\mathrm{p}}$-submodules of $\mathfrak{I} + \partial_i\mathbb{O}$ consisting of polynomials of total degree at most N in $\mathbf{x}$

for $N = 0, 1, 2, \ldots$ A suitable generalization of Gröbner bases of these (finitely generated) modules is computed in three steps: first the generators of $\mathfrak{I}$ are left-multiplied by powers of the elements of $\mathbf{x}$ to produce all of the possible operators of total degree at most N in $\mathbf{x}$. Then ∂_i is eliminated from these operators using (3.6). Finally a generalized Gröbner basis for this system of operators in $\mathbb{K}[\mathbf{x}][\boldsymbol{\partial} \setminus \partial_i]$ is computed by the usual Buchberger algorithm, except that multiplications by elements of $\mathbf{x}$ are not allowed when computing syzygies or reductions.

An optimized version of the algorithm is as follows (here we denote by $G_0 \subset G$ the set of polynomials in G of degree 0 in $\mathbf{x}$). Our optimization consists in the preprocessing via (3.6).

 # The input is a set P of operators $P_1, \ldots, P_p$ of degree $d_1, \ldots, d_p$ in $\mathbf{x}$
$G := \{\}$
for N from $\min(d_1, \ldots, d_p)$ while $G_0 = \{\}$ do
  $H := \left\{ x_1^{p_1} \cdots x_s^{p_s} P_i ; p_j \geq 0, \sum p_j = N - d_i, d_i \leq N \right\}$
  reduce H using (3.6)
  $G :=$ module-gbasis$(G \cup H)$
od
return G_0

It is worth noting that the reduction by (3.6) is usually very simple. In the case of a differential operator D_x, it consists in replacing monomials $p(x)D_x^k$ by $(-1)^k p^{(k)}(x)$. In the case of a difference operator $\Delta_n = S_n - 1$, it consists in replacing monomials $p(n)S_n^k$ by $p(n-k)$. Similarly, in the case of the q-difference operator, it consists in replacing monomials $p(n, q^n)S_n^k$ by $p(n-k, q^{n-k})$.

The condition under which the algorithm should be stopped can be modified depending on the context. In the Weyl algebra case, Takayama chooses to stop the loop when the basis spans a holonomic ideal and he proves that this always happens in finite time. We do not have such a result in the general Ore algebra case. Thus we stop the algorithm as soon as one polynomial free of $\mathbf{x}$ has been found and termination is not guaranteed unless there exists such a polynomial (i.e., the definite $\partial_i^{-1}|_\Omega$ can be found by creative telescoping). Termination can only be guaranteed for special cases of ideals such as holonomic ideals in the Weyl algebra, or q-holonomic ideals in the q-case. In many cases, termination for particular ideals can also be algorithmically predicted by computing the *dimension* of the ideal (see §4.2).

The speed of this algorithm compared to the general one described in §3.1 may well make it the only practical one on large examples. However, it is worth noting that this algorithm computes in a different ideal than the general method. Thus the ideal generated by its output when stopping the loop as soon as a polynomial free of $\mathbf{x}$ has been found may be larger or smaller than the ideal obtained by the other algorithm. (But running the loop forever computes an increasing sequence of ideals which is stationary on a larger ideal than the one obtained by the other algorithm.) In practice, this new algorithm often returns operators of a smaller order than the general method. This increases the speed of subsequent computations.

Of course, the method is applicable to simulaneous multiple summations and/or integrations.

