Algorithmic Work with Orthogonal Polynomials and Special Functions

Wolfram Koepf

Konrad-Zuse-Zentrum für Informationstechnik Berlin, Heilbronner Str. 10, D-10711 Berlin, Federal Republic of Germany

Abstract:

In this article we present a method to implement orthogonal polynomials and many other special functions in Computer Algebra systems enabling the user to work with those functions appropriately, and in particular to verify different types of identities for those functions. Some of these identities like differential equations, power series representations, and hypergeometric representations can even dealt with algorithmically, i. e. they can be computed by the Computer Algebra system, rather than only verified.

The types of functions that can be treated by the given technique cover the generalized hypergeometric functions, and therefore most of the special functions that can be found in mathematical dictionaries.

The types of identities for which we present verification algorithms cover differential equations, power series representations, identities of the Rodrigues type, hypergeometric representations, and algorithms containing symbolic sums.

The current implementations of special functions in existing Computer Algebra systems do not meet these high standards as we shall show in examples. They should be modified, and we show results of our implementations.

1 Introduction

Many special functions can be looked at from the following point of view: They represent functions $f(n,x)$ of one "discrete" variable $n \in D$ defined on a set D that has the property that $n \in D \Rightarrow n+1 \in D$ (or $n \in D \Rightarrow n-1 \in D$), e. g. $D = \mathbb{N}_0, \mathbb{Z}, \mathbb{R}$, or $\mathbb{C}$, and one "continuous" variable $x \in I$ where I represents a real interval, either finite $I=[a,b]$, infinite ($I=[a,\infty)$, $I=(-\infty,a]$, or $I=\mathbb{R}$), or a subset of the complex plane $\mathbb{C}$.

In the given situation we may speak of the family $(f_n)_{n\in D}$ of functions $f_n(x) := f(n,x)$.

In this paper we will deal with special functions and orthogonal polynomials of a real/complex variable x. Many of our results can be generalized to special and orthogonal functions of a discrete variable x which we will consider in a forthcoming paper.

Many of those families, especially all families of orthogonal polynomials, have the following properties:

1. **(Derivative rule)**
   The functions f_n are differentiable with respect to the variable x, and satisfy a derivative rule of the form

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

   where the derivative with respect to x is represented by a finite number of lower or higher indexed functions of the family, and where r_k are rational functions in x. If $r_{m-1}(n,x) \not\equiv 0$ then the number m is called the order of the given derivative rule. We call the two different types of derivative rules backward and forward derivative rule, respectively.

2. **(Differential equation)**
   The functions f_n are m times differentiable ($m \in \mathbb{N}$) with respect to the variable x, and satisfy a homogeneous linear differential equation

$$\sum_{k=0}^{m} p_k(n,x)\, f_n^{(k)}(x) = 0 \;, \tag{2}$$

   where p_k are polynomials in x. If $p_m(n,k) \not\equiv 0$ then the number m is called the order of the given differential equation.

3. **(Recurrence equation)**
   The functions f_n satisfy a homogeneous linear recurrence equation with respect to n

$$\sum_{k=0}^{m} q_k(n,x)\, f_{n-k}(x) = 0 \;, \tag{3}$$

   where q_k are polynomials in x, and $m \in \mathbb{N}$. If $q_0(n,k), q_m(n,k) \not\equiv 0$ then the number m is called the order of the given recurrence equation.

Some of those families, especially all "classical" families of orthogonal polynomials, have the following further property:

4. **(Rodrigues representation)**
   The functions f_n have a representation of the Rodrigues type

$$f_n(x) = \frac{1}{K_n\, g(x)} \frac{\partial^n}{\partial x^n} h_n(x) \tag{4}$$

   for some functions g depending on x, and h_n depending on n and x, and a constant K_n depending on n.

From an algebraic point of view these properties read as follows: Let $K[x]$ denote the field of rational functions over K where K is one of $\mathbb{Q}$, $\mathbb{R}$, or $\mathbb{C}$. Then if the coefficients of the occurring polynomials and rational functions are elements of K,

1. the derivative rule states that f_n' is an element of the linear space over $K[x]$ which is generated by $\{f_n, f_{n-1}, \ldots, f_{n-(m-1)}\}$ or $\{f_n, f_{n+1}, \ldots, f_{n+m-1}\}$, respectively;

2. the differential equation states that the $m+1$ functions $f_n^{(k)}$ $(k=0,\ldots,m)$ are linearly dependent over $K[x]$; moreover, by an induction argument, any $m+1$ functions $f_n^{(k)}$ $(k\in\mathbb{N}_0)$ are linearly dependent over $K[x]$;

3. the recurrence equation states that the $m+1$ functions f_{n-k} $(k=0,\ldots,m)$ are linearly dependent over $K[x]$; moreover, by an induction argument, any $m+1$ functions f_n $(n\in D)$, are linearly dependent over $K[x]$.

One important question when dealing with special functions is the following: Which properties of those functions does one have to know to be able to establish various types of identities that those functions satisfy? With respect to the implementation of special functions in Computer Algebra systems this question reads: Which properties should be implemented for those functions, and in which form should this be done such that the user is enabled to verify various types of identities, or at least to implement algorithms for this purpose?

Nikiforov and Uvarov [18] gave a unified introduction to special functions of mathematical physics based primarily on the Rodrigues formula and the differential equation. They dealt, however, only with second order differential equations, which makes their treatment quite restricted, and moreover their development does not have algorithmic applications.

Truesdell [25] gave a unified approach to special functions based entirely on a special form of the derivative rule. His development has some algorithmic content, which, however, is difficult or impossible to implement in Computer Algebra. Truesdell's approach—although nice—has the further disadvantage that one can obtain only results of a very special form, see [13].

From the algorithmic point of view another approach is better: We will base our treatment of special functions on the derivative rule (1) in combination with the recurrence equation (3). We will show that an implementation of special functions in Computer Algebra systems based on these two properties gives a simplification mechanism at hand which, in particular, enables the user to verify many kinds of identities for those functions. Some of these identities like differential equations, and power series representations can even be dealt with algorithmically, i. e. they can be computed by the Computer Algebra system.

Our treatment is connected with the holonomic system approach due to Zeilberger [27]–[29] which is based on the valididy of partial differential equations, mixed recurrence equations, and difference-differential equations. This connection will be made more precise later.

The class of functions that can be treated this way contains the Airy functions $\mathrm{Ai}\,(x)$, $\mathrm{Bi}\,(x)$ (see e. g. [2], § 10.4), the Bessel functions $J_n(x), Y_n(x), I_n(x)$, and $K_n(x)$ (see e. g. [2], Ch. 9–11), the Hankel functions $H_n^{(1)}(x)$ and $H_n^{(2)}(x)$ (see e. g. [2], Ch. 9), the Kummer functions $M(a,b,x) = {}_1F_1\left(\begin{array}{c} a \\ b \end{array}\middle|\, x\right)$ and $U(a,b,x)$ (see e. g. [2], Ch. 13), the Whittaker functions $M_{n,m}(x)$ and $W_{n,m}(x)$ (see e. g. [2], § 13.4), the associated Legendre functions $P_a^b(x)$ and $Q_a^b(x)$ (see e. g. [2], § 8), all kinds of orthogonal polynomials: the Jacobi polynomials $P_n^{(\alpha,\beta)}(x)$, the Gegenbauer polynomials $C_n^{(\alpha)}(x)$, the Chebyshev polynomials of the first kind $T_n(x)$ and of the second kind $U_n(x)$, the Legendre polynomials $P_n(x)$, the Laguerre polynomials $L_n^{(\alpha)}(x)$, and the Hermite polynomials $H_n(x)$ (see [23], [24], and [2], § 22), many more special functions, and furthermore sums, products, derivatives, antiderivatives, and the composition with rational functions and rational powers of those functions (see [22], [27], [21] and [15]).

In the case of the classical orthogonal polynomials the properties above can be made much more precise (see e. g. [24], Kapitel IV). Therefore let $f_n : [a,b] \to \mathbb{R}$ $(n\in\mathbb{N}_0)$ denote the family of orthogonal polynomials

$$f_n(x) = k_n\, x^n + k_n'\, x^{n-1} + \ldots$$

with respect to the weight function $w(x) \geq 0$, i. e. with the property that

$$\int\limits_a^b w(x)\, f_n(x)\, f_m(x)\, dx = 0 \qquad (n \neq m)$$

and

$$\int\limits_a^b w(x)\, f_n^2(x)\, dx = h_n \neq 0 \; .$$

Then we have the properties:

1. **(Derivative rule)**
   The functions f_n satisfy a derivative rule of the form

   $$X\, f_n' = \beta_n\, f_{n-1} + \left(\frac{n}{2} X'' x + \alpha_n\right)\, f_n$$

   (see e. g. [24], p. 135, formula (4.8)) where

   $$\alpha_n = n\, X'(0) - \frac{1}{2}\, X''\, \frac{k_n'}{k_n} \; , \qquad \beta_n = -\frac{h_n\, k_{n-1}}{h_{n-1}\, k_n} \left(K_1\, k_1 - \frac{2n-1}{2}\, X''\right) \; ,$$

   and

   $$X(x) = \begin{cases} (b-x)(x-a) & \text{if } a, b \text{ are finite} \\ x-a & \text{if } b = \infty \\ 1 & \text{if } -a, b = \infty \end{cases} \; . \tag{5}$$

   Especially is the order of the derivative rule 2.

2. **(Differential equation)**
   The functions f_n satisfy the homogeneous linear differential equation with polynomial coefficients

   $$X\, f_n''(x) + K_1\, f_1\, f_n'(x) + \lambda_n\, f_n(x) = 0$$

   (see e. g. [24], p. 133, formula (4.1)) where

   $$\lambda_n = -n \left(K_1\, k_1 - \frac{n-1}{2}\, X''\right) \; ,$$

   and $X(x)$ is given by (5). Especially is the order of the differential equation 2.

3. **(Recurrence equation)**
   The functions f_n satisfy the recurrence equation

   $$f_{n+1}(x) = -C_n\, f_{n-1}(x) + (A_n\, x + B_n)\, f_n(x) \tag{6}$$

   (see e. g. [24], p. 126, formula (2.1)) with

   $$A_n = \frac{k_{n+1}}{k_n} \; , \qquad B_n = \frac{k_{n+1}}{k_n} \left(\frac{k_{n+1}'}{k_{n+1}} - \frac{k_n'}{k_n}\right) \; , \qquad \text{and} \qquad C_n = \frac{k_{n+1}\, k_{n-1}\, h_n}{k_n^2\, h_{n-1}} \; .$$

   Especially is the order of the recurrence equation 2.

4. **(Rodrigues representation)**
   The functions f_n have a representation of the Rodrigues type

$$f_n(x) = \frac{1}{K_n\, w(x)} \frac{\partial^n}{\partial x^n} \Big(w(x)\, X(x)^n \Big) \tag{7}$$

   (see e. g. [24], p. 129, formula (3.2)), where $X(x)$ is given by (5), i. e. (4) is valid with $g(x) = w(x)$, and $h_n(x) = w(x)\, X(x)^n$. Especially: The order of the polynomial $X(x)$ is ≤ 2.

Further it turns out that in the case of classical orthogonal polynomials all coefficient functions of f_{n-k} are rational also with respect to the variable n, a fact that depends, however, on the special normalizations that are used in these cases.

We mention that no system of orthogonal polynomials besides the classical ones satisfies a Rodrigues representation of type (7) with a polynomial X (see e. g. [24], Kapitel IV, §3).

We note that using the recurrence equation (6), which is valid also for non-classical orthogonal polynomials, or any recurrence equation of type (3) of order two (also called three-term recursion), recursively, each (backward or forward) derivative rule (1) is equivalent to a derivative rule

$$f_n'(x) = k(n,x)\, f_n(x) + l(n,x)\, f_{n+1}(x) \tag{8}$$

(k, l rational functions with respect to x) of order two. In general, the order of the derivative rule can always be assumed to be less than or equal to the order of the recurrence equation. In some nice work [25] Truesdell presented a treatment of special functions entirely based on the functional equation (8). He showed that this difference-differential equation is independent of the differential equation (2) and the recurrence equation (3), i. e. it does not imply the existence of one of these.