Example. In the same example as above, the computation now takes place in the simpler algebra

```
A:=orealg(comm=[a,b,n],diff=[Dx,x],diff=[Dy,y]):
```

It takes less than 6 seconds to find the following system of three operators which annihilate the generating function of the Jacobi polynomials:

$$\begin{cases} (x-1)(x+1)D_x^2 + (ax+bx+2x+a-b)D_x - y(a+b+2)D_y - y^2D_y^2, \\ 2y^2(x-y)D_y^2 + y(-2ay-2by-6y+2x+b-a+ax+bx)D_y \\ \qquad -(x-1)(x+1)(a+b)D_x - 2y(x-1)(x+1)D_xD_y - 2y(a+1)(b+1), \\ 4y^3(y^2-2xy+1)D_y^3 \\ \qquad -2y^2(-3y^2a-16y^2-3y^2b+6yxa+6yxb+22xy-3b-3a-6)D_y^2 \\ \qquad -y(-8y^2ab-26y^2a-52y^2-2y^2b^2-2y^2a^2-26y^2b-b^2y \\ \qquad\quad +32yxb+a^2y+40xy+5yb^2x+32yxa+14yaxb+5ya^2x-6b \\ \qquad\quad +a^2x-b^2x-4-a^2-6ab-6a-b^2)D_y \\ \qquad +(x-1)(x+1)(b^2y+a^2-4y-b^2-2yab+a^2y)D_x \\ \qquad -2y(1+b)(a+1)(-ay+2ax+a-by+2bx-b-6y+2x). \end{cases}$$

This is obtained with $N = 2$. It is not difficult to check that the ideal generated by these operators is ∂-finite. The next iteration of the loop takes 22 more seconds and produces the same more refined basis as the general algorithm.

3.4. Hypergeometric examples

Apart from his general theory of holonomic identities (Zeilberger, 1990b), Zeilberger, together with Wilf, developed specialized algorithms for the cases of hypergeometric and q-hypergeometric identities (Wilf and Zeilberger, 1992a)—see also (Koornwinder, 1993). It would be interesting to compare their efficiency to our approach and generalize as much as possible their good features (see (Chyzak, 1997) for first results). We now show using a few examples that the general approach outlined in this paper performs rather well in practice.

3.4.1. An identity between Franel and Apéry numbers

The following identity was proved by Strehl (1994):

$$\sum_{k=0}^{n} \binom{n}{k}^2 \binom{n+k}{k}^2 = \sum_{k=0}^{n} \binom{n}{k} \binom{n+k}{k} \sum_{j=0}^{k} \binom{k}{j}^3. \tag{3.7}$$

Both sides of this equation satisfy the operator

$$(n+2)^3 S_n^2 - \left((n+2)^3 + (n+1)^3 + 4(2n+3)^3\right) S_n + (n+1)^3. \tag{3.8}$$

(This operator was used by Apéry in his proof of the irrationality of $\zeta(3)$.)

Using the algorithm of §3.1, the computation is performed by *Mgfun* in 82 seconds. First, an operator of order 3 annihilating the inner sum of the right hand-side is obtained in 5 seconds; then 2 more seconds are necessary to compute operators annihilating the product by the two binomials using the technique of §2.2 and creative telescoping applied to these latter operators requires 31 seconds to yield an operator of order 7 annihilating the right hand-side of (3.7). Another creative telescoping yields an operator of order 4 annihilating the left hand-side of (3.7) in 44 seconds. The identity is then proved by checking 11 initial conditions (an upper bound for the order of the operator annihilating the difference). Then taking the gcrd of both operators yields (3.8).

A similar calculation using our version of Takayama's algorithm is performed in 11 seconds. Interestingly, the operators found by this method have a smaller order than those produced by the general algorithm. The inner sum of the right hand-side is found to satisfy an operator of order 2 in 4 seconds; then the product still takes 2 seconds and the second creative telescoping takes 2 seconds and yields an operator of order 2. The same operator (3.8) is obtained by applying this algorithm to the left hand-side and the computation takes 2.5 seconds.

3.4.2. A Rogers-Ramanujan identity

We consider the following finite version due to Andrews of one of the famous Rogers-Ramanujan identities:

$$\sum_k \frac{q^{k^2}}{(q;q)_k (q;q)_{n-k}} = \sum_k \frac{(-1)^k q^{(5k^2-k)/2}}{(q;q)_{n-k}(q;q)_{n+k}},$$

where $(q;q)_n = (1-q)\cdots(1-q^n)$.

Using the general method of §3.1, it takes one second to find a second order operator annihilating the left hand-side of this identity, and 56 seconds to find a fifth order operator annihilating the right hand-side. From this a proof is easily derived as above. Our generalization of Takayama's algorithm finds *the same operators* as the general method in 1 second and 23 seconds respectively.

It was noted by Paule (1994) that summing only the even part of the right-hand side (i.e., multiplying it by $(1+q^k)/2$) results in Zeilberger's algorithm finding an operator of order 2 for the right-hand side. Using the same trick with our algorithms, we find that Takayama's method benefits from it and yields an operator of order 3 instead of 5, while the more general algorithm yields an operator of order 6. As in the hypergeometric case, the reasons for this trick to work or not to work are not fully understood. This idea of *creative symmetrizing* is however of general applicability and extends to more general symmetries (Paule and Riese, 1996).

3.4.3. q-Dixon identity

The left hand-side of the identity

$$\sum_k (-1)^k q^{k(3k+1)/2} \binom{a+b}{a+k}_q \binom{a+c}{c+k}_q \binom{b+c}{b+k}_q = \frac{(q;q)_{a+b+c}}{(q;q)_a (q;q)_b (q;q)_c}$$

satisfies a system of operators in the variables S_a, S_b and S_c which can be obtained in 490 seconds using *Mgfun*. A simpler (but less complete) system is obtained using our version of Takayama's algorithm in only 70 seconds. Here is this simpler system:

$$\begin{cases} q^a(q^{b+c+1}-1)S_a - q^b(q^{a+c+1}-1)S_b + (q^a-q^b)S_aS_b, \\ q^a(q^{b+c+1}-1)S_a - q^c(q^{a+b+1}-1)S_c + (q^a-q^c)S_cS_a, \\ q^c(q^{a+b+1}-1)S_c - q^b(q^{a+c+1}-1)S_b + (q^c-q^b)S_cS_b. \end{cases}$$

It can be checked that these operators do not generate a ∂-finite ideal. One more iteration of the algorithm takes 166 seconds to produce generators of a ∂-finite ideal.

The same computation could be performed in an algebra containing the differentiation operator D_a instead of the shift operator S_a. What happens then is that our algorithm does not produce any operator in D_a (no such operator exists), but only the operators in S_b and S_c. This shows the importance of selecting an appropriate ambient algebra.

It is also possible to consider the operators S_k and S_a only, keeping b and c as parameters. Then, a third order equation in S_a is found, which is a ∂-finite description of the sum. (In (Paule and Riese, 1996), it is shown that creative symmetrizing applies to checking the identity in this context; strangely enough, our algorithm does not seem to benefit from creative symmetrizing in this example.)

4. Conclusion

4.1. The Weyl algebra case

It is well-known that in the special case of the Weyl algebra, many algorithms make it possible to compute equations for interesting operations. These operations apply to both univariate and multivariate cases.

In particular, algebraic functions are ∂-finite and an algorithm to compute differential equations from the polynomial equation exists (Comtet, 1964). Also, the composition of a ∂-finite function with algebraic functions is again ∂-finite and equations can be computed (Lipshitz, 1989; Stanley, 1980).

∂-finite functions are defined as solutions of differential equations with polynomial (or equivalently, rational) coefficients. There is in fact no enlargement of the class if we allow algebraic functions as coefficients: a function that satisfies a rectangular system with algebraic coefficients is ∂-finite and annihilators with polynomial coefficients can be computed.