In contrast to this work, our main notion is the

Definition (Admissible family of special functions) We call a family f_n of special functions admissible if the functions f_n satisfy a recurrence equation of type (3) and a derivative rule of type (1). We call the order of the recurrence equation the order of the admissible family f_n. □

Note that the recurrence equation (3) together with m initial functions $f_{n_0}, f_{n_0+1}, \ldots, f_{n_0+m-1}$ determine the functions f_n $(n \in D)$ uniquely.

So an admissible family of special functions (with given initial functions) is overdetermined by its two defining properties, i. e. the recurrence equation and the derivative rule must be compatible. This fact, however, gives our notion a considerable strength:

Theorem 1 For any admissible family f_n of order m the linear space V_{f_n} over $K[x]$ of functions generated by the set of shifted derivatives $\{f_{n\pm k}^{(j)} \mid j, k \in \mathbb{N}_0\}$ is at most m-dimensional. On the other hand, if the family $\{f_{n\pm k}^{(j)} \mid j, k \in \mathbb{N}_0\}$ spans an m-dimensional linear space, then f_n forms an admissible family of order m.

Proof: By the recurrence equation and an induction argument it follows that the linear space V spanned by $\{f_{n\pm k} \mid k \in \mathbb{N}_0\}$ is at most m-dimensional. Using the derivative rule, by a further induction it follows that the derivative of any order $f_n^{(k)}$ $(k \in \mathbb{N}_0)$ is an element of V. Therefore $V_{f_n} = V$.

If on the other hand for a family f_n the set of derivatives $\{f_{n\pm k}^{(j)} \mid j, k \in \mathbb{N}_0\}$ is m-dimensional, then the existence of a recurrence equation and a derivative rule of order m are obvious. □

From the algebraic point of view this is the main reason for the importance of admissible families: Any $m+1$ distinguished elements of V_{f_n} are linearly dependent, i. e. any arbitrary element of V_{f_n} can be represented by a linear combination (with respect to $K[x]$) of any m of the others. This is the algebraic background for the fact that so many identities between the members and their derivatives of an admissible family exist.

In particular we have

Corollary 1 Any admissible family f_n of order m satisfies a simple differential equation of order m. □

In § 8 we give an algorithm which, in particular, generates this differential equation of f_n.

With regard to Zeilberger's approach Corollary 1 can be interpreted as follows: Any admissible family $f_n(x)$ forms a holonomic system with respect to the two variables n, and x, whose defining recurrence equation, and the differential equation corresponding to Corollary 1 together with the initial conditions

$$f_0^{(k)}(0)\ , \qquad \text{and} \qquad f_k(0) \qquad (k=0,\ldots,m-1) \tag{9}$$

yield the canonical holonomic representation of $f_n(x)$ (see [27], Lemma 4.1).

On the other hand, not all holonomic systems $f_n(x)$ form admissible families so that our notion is stronger: Let $f_n(x) := \text{Ai}\,(x)$ for all $n \in \mathbb{Z}$, then obviously $f_n(x)$ is the holonomic system generated by the equations

$$f_n''(x) = x\, f_n(x)\ , \qquad f_{n+1}(x) = f_n(x)\ ,$$

and some initial values, that does not form an admissible family as the derivative f_n' is linearly independent of $\{f_n \mid n \in \mathbb{Z}\}$ over $K[x]$, see § 5, and thus no derivative rule of the form (1) exists.

A further advantage of our approach is the separation of the variables, i. e. the work with ordinary differential equations, and one-variable recurrence equations rather than partial differential equations, mixed recurrence equations, and difference-differential equations. So our approach—if applicable—seems to be more natural.

To present an example of an admissible family that cannot be found in mathematical dictionaries, we consider the functions

$$k_n(x) := \frac{2}{\pi} \int\limits_0^{\pi/2} \cos\left(x\, \tan\theta - n\,\theta\right) d\theta\ ,$$

that Bateman introduced in [4], see also [14]. He verified that ([4], formula (2.7))

$$F_n(x) := (-1)^n\, k_{2n}(x) = (-1)^n\, e^{-x} \Big(L_n(2x) - L_{n-1}(2x) \Big)\ . \tag{10}$$

We call F_n the family of Bateman functions which turns out to be an admissible family of order two.

Bateman obtained the property ([4], formula (4.1))

$$(n-1)\Big(F_n(x) - F_{n-1}(x)\Big) + (n+1)\Big(F_n(x) - F_{n+1}(x)\Big) = 2\,x\,F_n(x)$$

leading to

$$n\,F_n(x) - 2\,(n-1-x)\,F_{n-1}(x) + (n-2)\,F_{n-2}(x) = 0 \tag{11}$$

which is a recurrence equation of type (3) and order two that determines the Bateman functions uniquely using the two initial functions

$$F_0(x) = e^{-x} \qquad \text{and} \qquad F_1(x) = -2\,x\,e^{-x}$$

which follow from (10).

Bateman obtained further a difference differential equation ([4], formula (4.2))

$$(n+1)\,F_{n+1}(x) - (n-1)\,F_{n-1}(x) = 2\,x\,F_n'(x)\;, \tag{12}$$

which can be brought into the form

$$F_n'(x) = \frac{1}{x}\Big((n-x)\,F_n(x) - (n-1)\,F_{n-1}(x)\Big) \tag{13}$$

using (11). This is a derivative rule of the form (1) and order two. Therefore $F_n(x)$ form an admissible family of order two.

We note that the functions F_n satisfy the differential equation

$$x\,F_n''(x) + (2n-x)\,F_n(x) = 0\;, \tag{14}$$

(see [4], formula (5.1)), and the Rodrigues type representation

$$F_n(x) = \frac{x\,e^x}{n!}\frac{d^n}{dx^n}\left(e^{-2x}\,x^{n-1}\right)\;, \tag{15}$$

(see [4], formula (31)).

2 Properties of admissible families

Theorem 2 Let f_n form an admissible family of order m. Then

(a) **(Shift)** $f_{n\pm k}$ $(k \in \mathbb{N})$ forms an admissible family of order m;

(b) **(Derivative)** f_n' forms an admissible family of order $\leq m$;

(c) **(Composition)** $f_n \circ r$ forms an admissible family of order $\leq m$, if r is a rational function, and of order $\leq m\,q$, if $r(x) = x^{p/q}$ $(p, q \in \mathbb{N})$.

If furthermore g_n forms an admissible family of order $\leq l$, then moreover

(d) **(Sum)** $f_n + g_n$ forms an admissible family of order $\leq m + l$;

(e) **(Product)** $f_n\,g_n$ forms an admissible family of order $\leq m\,l$.

Proof: (a): This is an obvious consequence of Theorem 1.
(b): Let $g_n := f_n'$. We start with the recurrence equation for f_n and take derivative to get

$$\sum_{k=0}^{m} q_k'(n,x)\,f_{n-k}(x) + \sum_{k=0}^{m} q_k(n,x)\,f_{n-k}'(x) = 0\;. \tag{16}$$

From Theorem 1, we know that each of the functions f_{n-j} $(j = 0, \ldots, m)$ can be represented as a linear combination of the functions f'_{n-k} $(k = 0, \ldots, m-1)$ over $K[x]$, which generates a recurrence equation for g_n. Similarly a derivative rule for g_n is obtained.

(c): For the composition $h_n := f_n \circ r$ with a rational function r, the recurrence equation is obtained by substitution, and the derivative rule is a result of the chain rule. If $r(x) = x^{1/q}$, then, by [15], Lemma 1, the family $\{h_n^{(j)} \mid j \in \mathbb{N}_0\}$ is spanned by the mq functions $x^{r/q} f_n^{(j)}(x^{1/q})$ $(j = 1, \ldots, m-1,$ $r = 0, \ldots, q-1)$, and since $\{f_{n\pm k}^{(j)} \mid j, k \in \mathbb{N}_0\}$ has dimension m, the linear space spanned by $\{h_{n\pm k}^{(j)} \mid j, k \in \mathbb{N}_0\}$ has dimension $\leq m\,q$, implying the result. If finally $r(x) = x^{p/q}$, then a combination gives the result.

(d): By a simple algebraic argument, we see that $f_{n-k} + g_{n-k}$ $(k \in \mathbb{Z})$ span the linear space $V := V_{f_n + g_n} = V_{f_n} + V_{g_n}$ of dimension $\leq m + l$ over $K[x]$. Therefore $f_n + g_n$ satisfies a recurrence equation of order $\leq m + l$. If we add the derivative rules for f_n and g_n, we see that $f'_n + g'_n \in V$, and thus can be represented in the desired way.

(e): By a similar algebraic argument (see e. g. [22], Theorem 2.3) we see that $f_{n-k} \cdot g_{n-k}$ $(k \in \mathbb{Z})$ span a linear space V of dimension $\leq m\,l$ over $K[x]$, hence $f_n\, g_n$ satisfies a recurrence equation of order $\leq m\,l$. By the product rule, and the derivative rules for f_n and g_n we see that the derivative of $f_n\, g_n$ is represented by products of the form $f_{n-k}\, g_{n-j}$ $(k, j \in \mathbb{Z})$, and as those span the linear space V (see e. g. [15], Theorem 3 (d)), we are done. □

As an application we again may state that the Bateman functions form an admissible family: Using the theorem, this follows immediately from representation (10).

Next we study algorithmic versions of the theorem. The following algorithm generates a representation of the members $f_{n\pm k}$ $(k = 0, \ldots, m-1)$ of an admissible family in terms of the derivatives $f'_{n\pm j}$ $(j = 0, \ldots, m-1)$. By Theorem 1 we know that such a representation exists. Without loss of generality, we assume that the admissible family is given by a backward derivative rule. In case of a forward derivative rule, a similar algorithm is valid.

Algorithm 1 Let f_n be an admissible family of order m, given by a backward derivative rule

$$f'_n(x) = \sum_{k=0}^{m-1} r_k(n,x)\, f_{n-k}(x) \; .$$

Then the following algorithm generates a list of backward rules $(k = 0, \ldots, m-1)$

$$f_{n-k}(x) = \sum_{j=0}^{m-1} R_j^k(n,x)\, f'_{n-j}(x) \tag{17}$$

(R_j^k rational with respect to x) for f_{n-k} $(k = 0, \ldots, m-1)$ in terms of the derivatives f'_{n-j} $(j = 0, \ldots, m-1)$:

(a) Shift the derivative rule $m-1$ times to obtain the set of m equations

$$f'_{n-j}(x) = \sum_{k=0}^{m-1} r_k(n-j,x)\, f_{n-j-k}(x) \qquad (j = 0, \ldots, m-1) \; .$$

(b) Utilize the recurrence equation to express all expressions on the right hand sides of these equations in terms of f_{n-k} $(k = 0, \ldots, m-1)$ leading to

$$f'_{n-j}(x) = \sum_{k=0}^{m-1} r_k^j(n,x)\, f_{n-k}(x) \qquad (j = 0, \ldots, m-1\ ,\ r_k^j \text{ rational with respect to } x)\ .$$

(c) Solve this linear equations system for the variables f_{n-k} $(k = 0, \ldots, m-1)$ to obtain the representations (17) searched for. □

The proof of the algorithm is obvious. It is also clear how the method can be adapted to obtain forward rules in terms of the derivatives. As an example, the algorithm generates the following representations for the Bateman functions

$$F_n(x) = \frac{1-n+x}{2\,n-1-x}\, F'_n(x) + \frac{n-1}{2\,n-1-x}\, F'_{n-1}(x)\ ,$$

and

$$F_n(x) = \frac{1+n-x}{1+2\,n-x}\, F'_n(x) - \frac{1+n}{1+2\,n-x}\, F'_{n+1}(x)$$

in terms of their derivatives.

We note that by means of Algorithm 1 and the results of [15] (see also [27], p. 342, and [21]), we are able to state algorithmic versions of the statements of Theorem 2.

Algorithm 2 The following algorithms lead to the derivative rules and recurrence equations of the admissible families presented in Theorem 2:

(a) **(Shift)** Direct use of derivative rule and recurrence equation lead to the derivative rule and the recurrence equation for $f_{n\pm1}$; a recursive application gives the results for $f_{n\pm k}$ $(k \in \mathbb{N})$.

(b) **(Derivative)** By Algorithm 1 we may replace all occurrences of f_{n-k} $(k = 0, \ldots, m)$ in (16), resulting in the recurrence equation for f'_n; similarly the derivative rule is obtained.