Diagonals of ∂-finite functions are ∂-finite, and this is also effective (Lipshitz, 1988). This leads to the result that the Hadamard product of two ∂-finite power series is again ∂-finite, and again equations can be computed. Also, recurrence equations satisfied by the coefficients of a ∂-finite power series can be computed. All these operations are implemented in the univariate case in *gfun* (Salvy and Zimmermann, 1994) and are or will be implemented in the multivariate case in Chyzak's *Mgfun* package.

4.2. Holonomy

In the context of the Weyl algebra, Zeilberger (1990b) uses Bernstein's theory of *holonomic* systems to outline an important class of "functions" enjoying numerous closure properties and for which the elimination of any x_i is always guaranteed to succeed. He extends this technique to sequences and definite summation by considering generating functions. A nice property of the Weyl algebra is that holonomy is equivalent to ∂-finiteness (Kashiwara, 1978)—see also (Takayama, 1992). More precisely, if $\mathfrak{I}$ is a ∂-finite ideal in $\mathbb{K}(\mathbf{x})[\boldsymbol{\partial};\boldsymbol{\sigma},\boldsymbol{\delta}]$, then $\mathfrak{I}\cap\mathbb{K}[\mathbf{x}][\boldsymbol{\partial};\boldsymbol{\sigma},\boldsymbol{\delta}]$ is holonomic, and conversely.

Unfortunately, this equivalence breaks down in the case of general Ore algebras, which is why in this paper, we have focussed on ∂-finite functions and on equations with rational functions coefficients. Bernstein's theory of holonomy (Bernstein, 1971,1972) deals with polynomial coefficients and relies on a theory of dimension for ideals and modules. In a Weyl algebra on n differentiation symbols $D_1,\dots,D_n$, holonomic modules are those of least possible Bernstein dimension, namely n. Thus it is easy to check whether an ideal is holonomic when a system of its generators has been given (via Gröbner basis computations for instance).

Computations of dimensions via Gröbner bases can also be performed in Ore algebras. When the dimension of an ideal is sufficiently small, then the existence of a polynomial without one of the variables is guaranteed. This gives an *a priori* test of a sufficient condition for creative telescoping to function (see (Chyzak, 1998) for more on this subject).

Another approach would be to generalize holonomy to Ore algebras. The difficulty consists in finding a class of ideals of Bernstein dimension less than or equal to n closed under product. This will be the subject of future work. A partial solution to this problem is available for q-calculus (Sabbah, 1993). The theory of *q-holonomy* mimicks that of holonomy and yields analogous results of closure. Algebras of operators under consideration are direct products of algebras of the form $\mathbb{K}(q)[q^n,q^{-n}][S_n;S_n,0][S_n^{-1};S_n^{-1},0]$. Creative telescoping in this framework is possible, but requires the simultaneous elimination of q^n and q^{-n}, which cannot be done by a direct Gröbner basis computation. Developping this approach further and extending it to other Ore algebras could allow us to perform all the calculations at the level of *polynomial* Ore algebras, working with "Ore-holonomic" functions rather than with ∂-finite ones. The clear advantage would be to ensure holonomy of the systems on which we perform creative telescoping, and to avoid the extension/contraction problem.

4.3. The extension/contraction problem

The functions that we work with are naturally specified by operators in an Ore algebra $\mathbb{O}_{\mathrm{r}}=\mathbb{K}(\mathbf{x})[\boldsymbol{\partial};\boldsymbol{\sigma},\boldsymbol{\delta}]$, while the algorithms we use for creative telescoping need the elimination of some of the x_i's. The use of Gröbner basis computations to perform this elimination leads us to describe functions with operators in the smaller *polynomial* Ore algebra $\mathbb{O}_{\mathrm{p}}=\mathbb{K}[\mathbf{x}][\boldsymbol{\partial};\boldsymbol{\sigma},\boldsymbol{\delta}]$. Let $p_1,\dots,p_r$ be polynomials in $\mathbb{O}_{\mathrm{p}}$ generating a left ideal $\mathfrak{I}\subseteq\mathbb{O}_{\mathrm{p}}$, then they also generate an ideal $\mathfrak{K}\subseteq\mathbb{O}_{\mathrm{r}}$. However the actual ideal we are interested in is the *contraction* $\mathfrak{I}'=\mathfrak{K}\cap\mathbb{O}_{\mathrm{p}}$, which can be larger than the original ideal $\mathfrak{I}$.