(c) **(Composition)** If r is a rational function, then an application of the chain rule leads to the derivative rule and the recurrence equation of $f_n \circ r$; an approach similar to the algorithmic version of Theorem 2 in [15] yields the derivative rule and the recurrence equation of $f_n \circ x^{1/q}$ by an elimination of the expressions $x^{r/q}\, f_n^{(j)}(x^{1/q})$ $(r = 1, \ldots, q-1,\ j = 1, \ldots, m-1)$.

(d) **(Sum)** Applying a discrete version of Theorem 3 (c) in [15] to $f_n + g_n$ (see also [27], p. 342, and [21], Maple function `rec+rec`) results in the recurrence equation, and a similar approach gives the derivative rule.

(e) **(Product)** Applying a discrete version of Theorem 3 (d) in [15] to $f_n\, g_n$ (see also [27], p. 342, and [21], Maple function `rec*rec`) yields the recurrence equation, and a similar approach gives the derivative rule. □

A Mathematica implementation of the given algorithms generate e. g. for the derivative $F'_n(x)$ of the Bateman function $F_n(x)$ the derivative rule

$$F''_n(x) = \frac{2\,n-x}{x-2\,n\,x+x^2}\Big(\,(n-1)\,F'_{n-1}(x) + (1-n+x)\,F'_n(x)\Big)\ ,$$

and the recurrence equation

$$F'_{n+1}(x) = \frac{1}{(1+n)(1-2n+x)}\Big((n-1)(x-2n-1)F'_{n-1}(x) + 2\,(1-2n^2+3nx-x^2)F'_n(x)\Big)\ ,$$

and for the product $A_n(x) := F_n^2(x)$ the derivative rule

$$\begin{aligned} A'_n(x) \quad &= \quad \frac{(1-n)\ (n-2)^2}{2\,n\,x\ (1-n+x)}\,A_{n-2}(x) \\ &+ \frac{2\ (n-1)\ (1-n+x)}{n\,x}\,A_{n-1}(x) \\ &+ \frac{(3\,n-3\,n^2-4\,x+8\,n\,x-4\,x^2)}{2\,x\ (1-n+x)}\,A_n(x)\ , \end{aligned}$$

and the recurrence equation

$$\begin{aligned} A_{n+1}(x) \quad = \quad &\frac{1}{(1+n)^2}\left(\frac{(n-2)^2\,(n-1)\,(x-n)}{n\,(1-n+x)}\,A_{n-2}\right. \\ &+ \frac{(n-1)\,(3\,n-3\,n^2-4\,x+8\,n\,x-4\,x^2)}{n}\,A_{n-1} \\ &+ \left.\frac{(x-n)\,(-3\,n+3\,n^2+4\,x-8\,n\,x+4\,x^2)}{1-n+x}\,A_n\right) \end{aligned}$$

are derived.

3 Derivative rules of special functions

Many Computer Algebra systems like AXIOM [3], MACSYMA [16], MAPLE [17], MATHEMATICA [26], or REDUCE [8] support the work with special functions. On the other hand, there are so many identities for special functions that it is a nontrivial task to decide which properties should be used by the system (and in which way) for the work with those functions.

Since all Computer Algebra systems support derivatives, as a first question it is natural to ask how the current implementations of Computer Algebra systems handle the derivatives of special functions. Here are some examples: MATHEMATICA (Version 2.2) gives

```
In[1]:= D[BesselI[n,x],x]

        BesselI[-1 + n, x] + BesselI[1 + n, x]
Out[1]= --------------------------------------
                          2
In[2]:= D[LaguerreL[n,a,x],x]

Out[2]= -LaguerreL[-1 + n, 1 + a, x]
```

We note that in MATHEMATICA the derivatives of all special functions symbolically are implemented. On the other hand, we notice that, given the function $I_n\,(x)$, MATHEMATICA's derivative introduces two new functions: $I_{n-1}\,(x)$, and $I_{n+1}\,(x)$. Given the Laguerre polynomial $L_n^{(\alpha)}(x)$, the derivative produced introduces a new function where both n, and α are altered. The representation used is optimal for numerical purposes, but is not a representation according to our classification.

With Maple (Version V.2) we get

```
> diff(BesselI(n,x),x);
                                          n BesselI(n, x)
                   BesselI(n + 1, x) + ---------------
                                                x

> diff(L(n,a,x),x);
                               d
                              ---- L(n, a, x)
                               dx
```

Thus Maple's derivative for the Bessel function $I_n(x)$ introduces only one new function $I_{n+1}(x)$, and is of type (1), whereas (even if `orthopoly` is loaded) no symbolic derivative of the Laguerre polynomial $L_n^{(\alpha)}(x)$ is implemented.

Obviously there is no unique way to declare the derivative of a special function. However, we note that if we declare the derivative of a special function by a derivative rule of type (1) of order m then we can be sure that the derivative of the special function $f_n(x)$ introduces at most m new functions, namely $f_{n-k}(x)$ $(k = 1, \ldots, m)$. Moreover, if the family of special functions depends on several parameters, then the given representation of the derivative does not use any functions with other parameters changed.

Here we give a list of the backward derivative rules of the form (1) for the families of special functions that we introduced in § 1 which all turn out to be of order two (see e. g. [2], (9.1.27) (Bessel and Hankel functions), (9.2.26) (Bessel functions), (13.4.11), (13.4.26) (Kummer functions), (13.4.29)–(13.4.33) (Whittaker functions), (8.5.4) (associated Legendre functions), and § 22.8 (orthogonal polynomials)):

$$\begin{aligned}
J_n'(x) &= J_{n-1}(x) - \frac{n}{x} J_n(x) \,, \\
Y_n'(x) &= Y_{n-1}(x) - \frac{n}{x} Y_n(x) \,, \\
I_n'(x) &= I_{n-1}(x) - \frac{n}{x} I_n(x) \,, \\
K_n'(x) &= -K_{n-1}(x) - \frac{n}{x} K_n(x) \,, \\
\frac{\partial}{\partial x} H_n^{(1)}(x) &= H_{n-1}^{(1)}(x) - \frac{n}{x} H_n^{(1)}(x) \,, \\
\frac{\partial}{\partial x} H_n^{(2)}(x) &= H_{n-1}^{(2)}(x) - \frac{n}{x} H_n^{(2)}(x) \,, \\
\frac{\partial}{\partial x} M(a,b,x) &= \frac{1}{x}\Big((b-a)\,M(a-1,b,x) - (b-a-x)\,M(a,b,x)\Big) \,, \\
\frac{\partial}{\partial x} U(a,b,x) &= \frac{1}{x}\Big(-U(a-1,b,x) + (a-b+x)\,U(a,b,x)\Big) \,,
\end{aligned}$$

$$
\begin{aligned}
M'_{n,m}(x) &= \frac{1}{2x}\Big((1+2m-2n)\,M_{n-1,m}(x) + (2n-x)\,M_{n,m}(x)\Big)\;, \\
W'_{n,m}(x) &= \frac{1}{4x}\Big((1-4m^2-4n+4n^2)\,W_{n-1,m}(x) + (4n-2x)\,W_{n,m}(x)\Big)\;, \\
\frac{\partial}{\partial x}P_a^b(x) &= \frac{1}{1-x^2}\Big((a+b)\,P_{a-1}^b(x) - a\,x\,P_a^b(x)\Big)\;, \\
\frac{\partial}{\partial x}Q_a^b(x) &= \frac{1}{1-x^2}\Big((a+b)\,Q_{a-1}^b(x) - a\,x\,Q_a^b(x)\Big)\;, \\
\frac{\partial}{\partial x}P_n^{(\alpha,\beta)}(x) &= \frac{1}{(2n+\alpha+\beta)(1-x^2)}\Big(2(n+\alpha)(n+\beta)P_{n-1}^{(\alpha,\beta)}(x) + n(\alpha-\beta-(2n+\alpha+\beta)x)P_n^{(\alpha,\beta)}(x)\Big), \\
\frac{\partial}{\partial x}C_n^{(\alpha)}(x) &= \frac{1}{1-x^2}\Big((n+2\alpha-1)\,C_{n-1}^{(\alpha)}(x) - n\,x\,C_n^{(\alpha)}(x)\Big)\;, \\
T'_n(x) &= \frac{1}{1-x^2}\Big(n\,T_{n-1}(x) - n\,x\,T_n(x)\Big)\;, \\
U'_n(x) &= \frac{1}{1-x^2}\Big((n+1)\,U_{n-1}(x) - n\,x\,U_n(x)\Big)\;, \\
P'_n(x) &= \frac{1}{1-x^2}\Big(n\,P_{n-1}(x) - n\,x\,P_n(x)\Big)\;, \\
\frac{\partial}{\partial x}L_n^{(\alpha)}(x) &= \frac{1}{x}\Big(-(n+\alpha)\,L_{n-1}^{(\alpha)}(x) + n\,L_n^{(\alpha)}(x)\Big)\;, \qquad (18) \\
H'_n(x) &= 2n\,H_{n-1}(x)\;.
\end{aligned}
$$

4 Recurrence equations of special functions

Whenever in any expression subexpressions of the form $r_k\,f_{n-k}$ (r_k rational, $k \in \mathbb{Z}$) occur, in an admissible family of order m with the recursive use of the recurrence equation we may replace so many occurrences of those expressions $r_k\,f_{n-k}$ that finally only m successive terms of the same type remain.

This allows for example to eliminate the number of occurrences in any linear combination (over $K[x]$) of derivatives of f_n to m, a fact with which we will deal in more detail in § 8.

We show how Mathematica and Maple work with regard to this question. Whereas Mathematica does not have any built-in capabilities to simplify the following linear combinations of Bessel and Laguerre functions,

```
In[3]:= BesselI[n+1,x]+2*n/x*BesselI[n,x]-BesselI[n-1,x]

                                2 n BesselI[n, x]
Out[3]= -BesselI[-1 + n, x] + ----------------- + BesselI[1 + n, x]
                                        x
```

```
In[4]:= Simplify[%]
                                   2 n BesselI[n, x]
Out[4]= -BesselI[-1 + n, x] + ----------------- + BesselI[1 + n, x]
                                           x

In[5]:= LaguerreL[n+1,a,x]-(2*n+a+1-x)*LaguerreL[n,a,x]+(n+a)*LaguerreL[n-1,a,x]

Out[5]= (a + n) LaguerreL[-1 + n, a, x] -

>    (1 + a + 2 n - x) LaguerreL[n, a, x] + LaguerreL[1 + n, a, x]

In[6]:= Simplify[%]

Out[6]= (a + n) LaguerreL[-1 + n, a, x] -

>    (1 + a + 2 n - x) LaguerreL[n, a, x] + LaguerreL[1 + n, a, x]
```

with MAPLE we get

```
> BesselI(n+1,x)+2*n/x*BesselI(n,x)-BesselI(n-1,x);

                                    n BesselI(n, x)
          BesselI(n + 1, x) + 2 --------------- - BesselI(- 1 + n, x)
                                           x

> simplify(");
                                           0

> L(n+1,a,x)-(2*n+a+1-x)*L(n,a,x)+(n+a)*L(n-1,a,x);

     L(n + 1, a, x) - (2 n + a + 1 - x) L(n, a, x) + (n + a) L(n - 1, a, x)

> simplify(");

   L(n + 1, a, x) - 2 L(n, a, x) n - L(n, a, x) a - L(n, a, x) + L(n, a, x) x

        + L(n - 1, a, x) n + L(n - 1, a, x) a
```

i. e. MAPLE's `simplify` command supports simplification with the aid of the recurrence equations for the Bessel functions. On the other hand, for the orthogonal polynomials (even if `orthopoly` is loaded) no simplifications occur.