In the case of a ∂-finite function f, this extension/contraction problem means that even if we are given generators $(p_1,\dots,p_r)$ of the ideal $\mathfrak{K}_f$ of all polynomials in $\mathbb{O}_{\mathrm{r}}$ that vanish at f, the ideal $\mathfrak{I}=(p_1,\dots,p_r)\subseteq\mathbb{O}_{\mathrm{p}}$ is not necessarily an accurate description of f. Therefore, elimination of one $\mathbf{x}$ between the p_i's may lead to zero, even when $\mathfrak{K}_f\cap\mathbb{K}[\boldsymbol{\partial};\boldsymbol{\sigma},\boldsymbol{\delta}]$ contains a non-zero polynomial.

Example. The binomial coefficients $u_{n,k}=\binom{n}{k}$ are annihilated by the Ore polynomials $P=(n+1-k)S_n-(n+1)$ and $Q=(k+1)S_k-(n-k)$ in the Ore algebra $\mathbb{O}_{\mathrm{r}}=\mathbb{K}(n,k)[S_n;S_n,0][S_k;S_k,0]$ built on two shift operators S_n and S_k. Any ideal larger than $\mathfrak{K}=(P,Q)$ in $\mathbb{O}_{\mathrm{r}}$ is $\mathbb{O}_{\mathrm{r}}$ itself. Pascal's triangle rule is represented by the operator $R = S_n S_k - S_k - 1$, which is easily found to be an element of $\mathfrak{K}$. Therefore $R \in \mathfrak{I}'$. However, in the difference algebra $\mathbb{O}_{\mathrm{p}} = \mathbb{K}[n,k][S_n; S_n, 0][S_k; S_k, 0]$, the ideal $\mathfrak{I} = (P,Q)$ does *not* contain R, although it contains $(n+1)R$, and $(k+1)R$, which is sufficient to make it possible to find the result R by Gröbner basis computation (with ideals in $\mathbb{K}(n)[k][S_n; S_n, 0][S_k; S_k, 0]$).

Example. Diagonals can be computed by creative telescoping. If $f(x,y)$ is a ∂-finite power series, then its diagonal is the coefficient of s^{-1} in $F(x,s) = f(s, x/s)/s$. By Cauchy's theorem, this is obtained by computing the definite integral of F with respect to s. From generators of an ideal annihilating f, it is not difficult to obtain generators of an ideal $\mathfrak{K} \subseteq \mathbb{O}_{\mathrm{r}} = \mathbb{K}(s,x)[D_s; 1, D_s][D_x; 1, D_x]$ annihilating F, and then one has to eliminate s. However, the success of this elimination with our algorithms requires generators of a sufficiently large ideal $\mathfrak{I} \subseteq \mathfrak{K} \cap \mathbb{O}_{\mathrm{p}}$, with $\mathbb{O}_{\mathrm{p}} = \mathbb{K}[s,x][D_s; 1, D_s][D_x; 1, D_x]$, such that $\mathfrak{I}$ contains a polynomial free of s.