In the rest of this section we give a list of the recurrence equations of the given type for the families of special functions that we consider which all turn out to be of order two (see e. g. [2], (9.1.27), (9.2.26), (13.4.1), (13.4.15), (13.4.29), (13.4.31), (8.5.3), and § 22.7). We list them in the form explicitly solved for F_{n+1} as this is the usual form found in mathematical dictionaries.

$$
\begin{aligned}
J_{n+1}(x) &= -J_{n-1}(x) + \frac{2n}{x} J_n(x) , \\
Y_{n+1}(x) &= -Y_{n-1}(x) + \frac{2n}{x} Y_n(x) ,
\end{aligned}
$$

$$
\begin{aligned}
I_{n+1}\,(x) &= I_{n-1}\,(x) - \frac{2n}{x}\, I_n\,(x)\ , \\
K_{n+1}\,(x) &= K_{n-1}\,(x) + \frac{2n}{x}\, K_n\,(x)\ , \\
H^{(1)}_{n+1}\,(x) &= -H^{(1)}_{n-1}\,(x) + \frac{2n}{x}\, H^{(1)}_n\,(x)\ , \\
H^{(2)}_{n+1}\,(x) &= -H^{(2)}_{n-1}\,(x) + \frac{2n}{x}\, H^{(2)}_n\,(x)\ , \\
M(a+1,b,x) &= \frac{1}{a}\Big((b-a)\,M(a-1,b,x) + (2a-b+x)\,M(a,b,x)\Big)\ , \\
U(a+1,b,x) &= -\frac{1}{a\,(1+a-b)}\Big(U(a-1,b,x) + (b-2a-x)\,U(a,b,x)\Big)\ , \\
M_{n+1,m}\,(x) &= \frac{1}{1+2m+2n}\Big((1+2m-2n)\,M_{n-1,m}\,(x) + (4n-2x)\,M_{n,m}\,(x)\Big)\ , \\
W_{n+1,m}\,(x) &= \frac{1}{4}\Big((-1+4m^2+4n-4n^2)\,W_{n-1,m}\,(x) - (8n-4x)\,W_{n,m}\,(x)\Big)\ , \\
P^b_{a+1}\,(x) &= \frac{1}{a-b+1}\Big(-(a+b)\,P^b_{a-1}\,(x) + (2a+1)\,x\ P^b_{a-1}\,(x)\Big)\ , \\
Q^b_{a+1}\,(x) &= \frac{1}{a-b+1}\Big(-(a+b)\,Q^b_{a-1}\,(x) + (2a+1)\,x\ Q^b_{a-1}\,(x)\Big)\ , \\
P^{(\alpha,\beta)}_{n+1}(x) &= \frac{1}{2\,(n{+}1)\,(n{+}\alpha{+}\beta{+}1)\,(2n{+}\alpha{+}\beta)}\Big(-2(n{+}\alpha)(n{+}\beta)(2n{+}\alpha{+}\beta{+}2)\,P^{(\alpha,\beta)}_{n-1}(x) \\
&\quad + \Big((2n{+}\alpha{+}\beta{+}1)(\alpha^2-\beta^2) + (2n{+}\alpha{+}\beta)_3\,x\Big)\,P^{(\alpha,\beta)}_n(x)\Big)\ , \\
C^{(\alpha)}_{n+1}\,(x) &= \frac{1}{n+1}\Big(-(n+2\alpha-1)\,C^{(\alpha)}_{n-1}\,(x) + 2(n+\alpha)\,x\,C^{(\alpha)}_n\,(x)\Big)\ , \\
T_{n+1}\,(x) &= -T_{n-1}\,(x) + 2\,x\,T_n\,(x)\ , \\
U_{n+1}\,(x) &= -U_{n-1}\,(x) + 2\,x\,U_n\,(x)\ , \\
P_{n+1}\,(x) &= \frac{1}{n+1}\Big(-n\,P_{n-1}\,(x) + (2n+1)\,x\,P_n\,(x)\Big)\ , \\
L^{(\alpha)}_{n+1}(x) &= \frac{1}{n+1}\Big(-(n+\alpha)\,L^{(\alpha)}_{n-1}(x) + (2n+\alpha+1-x)\,L^{(\alpha)}_n(x)\Big)\ , \\
H_{n+1}(x) &= -2n\,H_{n-1}(x) + 2x\,H_n(x)\ .
\end{aligned}
$$

Note that $(a)_k$ (which is used in the recurrence equation for the Jacobi polynomials $P^{(\alpha,\beta)}_n(x)$)

denotes the Pochhammer symbol (or shifted factorial) defined by $(a)_k := \prod_{j=1}^{k} (a+j-1)$.

We note further that for functions with several "discrete" variables it may happen that for each of them there exists a recurrence equation. As an example we consider the Laguerre polynomials for which we have ([2] (22.7.29), in combination with (22.7.30))

$$L_n^{(\alpha+1)}(x) = \frac{1}{x}\left(-(n+\alpha)\, L_n^{(\alpha-1)}(x) + (\alpha+x)\, L_n^{(\alpha)}(x)\right) . \tag{19}$$

In § 7 we will demonstrate that generalized hypergeometric functions satisfy recurrence equations with respect to all their parameters.

To be safely enabled that the algorithms of § 9–§ 11 apply, all of those recurrence equations should be implemented and applied recursively for simplification purposes.

5 Embedding of one-variable functions into admissible families

In this section we consider first, how the elementary transcendental functions are covered by the given approach.

Consider the exponential function $f(x) = e^x$. This function can be embedded into the admissible family f_n, defined by the properties

$$f_n'(x) = f_n(x) , \qquad f_{n+1}(x) = f_n(x) \qquad \text{and} \qquad f_0(x) = e^x ,$$

i. e. the family of iterated derivatives of e^x.

Obviously this is a representation of an admissible family of order one.

Moreover in the given case it turns out that $f_n(x) = e^x = f_0(x)$ for all $n \in \mathbb{Z}$, so there is no actual need to give the functions numbers, and therefore we (obviously) keep the usual notation.

Similarly the functions $\sin x$ and $\cos x$ are embedded into the admissible family f_n of order two given by the properties

$$f_n'(x) = f_{n-1}(x) , \qquad f_{n+1}(x) = -f_{n-1}(x) , \qquad \text{and} \qquad f_0(x) = \cos x , \qquad f_1(x) = \sin x .$$

Again, the family of functions f_n is finite, and our numbering is unnecessary:

$$f_n(x) = \begin{cases} \cos x & \text{if } n = 4m \ (m \in \mathbb{Z}) \\ \sin x & \text{if } n = 4m+1 \ (m \in \mathbb{Z}) \\ -\cos x & \text{if } n = 4m+2 \ (m \in \mathbb{Z}) \\ -\sin x & \text{if } n = 4m+3 \ (m \in \mathbb{Z}) \end{cases} .$$

Essentially there are only the two functions $\cos x$, and $\sin x$ involved. Note, however, that both functions are needed as no simple first order differential equation for $\sin x$ or $\cos x$ exists.

Other nontrivial examples of essentially finite admissible families of special functions are formed by the Airy functions. Let $\mathrm{Ai}_n\,(x) = \mathrm{Ai}^{(n)}\,(x)$, i. e.

$$\mathrm{Ai}_n'\,(x) = \mathrm{Ai}_{n+1}\,(x) .$$

By the differential equation for the Airy functions (see e. g. [2], (10.4)) we have $\mathrm{Ai}''(x) - x\ \mathrm{Ai}(x) = 0$, so that from Leibniz's rule it follows that

$$
\begin{aligned}
\mathrm{Ai}_{n+1}(x) &= \mathrm{Ai}^{(n+1)}(x) = \Big(\mathrm{Ai}''(x)\Big)^{(n-1)} \\
&= \Big(x\ \mathrm{Ai}(x)\Big)^{(n-1)} = \sum_{k=0}^{n-1} \binom{n-1}{k} x^{(k)} \Big(\mathrm{Ai}(x)\Big)^{(n-1-k)} \\
&= x\ \mathrm{Ai}^{(n-1)}(x) + (n-1)\ \mathrm{Ai}^{(n-2)}(x) = x\ \mathrm{Ai}_{n-1}(x) + (n-1)\ \mathrm{Ai}_{n-2}(x)\,,
\end{aligned}
$$

and therefore $\mathrm{Ai}(x)$ is embedded into the admissible family Ai_n of order three given by

$$
\mathrm{Ai}_n'(x) = \mathrm{Ai}_{n+1}(x)\,, \qquad \mathrm{Ai}_{n+1}(x) = x\ \mathrm{Ai}_{n-1}(x) + (n-1)\ \mathrm{Ai}_{n-2}(x)\,, \tag{20}
$$

and we have the initial functions

$$
\mathrm{Ai}_0(x) = \mathrm{Ai}(x)\,, \qquad \mathrm{Ai}_1(x) = \mathrm{Ai}'(x) \qquad \text{and} \qquad \mathrm{Ai}_2(x) = x\ \mathrm{Ai}(x)\,.
$$

Similarly $\mathrm{Bi}(x)$ is embedded into the admissible family of order three given by

$$
\mathrm{Bi}_n'(x) = \mathrm{Bi}_{n+1}(x)\,, \qquad \mathrm{Bi}_{n+1}(x) = x\ \mathrm{Bi}_{n-1}(x) + (n-1)\ \mathrm{Bi}_{n-2}(x)\,, \tag{21}
$$

and the initial functions

$$
\mathrm{Bi}_0(x) = \mathrm{Bi}(x)\,, \qquad \mathrm{Bi}_1(x) = \mathrm{Bi}'(x) \qquad \text{and} \qquad \mathrm{Bi}_2(x) = x\ \mathrm{Bi}(x)\,.
$$

Our indexed families turn out to be representable by

$$
\mathrm{Ai}_n(x) = p_n(x)\ \mathrm{Ai}(x) + q_n(x)\ \mathrm{Ai}'(x) \qquad \text{and} \qquad \mathrm{Bi}_n(x) = p_n(x)\ \mathrm{Bi}(x) + q_n(x)\ \mathrm{Bi}'(x)\,,
$$

with polynomials p_n and q_n in x. This shows, however, that to deal with the Airy functions algorithmically as is suggested in this paper, besides the functions $\mathrm{Ai}(x)$ and $\mathrm{Bi}(x)$ the two independent functions $\mathrm{Ai}'(x)$ and $\mathrm{Bi}'(x)$ are needed, but none else. Let's look, how Computer Algebra systems work with the Airy functions.

MAPLE handles them as follows:

```
> Ai(x);
                                    Ai(x)
> diff(Ai(x),x);

       1/2                  3/2
      2    BesselK(1/3, 2/3 x   )
  1/4 ---------------------------
                 1/4
                x    Pi

                    /                                        3/2 \
           1/2  5/4 |                    3/2       BesselK(1/3, 2/3 x   )|
          2    x    |- BesselK(4/3, 2/3 x   ) + 1/2 ----------------------|
                    |                                      3/2        |
                    \                                     x           /
       + 1/3 -----------------------------------------------------------------
                                          Pi
```

```
> simplify(diff(Ai(x),x$2)-x*Ai(x));

            1/2                   3/2        1/2                    3/2   3/2
  1/48 (- 3 2    BesselK(1/3, 2/3 x   ) - 8 2    BesselK(-2/3, 2/3 x   ) x

            1/2  3                   3/2        9/4              /   5/4
       + 16 2    x  BesselK(1/3, 2/3 x   ) - 48 x    Ai(x) Pi)  /  (x    Pi)
                                                                /
> diff(Bi(x),x);
                                     d
                                    ---- Bi(x)
                                     dx
> diff(Bi(x),x$2);
                                      2
                                     d
                                   ----- Bi(x)
                                      2
                                    dx
```

So the derivative of Ai (x) is represented by Bessel functions, whereas the function Ai (x) itself is not, and therefore the expression `diff(Ai(x),x$2)-x*Ai(x)` is not simplified. On the other hand the derivative of Bi (x) is not a valid MAPLE function. With MATHEMATICA we get

```
In[7]:= D[AiryAi[x],x]

Out[7]= AiryAiPrime[x]

In[8]:= D[AiryAiPrime[x],x]

Out[8]= x AiryAi[x]

In[9]:= D[AiryAi[x],{x,2}]-x*AiryAi[x]

Out[9]= 0

In[10]:= D[AiryBi[x],x]

Out[10]= AiryBiPrime[x]

In[11]:= D[AiryBiPrime[x],x]

Out[11]= x AiryBi[x]

In[12]:= D[AiryBi[x],{x,2}]-x*AiryBi[x]

Out[12]= 0
```

Thus we see that in this situation MATHEMATICA does exactly what we suggest: It works with the independent functions Ai (x), Ai$'$ (x), Bi (x), Bi$'$ (x), and the derivative rules (20) and (21).