For instance, to compute the diagonal of $f = 1/(1-(x+y))$ requires finding an operator free of s in $\mathbb{O}_{\mathrm{p}}$ which annihilates $F = 1/(s^2 - s + x)$. The annihilating ideal of F in $\mathbb{O}_{\mathrm{r}}$ is $\mathfrak{K} = (P,Q)$ where $P = D_s(s^2 - s + x) = (s^2 - s + x)D_s + (2s-1)$ and $Q = D_x(s^2 - s + x) = (s^2 - s + x)D_x + 1$. Once again, any larger ideal in $\mathbb{O}_{\mathrm{r}}$ is $\mathbb{O}_{\mathrm{r}}$ itself. The operator $U = D_s^2 + (4x-1)D_x^2 + 6D_x$ vanishes at F, so that $U \in \mathfrak{K}$, hence $U \in \mathfrak{K} \cap \mathbb{O}_{\mathrm{p}}$. However, U is *not* an element of $\mathfrak{I} = (P,Q)$ in $\mathbb{O}_{\mathrm{p}}$. It follows that the calculation of the diagonal of f from P and Q cannot be performed by elimination in $\mathbb{O}_{\mathrm{p}}$. However, if one takes the generator $R = (s^2 - s + x)D_s D_x + 2D_s \in \mathfrak{K}$, then the ideal $(P,Q,R) \subseteq \mathbb{O}_{\mathrm{p}}$ contains the operator U and our algorithm finds it. Furthermore, our version of Takayama's algorithm then finds the simpler $(1-4x)D_x - 2$.

In the commutative case, the contraction $\mathfrak{I} \cap \mathbb{K}[\mathbf{x}, \mathbf{y}]$ of an ideal $\mathfrak{I} \subseteq \mathbb{K}(\mathbf{x})[\mathbf{y}]$ can be computed. An algorithm (Becker and Weispfenning, 1993, algorithms cont and ideal-div2) is based on the calculation of the *ideal quotient* $\mathfrak{I} : f^\infty$, i.e., the set of all p such that $f^s p \in \mathfrak{I}$ for a positive integer s. This algorithm does not extend trivially to the skew case. A recent algorithm to compute a basis of this ideal $\mathfrak{I}'$ from a basis of $\mathfrak{K}$ will be presented in (Chyzak, 1998). This algorithm recovers the operators missing in the previous examples.

Acknowledgement

We thank the anonymous referees for encouraging us to rewrite this paper in a more pedagogical way. Philippe Dumas, Peter Paule, Marko Petkovšek and Volker Strehl spent a lot of time on a careful reading of a preliminary version and sent us many useful comments. This led us to clarify and simplify many of the ideas and to find better examples.

This work was supported in part by the Long Term Research Project Alcom-IT (#20244) of the European Union.

References

Almkvist, G., Zeilberger, D. (1990). The method of differentiating under the integral sign. *Journal of Symbolic Computation*, **10**:571–591.

Askey, R. (1992). The world of q. *CWI Quarterly*, **5**(4):251–270.

Becker, T., Weispfenning, V. (1993). *Gröbner bases*, volume 141 of *Graduate Texts in Mathematics*. Springer Verlag. A computational approach to commutative algebra (in cooperation with Heinz Kredel).

Bernstein, I. N. (1971). Modules over a ring of differential operators, study of the fundamental solutions of equations with constant coefficients. *Functional Analysis and Applications*, **5**(2):1–16 (Russian); 89–101 (English translation).

Bernstein, I. N. (1972). The analytic continuation of generalized functions with respect to a parameter. *Functional Analysis and Applications*, **6**(4):26–40 (Russian); 273–285 (English translation).

Buchberger, B. (1965). Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal. Ph. D. Thesis.

Buchberger, B. (1970). Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems. *Aequationes Mathematicae*, **4**:374–383.

Buchberger, B. (1985). Gröbner bases: an algorithmic method in polynomial ideal theory. In Bose, N. K., editor, *Recent trends in multidimensional system theory*, pages 184–232. D. Reidel Publishing Company, Dordrecht.

Cartier, P. (1992). Démonstration "automatique" d'identités et fonctions hypergéométriques. *Astérisque*, **206**:41–91. Séminaire Bourbaki.

Chyzak, F. (1994). Holonomic systems and automatic proofs of identities. Research Report 2371, Institut National de Recherche en Informatique et en Automatique.