As a further example of an admissible family we consider the iterated integrals

$$\operatorname{erfc}_n(x) = \int_x^\infty \operatorname{erfc}_{n-1}(t)\, dt$$

of the (complementary) error function $\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) = \operatorname{erfc}_0(x)$ (see e. g. [2], (7.2)) that form the admissible family with

$$\operatorname{erfc}'_n(x) = -\operatorname{erfc}_{n-1}(x), \qquad \operatorname{erfc}_{n+1}(x) = \frac{1}{2(n+1)} \operatorname{erfc}_{n-1}(x) - \frac{x}{n+1} \operatorname{erfc}_n(x),$$

and the initial functions

$$\operatorname{erfc}_0(x) = \operatorname{erfc}(x), \qquad \operatorname{erfc}_1(x) = -\frac{1}{\sqrt{\pi}} \left(\sqrt{\pi}\, x \operatorname{erfc}(x) - e^{-x^2} \right)$$

(one may also use the initial value function $\operatorname{erfc}_{-1}(x) = \frac{2}{\sqrt{\pi}} e^{-x^2}$). In particular, $\operatorname{erfc} x$ is embedded into an admissible family.

Maple deals with these functions as suggested:

```
> diff(erfc(n,x),x);
                              - erfc(n - 1, x)

> simplify(diff(erfc(n,x),x$2)+2*x*diff(erfc(n,x),x)-2*n*erfc(n,x));

                                     0
```

As a final example, we mention another family of iterated integrals, the Abramowitz functions

$$A_n(x) := \int_0^\infty t^n\, e^{-t^2 - x/t}\, dt$$

(see [1], and [2], (27.5)) which form an admissible family with derivative rule

$$A'_n(x) = \frac{\partial}{\partial x} \left(\int_0^\infty t^n\, e^{-t^2 - x/t}\, dt \right) = \int_0^\infty \frac{\partial}{\partial x} \left(t^n\, e^{-t^2 - x/t} \right) dt = -\int_0^\infty t^{n-1}\, e^{-t^2 - x/t}\, dt = -A_{n-1}(x)$$

of order one (see [2], (27.5.2)), and recurrence formula

$$A_{n+1}(x) = \frac{n}{2} A_{n-1}(x) + \frac{x}{2} A_{n-2}(x)$$

of order three ([2], (27.5.3)).

Again, embedded into an admissible family, especially the function $A_0(x) = \int\limits_0^\infty e^{-t^2 - x/t}\, dt$ is covered by our approach.

6 Embedding the inhomogeneous case

Some families of functions are characterized by inhomogeneous differential rules and recurrence equations. Examples for this situation are the exponential integrals given by

$$E_n\,(x) = \int\limits_1^\infty \frac{e^{xt}}{t^n}\,dt$$

(see e. g. [2], (5.1)), and the Struve functions $\mathbf{H}_n(x)$ and $\mathbf{L}_n(x)$ (see e. g. [2], Chapter 5), for which we have the inhomogeneous properties

$$E_n'\,(x) = -E_{n-1}\,(x)\;, \qquad E_{n+1}\,(x) = \frac{e^{-x}}{n} - \frac{x}{n}\,E_n\,(x)\;,$$

([2], (5.1.14) and (5.1.26)),

$$\mathbf{H}_{n-1}(x) - \mathbf{H}_{n+1}(x) = 2\,\mathbf{H}_n'(x) - \frac{x^n}{2^n\,\sqrt{\pi}\,\Gamma(n+3/2)}\;, \tag{22}$$

$$\mathbf{H}_{n-1}(x) + \mathbf{H}_{n+1}(x) = \frac{2n}{x}\,\mathbf{H}_n(x) + \frac{x^n}{2^n\,\sqrt{\pi}\,\Gamma(n+3/2)}$$

([2], (12.1.9)–(12.1.10)), and

$$\mathbf{L}_{n-1}(x) + \mathbf{L}_{n+1}(x) = 2\,\mathbf{L}_n'(x) - \frac{x^n}{2^n\,\sqrt{\pi}\,\Gamma(n+3/2)}\;, \tag{23}$$

$$\mathbf{L}_{n-1}(x) - \mathbf{L}_{n+1}(x) = \frac{2n}{x}\,\mathbf{L}_n(x) + \frac{x^n}{2^n\,\sqrt{\pi}\,\Gamma(n+3/2)}$$

([2], (12.2.4)–(12.2.5)), respectively. Eliminating the inhomogeneous parts (using $\Gamma(3/2+n) = (1/2+n)\,\Gamma(1/2+n)$), these examples are made into admissible families with the derivative rules

$$E_n'\,(x) = -E_{n-1}\,(x)\;,$$

$$\mathbf{H}_n'(x) = \mathbf{H}_{n-1}(x) - \frac{n}{x}\,\mathbf{H}_n(x)\;, \tag{24}$$

$$\mathbf{L}_n'(x) = \mathbf{L}_{n-1}(x) - \frac{n}{x}\,\mathbf{L}_n(x)\;, \tag{25}$$

and the recurrence equations

$$\begin{aligned}
E_{n+1}\,(x) &= \frac{1}{n}\Big(x\,E_{n-1}(x) + (n-1-x)\,E_n(x)\Big)\;,\\
\mathbf{H}_{n+1}(x) &= \frac{1}{2n+1}\Big(x\,\mathbf{H}_{n-2}(x) + (1-4n)\,\mathbf{H}_{n-1}(x) + \frac{x^2+2n+4n^2}{x}\,\mathbf{H}_n(x)\Big)\;,\\
\mathbf{L}_{n+1}(x) &= \frac{1}{2n+1}\Big(-x\,\mathbf{L}_{n-2}(x) - (1-4n)\,\mathbf{L}_{n-1}(x) + \frac{x^2-2n-4n^2}{x}\,\mathbf{L}_n(x)\Big)\;,
\end{aligned}$$

so that the exponential integrals form an admissible family of order two, and the Struve functions $\mathbf{H}_n(x)$ and $\mathbf{L}_n(x)$ form admissible families of order three. Note that the above derivative rules (24)–(25) are not listed in [2] although they are much simpler than the inhomogeneous relations (22)–(23).

After bringing the inhomogeneous rules into the desired form, those families are recognized as admissible families, and our method can be applied.

7 Functions of the hypergeometric type as admissible families

All functions introduced in this paper are special cases of functions of the hypergeometric type (see [9]). In this section we will show that the generalized hypergeometric function ${}_pF_q$ defined by

$$ {}_pF_q \left(\begin{array}{cccc} a_1 & a_2 & \cdots & a_p \\ b_1 & b_2 & \cdots & b_q \end{array} \middle| \, x \right) := \sum_{k=0}^{\infty} A_k \, x^k = \sum_{k=0}^{\infty} \frac{(a_1)_k \cdot (a_2)_k \cdots (a_p)_k}{(b_1)_k \cdot (b_2)_k \cdots (b_q)_k \, k!} x^k \; , \tag{26} $$

and thus by Theorem 2 (c) all functions of the hypergeometric type, form admissible families. Therefore we first deduce a derivative rule of order two for ${}_pF_q$.

Let us choose any of the numerator parameters $n := a_k$ $(k = 1, \ldots, p)$ of ${}_pF_q$ as parameter n. Further we use the abbreviations

$$ F_n(x) = {}_pF_q \left(\begin{array}{cccc} n & a_2 & \cdots & a_p \\ b_1 & b_2 & \cdots & b_q \end{array} \middle| \, x \right) = \sum_{k=0}^{\infty} A_k(n) \, x^k \; . $$

From the relation

$$ \frac{(n+1)_k}{(n)_k} = \frac{n+k}{n} $$

it follows that

$$ n \, A_k(n+1) = (n+k) \, A_k(n) \; . $$

Using the differential operator $\theta f(x) = x \, f'(x)$, we get by summation

$$ \begin{aligned} n \, F_{n+1}(x) &= n \sum_{k=0}^{\infty} A_k(n+1) \, x^k = (n+k) \sum_{k=0}^{\infty} A_k(n) \, x^k \\ &= n \, F_n(x) + \sum_{k=0}^{\infty} k \, A_k(n) \, x^k = n \, F_n(x) + \theta F_n(x) \; , \end{aligned} $$

and therefore we are led to the derivative rule

$$ \theta F_n(x) = n \Big(F_{n+1}(x) - F_n(x) \Big) \; , \qquad \text{or} \qquad F_n'(x) = \frac{n}{x} \Big(F_{n+1}(x) - F_n(x) \Big) \; . \tag{27} $$

Hence we have established that for any of the numerator parameters $n := a_k$ $(k = 1, \ldots, p)$ of ${}_pF_q$ such a simple (forward) derivative rule is valid.

We note that by similar means for each of the denominator parameters $n := b_k$ $(k = 1, \ldots, q)$ of ${}_pF_q$ the simple (backward) derivative rule

$$ \theta F_n(x) = (n-1) \Big(F_{n-1}(x) - F_n(x) \Big) \; , \qquad \text{or} \qquad F_n'(x) = \frac{n-1}{x} \Big(F_{n-1}(x) - F_n(x) \Big) \tag{28} $$

is derived.

Next, we note that F_n satisfies the well-known hypergeometric differential equation

$$\theta(\theta + b_1 - 1) \cdots (\theta + b_q - 1) F_n(x) = x(\theta + a_1)(\theta + a_2) \cdots (\theta + a_p) F_n(x) \ . \tag{29}$$

Replacing all occurrences of θ in (29) recursively by the derivative rule (27) or (28), a recurrence equation for F_n is obtained that turns out to have the same order as the differential equation (29), i. e. $\max\{p, q+1\}$.

We summarize the above results in the following

Theorem 3 The generalized hypergeometric function ${}_pF_q \left(\begin{array}{cccc} a_1 & a_2 & \cdots & a_p \\ b_1 & b_2 & \cdots & b_q \end{array} \middle| x \right)$ satisfies the derivative rules

$$\theta F_n(x) = n \left(F_{n+1}(x) - F_n(x) \right)$$

for any of its numerator parameters $n := a_k$ $(k = 1, \ldots, p)$, and

$$\theta F_n(x) = (n-1) \left(F_{n-1}(x) - F_n(x) \right)$$

for any of its denominator parameters $n := b_k$ $(k = 1, \ldots, q)$, and recursive substitution of all occurrences of θ in the hypergeometric differential equation

$$\theta(\theta + b_1 - 1) \cdots (\theta + b_q - 1) F_n(x) = x(\theta + a_1)(\theta + a_2) \cdots (\theta + a_p) F_n(x)$$

generates a recurrence equation of the type (3) of order $\max\{p, q+1\}$ with respect to the parameter chosen. This recurrence equation has coefficients that are rational with respect to x, and n. In particular, ${}_pF_q$ forms an admissible family of order $\max\{p, q+1\}$ with respect to all of its parameters a_k, b_k. $\square$

We note that if some of the parameters of ${}_pF_q$ are specified, there may exist a lower order differential equation, and thus the order of the admissible family may be lower than the theorem states. We note further that this theorem is the main reason for the fact that so many special functions form admissible families: Most of them can be represented in terms of generalized hypergeometric functions.

8 Algorithmic generation of differential equations

In this section we show that the algorithm to generate the uniquely determined differential equation of type (2) of lowest order valid for f which was developed in [9] (see also [15]), does apply if f is constructed from functions that are embedded into admissible families.

Algorithm 3 (Find a simple differential equation) Let f be a function given by an expression that is built from the functions $\exp x$, $\ln x$, $\sin x$, $\cos x$, $\arcsin x$, $\arctan x$, and any other functions that are embedded into admissible families, with the aid of the following procedures: differentiation, antidifferentiation, addition, multiplication, and the composition with rational functions and rational powers.

Then the following procedure generates a simple differential equation valid for f:

(a) Find out whether there exists a simple differential equation for f of order $N := 1$. Therefore differentiate f, and solve the linear equation

$$f'(x) + A_0 f(x) = 0$$

for A_0; i. e. set $A_0 := -\frac{f'(x)}{f(x)}$. Is A_0 rational in x, then you are done after multiplication with its denominator.