Chyzak, F. (1997). An extension of Zeilberger's fast algorithm to general holonomic functions. In *Formal Power Series and Algebraic Combinatorics, 9th Conference*, pages 172–183. Universität Wien.

Chyzak, F. (1998). Thèse. In preparation.

Cohn, P. M. (1971). *Free Rings and Their Relations*. Number 2 in London Mathematical Society Monographs. Academic Press.

Comtet, L. (1964). Calcul pratique des coefficients de Taylor d'une fonction algébrique. *L'Enseignement Mathématique*, **10**:267–270.

Cox, D., Little, J., O'Shea, D. (1992). *Ideals, Varieties, and Algorithms*. Springer Verlag. An Introduction to Computational Algebraic Geometry and Commutative Algebra.

Ekhad, S. B., Majewicz, J. E. (1996). A short WZ-style proof of Abel's identity. *The Electronic Journal of Combinatorics*, **3**(2). The Foata Festschrift.

Erdélyi, A. (1981). *Higher Transcendental Functions*, volume 1–2–3. R. E. Krieger publishing Company, Inc., Malabar, Florida, second edition.

Fasenmyer, M. C. (1945). *Some generalized hypergeometric polynomials*. Ph. D. dissertation, University of Michigan.

Fasenmyer, M. C. (1947). Some generalized hypergeometric polynomials. *Bulletin of the American Mathematical Society*, **53**:806–812.

Fasenmyer, M. C. (1949). A note on pure recurrence relations. *American Mathematical Monthly*, **56**:14–17.

Faugère, J.-C., Gianni, P., Lazard, D., Mora, T. (1993). Efficient computation of zero-dimensional Gröbner bases by change of ordering. *Journal of Symbolic Computation*, **16**:329–344.

Galligo, A. (1985). Some algorithmic questions on ideals of differential operators. In Caviness, B. F., editor, *Proceedings EUROCAL'85*, volume 204 of *Lecture Notes in Computer Science*, pages 413–421. Springer Verlag.

Gasper, G., Rahman, M. (1990). *Basic Hypergeometric Series*, volume 35 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press.

Giovini, A., Mora, T., Niesi, G., Robbiano, L., Traverso, C. (1991). "One sugar cube, please" or selection strategies in the Buchberger algorithm. In Watt, S. M., editor, *Symbolic and Algebraic Computation*, pages 49–54. ACM Press.

Kandri-Rody, A., Weispfenning, V. (1990). Non-commutative Gröbner bases in algebras of solvable type. *Journal of Symbolic Computation*, **9**:1–26.

Kashiwara, M. (1978). On the holonomic systems of linear differential equations II. *Inventiones Mathematicae*, **49**:121–135.

Koornwinder, T. H. (1993). On Zeilberger's algorithm and its q-analogue. *Journal of Computational and Applied Mathematics*, **48**:91–111.

Kredel, H. (1993). *Solvable Polynomial Rings*. Reihe Mathematik. Verlag Shaker, Aachen, Germany. after the Ph. D. thesis, ISBN 3-86111-342-2.

Lipshitz, L. (1988). The diagonal of a D-finite power series is D-finite. *Journal of Algebra*, **113**:373–378.

Lipshitz, L. (1989). D-finite power series. *Journal of Algebra*, **122**:353–373.

Mora, T. (1994). An introduction to commutative and noncommutative Gröbner bases. *Theoretical Computer Science*, **134**:131–173.

Ore, O. (1933). Theory of non-commutative polynomials. *Annals of Mathematics*, **34**:480–508.

Parnes, S., Ekhad, S. B. (1992). A WZ-style proof of Jacobi polynomials' generating function. *Discrete Mathematics*, **110**:263–264.

Paule, P. (1994). Short and easy computer proofs of the Rogers-Ramanujan identities and of identities of similar type. *The Electronic Journal of Combinatorics*, **1**(R10):1–9.

Paule, P., Riese, A. (1996). A Mathematica q-analogue of Zeilberger's algorithm based on an algebraically motivated approach to q-hypergeometric telescoping. In *Fields Proceedings of the Workshop "Special Functions, q-Series and Related Topics", 12–23 June 1995*, Toronto, Ontario. Fields Institute for Research in Mathematical Sciences at University College. To appear.

Paule, P., Schorn, M. (1995). A Mathematica version of Zeilberger's algorithm for proving binomial coefficient identities. *Journal of Symbolic Computation*, **20**:673–698.

Petkovšek, M., Wilf, H., Zeilberger, D. (1996). *A=B*. A. K. Peters, Ltd., Wellesley, Massachusset. ISBN 1-56881-063-6.

Riese, A. (1996). A generalization of Gosper's algorithm to bibasic hypergeometric summation. *The Electronic Journal of Combinatorics*, **3**(R19):1–16.

Sabbah, C. (1993). Systèmes holonomes d'équations aux q-différences. In Kashiwara, M., Monteiro-Fernandes, T., Schapira, P., editors, *D-Modules and Microlocal Geometry*, pages 125–147, Berlin. Walter de Gruyter & Co. Proceedings of the Conference D-Modules and Microlocal Geometry, Lisbon, 1990.

Salvy, B., Zimmermann, P. (1994). Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable. *ACM Transactions on Mathematical Software*, **20**(2):163–177. Also available as Technical Report 143, INRIA.

Stanley, R. P. (1980). Differentiably finite power series. *European Journal of Combinatorics*, **1**:175–188.

Strehl, V. (1994). Binomial identities — combinatorial and algorithmic aspects. *Discrete Mathematics*, **136**:309–346.

Takayama, N. (1989). Gröbner basis and the problem of contiguous relations. *Japan Journal of Applied Mathematics*, **6**(1):147–160.

Takayama, N. (1990a). An algorithm of constructing the integral of a module — an infinite dimensional analog of Gröbner basis. In *Symbolic and Algebraic Computation*, pages 206–211. ACM and Addison-Wesley. Proceedings of ISSAC'90, Kyoto.

Takayama, N. (1990b). Gröbner basis, integration and transcendental functions. In *Symbolic and Algebraic Computation*, pages 152–156. ACM and Addison-Wesley. Proceedings ISSAC'90, Kyoto.

Takayama, N. (1992). An approach to the zero recognition problem by Buchberger algorithm. *Journal of Symbolic Computation*, **14**:265–282.

Takayama, N. (1995). Private communication.

Takayama, N. (1996). Algorithm finding recurrence relations of binomial sums and its complexity. *Journal of Symbolic Computation*, **20**(5 and 6):637–651.

Weispfenning, V. (1992). Finite Gröbner bases in non-Noetherian skew polynomial rings. In Wang, P. S., editor, *Symbolic and Algebraic Computation*, pages 329–334, New York. ACM Press. Proceedings ISSAC'92, July 1992, Berkeley, California.

Wilf, H. S., Zeilberger, D. (1992a). An algorithmic proof theory for hypergeometric (ordinary and "q") multisum/integral identities. *Inventiones Mathematicae*, **108**:575–633.

Wilf, H. S., Zeilberger, D. (1992b). Rational function certification of multisum/integral/"q" identities. *Bulletin of the American Mathematical Society*, **27**(1):148–153.

Zeilberger, D. (1990a). A fast algorithm for proving terminating hypergeometric identities. *Discrete Mathematics*, **80**:207–211.

Zeilberger, D. (1990b). A holonomic systems approach to special functions identities. *Journal of Computational and Applied Mathematics*, **32**:321–368.

Zeilberger, D. (1991a). A Maple program for proving hypergeometric identities. *SIGSAM Bulletin*, **25**(3):4–13.

Zeilberger, D. (1991b). The method of creative telescoping. *Journal of Symbolic Computation*, **11**:195–204.