(b) Increase the order N of the differential equation searched for by one. Expand the expression

$$f^{(N)}(x) + A_{N-1} f^{(N-1)}(x) + \cdots + A_0 f(x) ,$$

apply the recurrence formulas of any admissible family F_n of order m involved recursively to minimize the occurrences of F_{n-k} to at most m successive k-values, and check, if the remaining summands contain exactly N rationally independent expressions considering the numbers $A_0, A_1, \ldots, A_{N-1}$ as constants. Just in that case there exists a solution as follows: Sort with respect to the rationally independent terms and create a system of linear equations by setting their coefficients to zero. Solve this system for the numbers $A_0, A_1, \ldots, A_{N-1}$. Those are rational functions in x, and if there is a solution, this solution is unique. After multiplication by the common denominator of $A_0, A_1, \ldots, A_{N-1}$ you get the differential equation searched for. Finally cancel common factors of the polynomial coefficients.

(c) If part (b) was not successful, repeat step (b).

Proof: Theorem 3 of [15] (compare [22]) shows that for f a differential equation of type (2) exists. We assume that differentiation is done by recursive descent through the expression tree, and an application of the chain, product and quotient rules on the corresponding subexpressions. It is clear that the algorithm works for members of admissible families, compare Theorem 1 and Corollary 1. Similarly the algorithm obviously works for derivatives and antiderivatives of admissible families. Further it is easily seen that the derivatives of sums, products, and the composition with rational functions and rational powers form either sums, or sums of products all of which by a recursive use of the recurrence equations involved are represented by sums of fixed lengths, compare Theorem 2. Thus after a finite number of steps, part (b) of the algorithm will succeed (sharp a priory bounds for the resulting orders are given in [15]). $\square$

We note that from the implementational point of view the crucial step of the algorithm is the decision of the rational independency in part (b). If this decision can be handled properly, then the proof given in [9] shows that the algorithm generates the simple differential equation of lowest order valid for f.

In our implementations, for testing whether some terms are rationally dependent, we divide each one by any other and test whether the quotient is a rational function in x or not. This is an easy and fast approach which never leads to wrong results, but may miss a simpler solution, which in practice, rarely happens.

Typically this happens, however, for orthogonal polynomials with prescribed n, for which a first order differential equation exists. In this case, the recurrence equation hides these rational dependencies, and in some sense (s. [6], § 7) here it is even advantageous that the rational dependency is not realized.

Another example where our implementations yield a differential equation which is not of lowest order is given by

```
In[13]:= SimpleDE[Sin[2 x]-2 Sin[x] Cos[x],x]

Out[13]= 4 F[x] + F''[x] == 0
```

This happens because the functions $\sin(2x)$ and $2\sin x\cos x$ algebraically cannot be verified to be rationally dependent even though they are identical.

We note that, for elementary functions, we could use the Risch normalization procedure [20] to generate the rationally independent terms, but this does not work for special functions.

Further we note that in case of expressions of high complexity, the use of [15], Algorithm 2, typically is faster. This algorithm, however, in general leads to a differential equation of higher order than Algorithm 3.

As a first application of Algorithm 3 we consider the Airy functions Ai_n, again, for which the MATHEMATICA implementation of our algorithm yields

```
In[14]:= SimpleDE[AiryAi[n,x],x]

                                    (3)
Out[14]= (-1 - n) F[x] - x F'[x] + F   [x] == 0
```

i. e. the differential equation

$$\mathrm{Ai}_n'''(x) - x\ \mathrm{Ai}_n'(x) - (n+1)\ \mathrm{Ai}_n(x) = 0\ . \tag{30}$$

Similarly, we get for the square of the Airy function

```
In[15]:= SimpleDE[AiryAi[x]^2,x]

                                (3)
Out[15]= -2 F[x] - 4 x F'[x] + F   [x] == 0
```

The next calculation confirms the differential equation for the Bateman functions F_n (14)

```
In[16]:= SimpleDE[Bateman[n,x],x]

Out[16]= (2 n - x) F[x] + x F''[x] == 0
```

Other examples are given with the aid of the iterated integrals of the complementary error function, and the Abramowitz functions:

```
In[17]:= SimpleDE[Erfc[n,x],x]

Out[17]= -2 n F[x] + 2 x F'[x] + F''[x] == 0
```

(see [2] (7.2.2)) and

```
In[18]:= SimpleDE[Exp[a x]*Erfc[n,x],x]

            2
Out[18]= (a  - 2 n - 2 a x) F[x] + (-2 a + 2 x) F'[x] + F''[x] == 0
```

```
In[19]:= SimpleDE[Exp[a x^2]*Erfc[n,x],x]

                                2      2  2
Out[19]= (-2 a - 2 n - 4 a x  + 4 a  x ) F[x] + (2 x - 4 a x) F'[x] +

>     F''[x] == 0

In[20]:= SimpleDE[Abramowitz[n,x],x]

                                      (3)
Out[20]= 2 F[x] + (1 - n) F''[x] + x F   [x] == 0
```

(see [2] (26.2.41)).

We note that the algorithm obviously works for antiderivatives. An example of that type is Dawson's integral (see e. g. [2] (7.1.17)) for which we get the differential equation

```
In[21]:= SimpleDE[E^(-x^2)*Integrate[E^(t^2),{t,0,x}],x]

Out[21]= 2 F[x] + 2 x F'[x] + F''[x] == 0
```

For the Struve functions, our algorithm generates the differential equations

$$(n^2+n^3+x^2-nx^2)\,\mathbf{H}_n(x)+x\,(x^2-n-n^2)\,\mathbf{H}_n'(x)+(2-n)\,x^2\,\mathbf{H}_n''(x)+x^3\,\mathbf{H}_n'''(x)=0\ ,$$

and

$$(n^2+n^3-x^2+nx^2)\,\mathbf{L}_n(x)-x\,(x^2+n+n^2)\,\mathbf{L}_n'(x)+(2-n)\,x^2\,\mathbf{L}_n''(x)+x^3\,\mathbf{L}_n'''(x)=0\ ,$$

that are the homogeneous counterparts of the differential equation (12.1.1) in [2].

Finally we give examples involving hypergeometric functions:

```
In[22]:= SimpleDE[Hypergeometric2F1[a,b,c,x],x]

Out[22]= a b F[x] + (-c + x + a x + b x) F'[x] + (-1 + x) x F''[x] == 0

In[23]:= SimpleDE[Hypergeometric2F1[a,b,a+b+1/2,x]^2,x]

Out[23]= 8 a b (a + b) F[x] + 2 (-a - 2 a  - b - 4 a b - 2 b  + x + 3 a x +

              2                           2
>        2 a  x + 3 b x + 8 a b x + 2 b  x) F'[x] +

>     3 x (-1 - 2 a - 2 b + 2 x + 2 a x + 2 b x) F''[x] +

                   2  (3)
>     2 (-1 + x) x  F   [x] == 0
```

Here the last function considered $\left({}_2F_1\left(\left.\begin{array}{c} a\quad b \\ a+b+1/2 \end{array}\right| x\right)\right)^2$ is the left hand side of Clausen's formula (31) that we will consider again in § 9.

Now we investigate the case that a derivative rule and a differential equation are given, and show that these two imply the existence of a recurrence equation:

Algorithm 4 If a family f_n is given by a derivative rule of type (1) and a differential equation of type (2), then it forms an admissible family for which a recurrence equation can be found algorithmically.

Proof: We present an algorithm which generates a recurrence equation for f_n: Iterative differentiation of the derivative rule (1) with the explicit use of (1) at each step yields

$$f_n^{(j)}(x) = \sum_{k=0}^{M} r_k^j(n,x)\, f_{n-k}(x)$$

with rational functions r_k^j. The substitution of these derivative representations in the differential equation gives the recurrence equation searched for. □

As an example we consider the Airy functions Ai_n, again, for which we have the derivative rule (20)

$$\mathrm{Ai}_n'(x) = \mathrm{Ai}_{n+1}(x)$$

and the differential equation (30)

$$\mathrm{Ai}_n'''(x) - x\ \mathrm{Ai}_n'(x) - (n+1)\ \mathrm{Ai}_n(x) = 0\ .$$

Differentiating the derivative rule successively and substituting the resulting expressions into the differential equation immediately yields the recurrence equation (20), again.

If this family, however, is given by the backward derivative rule (compare (20))

$$\mathrm{Ai}_n'(x) = x\ \mathrm{Ai}_{n-1}(x) + (n-1)\ \mathrm{Ai}_{n-2}(x)\ ,$$

then differentiation yields

$$\begin{aligned}
\mathrm{Ai}_n''(x) &= \mathrm{Ai}_{n-1}(x) + x\ \mathrm{Ai}_{n-1}'(x) + (n-1)\ \mathrm{Ai}_{n-2}'(x) \\
&= \mathrm{Ai}_{n-1}(x) + x\Big(x\,\mathrm{Ai}_{n-2}(x) + (n-2)\,\mathrm{Ai}_{n-3}(x)\Big) + (n-1)\Big(x\,\mathrm{Ai}_{n-3}(x) + (n-3)\,\mathrm{Ai}_{n-4}(x)\Big) \\
&= \mathrm{Ai}_{n-1}(x) + x^2\ \mathrm{Ai}_{n-2}(x) + (2n-3)\,x\ \mathrm{Ai}_{n-3}(x) + (n^2-4n+3)\ \mathrm{Ai}_{n-4}(x)\ .
\end{aligned}$$

After a similar procedure we get

$$\begin{aligned}
\mathrm{Ai}_n'''(x) &= 3x\ \mathrm{Ai}_{n-2}(x) + (x^2+3n-5)\ \mathrm{Ai}_{n-3}(x) + (3n-6)\,x^2\ \mathrm{Ai}_{n-4}(x) \\
&\quad + (3n^2-15n+15)\,x\ \mathrm{Ai}_{n-5}(x) + (n^3-9n^2+23n-15)\ \mathrm{Ai}_{n-6}(x)\ ,
\end{aligned}$$

and the substitution into the differentiation equation gives finally

$$\begin{aligned}
&(n-1)\ \mathrm{Ai}_n(x) - x^2\ \mathrm{Ai}_{n-1}(x) + (4-n)\,x\ \mathrm{Ai}_{n-2}(x) + (3n-5+x^3)\ \mathrm{Ai}_{n-3}(x) \\
&+ (3n-6)\,x^2\ \mathrm{Ai}_{n-4}(x) + (3n^2-15n+15)\,x\ \mathrm{Ai}_{n-5}(x) + (n^3-9n^2+23n-15)\ \mathrm{Ai}_{n-6}(x) = 0\ ,
\end{aligned}$$

a recurrence equation of order 6 rather than the minimal order three. This shows, that, in general, the order of the resulting recurrence equation is not best possible.

Algebraically spoken, our result tells that if $\{f_n^{(j)} \mid j \in \mathbb{N}_0\}$ has finite dimension, and if f_n' is an element of the linear space V spanned by a finite number of the functions $\{f_{n\pm k}\}$, then the space generated by all of $\{f_{n\pm k}\}$ is of finite dimension, too. In contrast to Theorem 1, however, the dimension of this space generally may be higher than the dimension of V. This shows the advantage of the use of admissible families.

As a further result of this section we note that using our general procedure developed in [9] we have

Algorithm 5 (Find a Laurent-Puiseux representation) Let f be a function that is built from the functions $\exp x$, $\ln x$, $\sin x$, $\cos x$, $\arcsin x$, $\arctan x$, and any other functions that are embedded into admissible families, with the aid of the following procedures: differentiation, anti-differentiation, addition, multiplication, and the composition with rational functions and rational powers.

If furthermore f turns out to be of rational, exp-like, or hypergeometric type (see [9]), then a closed form Laurent-Puiseux representation $f(x) = \sum\limits_{k=k_0}^{\infty} a_k\, x^{k/n}$ can be obtained algorithmically. □

We remark that there is a decision procedure due to Petkovsek [19] to decide the hypergeometric type from the recurrence equation obtained.

With Algorithm 5, it is possible to reproduce most of the results of the extensive bibliography on series [7], and to generate others. As an example we present the power series representation of the square of the Airy function:

```
In[24]:= PowerSeries[AiryAi[x]^2,x]

               1 k    k  1 + 3 k
              (-)  27  x        (2 k)!
               9
Out[24]= Sum[-(------------------------), {k, 0, Infinity}] +
              Sqrt[3] Pi k! (1 + 3 k)!

          k  3 k            1
        12  x    Pochhammer[-, k]
                            6
>    Sum[-------------------------, {k, 0, Infinity}] +
           1/3              2 2
         3 3    (3 k)! Gamma[-]
                             3

          1/3   k          2 + 3 k           5
        2 3    12  (1 + k) x        Pochhammer[-, k]
                                             6
>    Sum[-------------------------------------------, {k, 0, Infinity}]
                                        1 2
                     (3 + 3 k)! Gamma[-]
                                        3
```

Note that, moreover, this technique generates hypergeometric representations, whenever such representations exist. The above example, e. g., is recognized as the hypergeometric representation

$$
\begin{aligned}
\mathrm{Ai}\,(x)^2 \;&=\; \frac{1}{3^{4/3}\,\Gamma\,(2/3)^2}\,{}_1F_2\left(\begin{array}{c} 1/6 \\ 1/3 \quad 2/3 \end{array}\,\middle|\, \frac{4}{9}\,x^3\right) \\
&\quad -\frac{x}{\sqrt{3}\,\pi}\,{}_1F_2\left(\begin{array}{c} 1/2 \\ 2/3 \quad 4/3 \end{array}\,\middle|\, \frac{4}{9}\,x^3\right) + \frac{x^2}{3^{2/3}\,\Gamma\,(1/3)^2}\,{}_1F_2\left(\begin{array}{c} 5/6 \\ 4/3 \quad 5/3 \end{array}\,\middle|\, \frac{4}{9}\,x^3\right) .
\end{aligned}
$$

As soon as a hypergeometric representation is obtained, by Theorem 3 derivatives rules and recurrence equations with respect to all parameters involved may be obtained. As an example, we consider the Laguerre polynomials: The power series representation for the Laguerre polynomial $L_n^{(\alpha)}(x)$ that our algorithm generates corresponds to the hypergeometric representation

$$L_n^{(\alpha)}(x) = \binom{n+\alpha}{n}\, {}_1F_1\left(\begin{array}{c} -n \\ \alpha+1 \end{array}\,\middle|\, x\right)$$

from which by an application of Theorem 3 we obtain the derivative rule

$$\begin{aligned} \frac{\partial}{\partial x} L_n^{(\alpha)}(x) &= \binom{n+\alpha}{n} \frac{-n}{x} \left({}_1F_1\left(\begin{array}{c} -n+1 \\ \alpha+1 \end{array}\,\middle|\, x\right) - {}_1F_1\left(\begin{array}{c} -n \\ \alpha+1 \end{array}\,\middle|\, x\right)\right) \\ &= \frac{-(n+\alpha)}{x} \binom{n-1+\alpha}{n-1}\, {}_1F_1\left(\begin{array}{c} -(n-1) \\ \alpha+1 \end{array}\,\middle|\, x\right) + \frac{n}{x}\binom{n+\alpha}{n}\, {}_1F_1\left(\begin{array}{c} -n \\ \alpha+1 \end{array}\,\middle|\, x\right) \\ &= \frac{1}{x}\left(-(n+\alpha)\, L_{n-1}^{(\alpha)}(x) + n\, L_n^{(\alpha)}(x)\right) , \end{aligned}$$

i. e. (18), again, but we are also led to the derivative rule with respect α:

$$\begin{aligned} \frac{\partial}{\partial x} L_n^{(\alpha)}(x) &= \binom{n+\alpha}{n} \frac{\alpha}{x} \left({}_1F_1\left(\begin{array}{c} -n \\ \alpha \end{array}\,\middle|\, x\right) - {}_1F_1\left(\begin{array}{c} -n \\ \alpha+1 \end{array}\,\middle|\, x\right)\right) \\ &= \frac{\alpha}{x}\,\frac{n+\alpha}{\alpha} \binom{n+\alpha-1}{n}\, {}_1F_1\left(\begin{array}{c} -n \\ \alpha \end{array}\,\middle|\, x\right) - \frac{\alpha}{x}\binom{n+\alpha}{n}\, {}_1F_1\left(\begin{array}{c} -n \\ \alpha+1 \end{array}\,\middle|\, x\right) \\ &= \frac{1}{x}\left((n+\alpha)\, L_n^{(\alpha-1)}(x) - \alpha\, L_n^{(\alpha)}(x)\right) . \end{aligned}$$

A further application of Theorem 3 yields the recurrence equation

$$F_{\alpha+1} = \frac{1+\alpha}{(1+\alpha+n)\,x}\Big(-\alpha\, F_{\alpha-1} + (\alpha+x)\, F_\alpha\Big)$$

for $F_\alpha := {}_1F_1\left(\begin{array}{c} -n \\ \alpha+1 \end{array}\,\middle|\, x\right)$ with respect to α, and the use of the algorithm for the product ([15], Theorem 3 (d), [27], p. 342, and [21], MAPLE function `rec*rec`), applied to $L_n^{(\alpha)} = \binom{n+\alpha}{n} \cdot F_\alpha$ generates (19), again.

9 Algorithmic verification of identities

On the lines of [27] we can now present an implementable algorithm to verify identities between expressions using the results of the last section.

Algorithm 6 (Verification of identities) Assume two functions $f_n(x)$ and $g_n(x)$ are given, to which Algorithm 3 applies. Then the following procedure verifies whether f_n and g_n are identical:

(a) `de1:=SimpleDE(f,x)`:
Determine the simple differential equation `de1` corresponding to f_n.

(b) `de2:=SimpleDE(g,x)`:
Determine the simple differential equation `de2` corresponding to g_n.

(c) **(Different differential equation implies different function)** If `de1` and `de2` have the same order, then

- if they do not coincide besides common factors, i. e. have rational ratio, then f_n and g_n do not coincide; return this, and quit.
- Otherwise f_n and g_n satisfy the same differential equation `de1` of order l, say, and it remains to check l initial values. Continue with (e).

(d) Let the orders of `de1` and `de2`, i. e.

$$\sum_{j=0}^{l} p_j\, f_n^{(j)} = 0 \qquad \text{and} \qquad \sum_{k=0}^{m} q_k\, g_n^{(k)} = 0$$

(p_j ($j = 0, \ldots, l$), q_k ($k = 0, \ldots, m$) polynomials) are different, and assume without loss of generality that $l > m$. Then, differentiate `de2` $l - m$ times to get equations

$$S_p := \sum_{k=0}^{p} q_k^p\, g_n^{(k)} = 0 \qquad\qquad (p = m, \ldots, l)\ .$$

Check if there are nontrivial rational functions $A_p \not\equiv 0$ ($p = m, \ldots, l$) such that a linear combination $\sum\limits_{p=m}^{l} A_p\, S_p$ is equivalent to the left hand side of `de1`, i. e. is a rational multiple of it.

If this is not the case, then f_n and g_n do not satisfy a common simple differential equation, and therefore are not identical; return this, and quit. Otherwise they satisfy a common simple differential equation; continue with (e).

(e) Let l be the order of the common simple differential equation for f_n and g_n. For $k = 0, \ldots, l-1$ check if $f_n^{(k)}(0) = g_n^{(k)}(0)$. (Note that by the holonomic structure the knowledge of the initial values (9) is sufficient to generate those.) These initial conditions may depend on n, and are proved by application of a discrete version of the same algorithm. If one of these equations is falsified, then the identity $f_n \equiv g_n$ is disproved; return this, and quit. Otherwise, if all equations are verified, the identity $f_n \equiv g_n$ is proved.

Proof: By a well-known result about differential equations of the type considered, the solution of an initial value problem

$$\sum_{k=0}^{l} p_k(x)\, f_n^{(k)}(x) = 0\ , \qquad f_n^{(k)}(0) = a_k\ (k = 0, \ldots, l-1)$$

is unique. To prove that f_n and g_n are identical, it therefore suffices to show that they satisfy a common differential equation, and the same initial values. This is done by our algorithm. $\square$

For the example expressions

$$f_n(x) := L_n^{(-1/2)}(x)$$

and

$$g_n(x) := \frac{(-1)^n}{n!\, 2^{2n}}\, H_{2n}\left(\sqrt{x}\right)$$

we get the common differential equation

$$2\,n\,f(x) + (1-2x)\,f'(x) + 2\,x\,f''(x) = 0\;.$$

Therefore to prove the identity

$$L_n^{(-1/2)}(x) = \frac{(-1)^n}{n!\,2^{2n}}\,H_{2n}\left(\sqrt{x}\right)\;,$$

(see e. g. [2], (22.5.38)), it is enough to verify the two initial equations $f_n(0) = g_n(0)$ and $f_n'(0) = g_n'(0)$. To establish the first of these conditions, with MATHEMATICA, e. g., we get

```
In[25]:= eq = Limit[LaguerreL[n,-1/2,x],x->0]==
         Limit[(-1)^n/(n!*2^(2*n))*HermiteH[2*n,Sqrt[x]],x->0]

                                1
         Pochhammer[1 + n, -(-)]        n
                                2      (-1)  Sqrt[Pi]
Out[25]= ----------------------- == ---------------
                 Sqrt[Pi]                     1
                                    n! Gamma[- - n]
                                              2
```

which is to be verified. In this situation, we establish the first order recurrence equations for both sides

```
In[26]:= FindRecursion[Limit[LaguerreL[n,-1/2,x],x->0],n]

Out[26]= (-1 + 2 n) a[-1 + n] - 2 n a[n] == 0

In[27]:= FindRecursion[Limit[(-1)^n/(n!*4^n)*HermiteH[2*n,Sqrt[x]],x->0],n]

Out[27]= (-1 + 2 n) a[-1 + n] - 2 n a[n] == 0
```

that coincide, so that it remains to prove the initial statement

```
In[28]:= eq /. n->0

Out[28]= True
```

and we are done. Similarly one may prove the second initial value statement $f_n'(0) = g_n'(0)$.

Applying the same method, (12) can be proved by the calculations

```
In[29]:= SimpleDE[(n+1)*Bateman[n+1,x]-(n-1)*Bateman[n-1,x],x]

                               2           2    3                             2
Out[29]= (2 n - 2 x + 4 n  x - 4 n x  + x ) F[x] + (-2 n x + 2 x ) F'[x] +

                      2
>      (2 n - x) x  F''[x] == 0
```

```
In[30]:= SimpleDE[2*x*D[Bateman[n,x],x],x]

                         2          2    3                      2
Out[30]= (2 n - 2 x + 4 n  x - 4 n x  + x ) F[x] + (-2 n x + 2 x ) F'[x] +

                  2
>     (2 n - x) x  F''[x] == 0
```

and using the initial values $F_n(0) = 0$ and $F_n'(0) = -2$ (see [14], (11)).

Also, one can prove Clausen's formula

$$\left({}_2F_1 \left(\begin{array}{c} a \quad b \\ a+b+1/2 \end{array} \middle| x \right) \right)^2 = {}_3F_2 \left(\begin{array}{c} 2a \quad 2b \quad a+b \\ a+b+1/2 \quad 2a+2b \end{array} \middle| x \right) , \tag{31}$$

generating the common differential equation

$$\begin{aligned} &8\,a\,b\,(a+b)\;f(x) + \\ &2\,(-a - 2\,a^2 - b - 4\,a\,b - 2\,b^2 + x + 3\,a\,x + 2\,a^2\,x + 3\,b\,x + 8\,a\,b\,x + 2\,b^2\,x)\;f'(x) + \\ &3\,x\;(-1 - 2\,a - 2\,b + 2\,x + 2\,a\,x + 2\,b\,x)\;f''(x) + \\ &2\;(-1+x)\;x^2\,f'''(x) = 0 \end{aligned}$$

for both sides of (31), or other hypergeometric identities like the Kummer transformation

$${}_1F_1 \left(\begin{array}{c} a \\ b \end{array} \middle| x \right) = e^x\;{}_1F_1 \left(\begin{array}{c} b-a \\ b \end{array} \middle| -x \right)$$

or like

$${}_0F_1 \left(a \,\middle|\, x \right) \cdot {}_0F_1 \left(b \,\middle|\, x \right) = {}_2F_3 \left(\begin{array}{c} \frac{a+b}{2} \quad \frac{a+b-1}{2} \\ a \quad b \quad a+b-1 \end{array} \middle| 4\,x \right)$$

and

$${}_1F_1 \left(\begin{array}{c} a \\ b \end{array} \middle| x \right) \cdot {}_1F_1 \left(\begin{array}{c} a \\ b \end{array} \middle| -x \right) = {}_2F_3 \left(\begin{array}{c} a \quad b-a \\ b \quad \frac{b}{2} \quad \frac{b+1}{2} \end{array} \middle| \frac{x^2}{4} \right)$$

corresponding to the Kummer differential equation

$$a\,f(x) - (b-x)\;f'(x) - x\,f''(x) = 0\;,$$

and to

$$\begin{aligned} &(1-a-b)\;(a+b)\;f(x) + \left(-a\,b + a^2\,b + a\,b^2 - 2\,x - 4\,a\,x - 4\,b\,x\right)\;f'(x) + \\ &+ \left(a + a^2 + b + 3\,a\,b + b^2 - 4\,x\right)\;x\,f''(x) + 2\;(1+a+b)\;x^2\,f'''(x) + x^3\,f''''(x) = 0\;, \end{aligned}$$

and

$$\begin{aligned} &4\,a\;(a-b)\;x\,f(x) + \left(b - 3\,b^2 + 2\,b^3 - x^2 - 2\,b\,x^2\right)\;f'(x) \\ &+x\;\left(-b + 5\,b^2 - x^2\right)\;f''(x) + (1+4\,b)\;x^2\,f'''(x) + x^3\,f''''(x) = 0\;, \end{aligned}$$

respectively.

Note that one can also reverse the order of the algorithm, i. e. first find common recurrence equations for f_n and g_n with respect to n, and then check the initial conditions (depending on x) with the aid of differential equations. This method should be compared with recent results of Zeilberger ([27]–[29]).

Moreover the given algorithm is easily extended to the case of several variables, if the family given forms an admissible family with respect to all of its variables, i. e. for each variable exists

- either a simple recurrence equation (corresponding to a "discrete" variable),
- or a simple derivative rule (corresponding to a "continuous" variable), depending on shifts with respect to one of the discrete variables.

Note, however, that (for the moment) the algorithm only works if f and g are "expressions", and no symbolic sums, derivatives of symbolic order, etc. occur. In the next sections, we will, however, extend the above algorithm to these situations.

10 Algorithmic verification of Rodrigues type formulas

Here we present an algorithm to verify identities of the Rodrigues type

$$g(n,x) = f^{(n)}(n,x) \qquad (f, g \text{ functions}\ , \quad n \text{ symbolic})\ .$$

This algorithm, however, does only work if the function f is of the hypergeometric type. On the other hand, for most Rodrigues type formulas in the literature, see e. g. [2], this condition is valid.

The procedure is based on the following

Algorithm 7 (Find differential equation for derivatives of symbolic order) Let f be of the hypergeometric type, i. e. there is a Laurent-Puiseux type representation $f(n,x) = \sum_k a_k x^k$. Then there is a simple differential equation for $g(n,x) := f^{(n)}(n,x)$ which can be obtained by the following algorithm:

(a) `de1:=SimpleDE(f,x)`:
Calculate the simple differential equation `de1` of f, see Algorithm 3.

(b) `re1:=DEtoRE(de1,f,x,a,k)`:
Transfer the differential equation `de1` into the corresponding recurrence equation `re1` for a_k, see [9], §6.

(c) If `re1` is not of the hypergeometric type (or is not equivalent to the hypergeometric type [19]), then quit.

(d) `re2:=SymbolicDerivativeRE(re1,a,k,n)`:
Otherwise set $c_k := (k+1)_n\, a_{k+n}$. Bring `re1` into the form

$$a_{k+m} = R(k)\, a_k\ ,$$

rational R, and calculate the hypergeometric type recurrence equation `re2`

$$c_{k+m} = \frac{(k+n+1)_m}{(k+1)_m}\, R(k+n)\, c_k \tag{32}$$

for c_k.

(e) `de2:=REtoDE(re2,a,k,G,x)`:
Transfer the recurrence equation `re2` into the corresponding differential equation `de2` for the nth derivative $g(n,x) := f^{(n)}(x)$ of f, see [9], § 11.

Proof: Parts (a), (b) and (e) of the algorithm are described precisely in [9]. Now, assume, $g(n,x) = f^{(n)}(n,x)$, and that f has the representation $f(n,x) = \sum_k a_k x^k$. Then we get

$$\sum_k c_k\, x^k = g(n,x) = f^{(n)}(n,x) = \sum_k (k+1-n)_n\, a_k\, x^{k-n} = \sum_k (k+1)_n\, a_{k+n}\, x^k \;.$$

Therefore we have $c_k = (k+1)_n\, a_{k+n}$, and we get the recurrence equation

$$\begin{aligned} c_{k+m} &= (k+m+1)_n\, a_{k+n+m} = (k+m+1)_n\, R(k+n)\, a_{k+n} \\ &= \frac{(k+m+1)_n}{(k+1)_n}\, R(k+n)\, c_k = \frac{(k+m+n)!}{(k+m)!}\frac{k!}{(k+n)!}\, R(k+n)\, c_k \\ &= \frac{(k+m+n)!}{(k+n)}\frac{k!}{(k+m)!}\, R(k+n)\, c_k = \frac{(k+n+1)_m}{(k+1)_m}\, R(k+n)\, c_k \;, \end{aligned}$$

and hence (32), for c_k. This finishes the proof. □

As a first example we consider the identity

$$\mathrm{erfc}_n(x) = \frac{(-1)^n\, e^{-x^2}}{2^n\, n!}\frac{\partial^n}{\partial x^n}\left(e^{x^2}\,\mathrm{erfc}\,x\right)$$

(see e. g. [2], (7.2.9)), or equivalently

$$(-1)^n\, 2^n\, n!\, e^{x^2}\;\mathrm{erfc}_n(x) = \frac{\partial^n}{\partial x^n}\left(e^{x^2}\,\mathrm{erfc}\,x\right) \;. \tag{33}$$

Algorithm 7 yields step by step

```
In[31]:= de1=SimpleDE[E^(x^2)*Erfc[x],x]

Out[31]= -2 F[x] - 2 x F'[x] + F''[x] == 0

In[32]:= re1=DEtoRE[de1,F,x,a,k]

Out[32]= -2 (1 + k) a[k] + (1 + k) (2 + k) a[2 + k] == 0

In[33]:= re2=SymbolicDerivativeRE[re1,a,k,n]
                                              2
Out[33]= -2 (1 + k + n) a[k] + (2 + 3 k + k ) a[2 + k] == 0

In[34]:= de2=REtoDE[re2,a,k,G,x]

Out[34]= -2 (1 + n) G[x] - 2 x G'[x] + G''[x] == 0
```

thus finally the differential equation

$$-2\,(1+n)\,g(x) - 2\,x\,g'(x) + g''(x) = 0$$

for the function $\frac{\partial^n}{\partial x^n}\left(e^{x^2}\,\mathrm{erfc}\,x\right)$, which also can be obtained by the single statement

```
In[35]:= RodriguesDE[E^(x^2)*Erfc[x],x,n]

Out[35]= -2 (1 + n) F[x] - 2 x F'[x] + F''[x] == 0
```

For the left hand term of (33) we get

```
In[36]:= de3=SimpleDE[E^(x^2)*Erfc[n,x],x]

Out[36]= -2 (1 + n) F[x] - 2 x F'[x] + F''[x] == 0
```

i. e. the same differential equation.

As next example we consider the Rodrigues type identity (15) for the Bateman functions, and rewrite it as

$$\frac{n!\,e^{-x}}{x}\,F_n(x) = \frac{d^n}{dx^n}\left(e^{-2x}\,x^{n-1}\right)\ . \tag{34}$$

Our implementation yields

```
In[37]:= RodriguesDE[E^(-2x)*x^(n-1),x,n]

Out[37]= 2 (1 + n) F[x] + 2 (1 + x) F'[x] + x F''[x] == 0

In[38]:= SimpleDE[E^(-x)/x*Bateman[n,x],x]

Out[38]= 2 (1 + n) F[x] + 2 (1 + x) F'[x] + x F''[x] == 0
```

Algorithm 7 shows the applicability of Algorithm 6 if in the expressions involved Rodrigues type expressions occur, as soon as we can handle the initial values. Since in Algorithm 7 the function f is assumed to be of hypergeometric type, this, however, can be done by a series representation using Algorithm 5 if f moreover is analytic, and if the function f of Algorithm 7 does not depend on n: In this case Algorithm 5 generates the generic coefficient a_k of the series representation $f(x) = \sum\limits_{k=0}^{\infty} a_k\,x^k$, and therefore we get the initial values by Taylor's theorem:

$$\left(\frac{\partial^n}{\partial x^n} f\right)(0) = n!\,a_n\ .$$

In our first example we conclude

```
In[39]:= PowerSeries[E^(x^2)*Erfc[x],x]

                2 k
               x
Out[39]= Sum[----, {k, 0, Infinity}] +
                k!

                k  1 + 2 k
            -2 4  x        k!
>     Sum[------------------, {k, 0, Infinity}]
          Sqrt[Pi] (1 + 2 k)!
```

so that the first initial condition for identity (33) is given by the calculation (see [2] (7.2.7))

$$\frac{(-1)^n\, n!}{\Gamma\left(\frac{n}{2}+1\right)} = (-1)^n\, 2^n\, n!\ \mathrm{erfc}_n(0) = \frac{\partial^n}{\partial x^n}\left(e^{x^2}\,\mathrm{erfc}\,x\right)(0) = \left\{\begin{array}{ll} \frac{(2k)!}{k!} & \text{if } n = 2k\ (k \in \mathbb{N}_0) \\ -\frac{2\,k!\,4^k}{\sqrt{\pi}} & \text{if } n{=}2k{+}1\ (k \in \mathbb{N}_0) \end{array}\right. ,$$

and the second one is established similarly.

To identify the first initial values of our second example, we proceed as follows: The left hand side of (34) yields

$$\lim_{x\to 0} \frac{n!\,e^{-x}}{x}\,F_n(x) = n!\,\lim_{x\to 0}\frac{F_n(x)}{x} = n!\,F_n'(0) = -2\,n! \tag{35}$$

(see [14], (11)), whereas from the identity

$$\left(x^n\right)^{(k)}(0) = \left\{\begin{array}{ll} n! & \text{if } k = n \\ 0 & \text{otherwise} \end{array}\right.$$

and Leibniz's formula we derive for the right hand side

$$\begin{aligned} \left(e^{-2x}\,x^{n-1}\right)^{(n)}(0) &= \left(\sum_{k=0}^{n}\binom{n}{k}\left(x^{n-1}\right)^{(k)}\left(e^{-2x}\right)^{(n-k)}\right)(0) \\ &= \binom{n}{n-1}(n-1)!\,\left(e^{-2x}\right)'(0) = -2\,n!\,, \end{aligned}$$

in agreement with (35).

It is easily seen that we can always identify the initial values algorithmically by the method given if $f(n,x) = w(x)\,X(x)^n$ with a polynomial X, i. e. is of the form (7).

These results are summarized by

Algorithm 8 (Verification of identities) With Algorithms 6 and 7 identities involving Rodrigues type expressions can be verified if only symbolic derivatives $f^{(n)}$ of hypergeometric type analytic expressions f occur that have the form $f(n,x) = w(x)\,X(x)^n$ for some polynomial X.

11 Algorithmic verification of formulas involving symbolic sums

In this section we study, how identities involving symbolic sums can be established. The results depend on the following algorithm (compare [21], Maple function `cauchyproduct`):

Algorithm 9 (Find recurrence equation for symbolic sums) Let $f_n(x)$ form an admissible family, and let $s_n(x)$ denote the symbolic sum $s_n(x) := \sum\limits_{k=0}^{n} f_k(x)$. Then the following algorithm generates a recurrence equation for s_n:

(a) `re:=FindRecursion(f,k)`:
Calculate the simple recurrence equation `re` of f_k, see [9], §11.

(b) `de1:=REtoDE(re1,f,k,F,z)`:
Transfer the recurrence equation `re` into the corresponding differential equation `de1` valid for the generating function $F(z) := \sum\limits_{k=0}^{\infty} f_k(x)\,z^k$, see [9], §11.

(c) `de2:=F(z)+(z-1)*F'(z)=0`:
Let `de2` be the differential equation corresponding to the function

$$G(z) := \sum_{k=0}^{\infty} g_k z^k = \sum_{k=0}^{\infty} z^k = \frac{1}{1-z} \, .$$

(d) `de:=ProductDE(de1,de2,F,z)`:
Calculate the simple differential equation `de` corresponding to the product $H(z) := F(z)\, G(z)$, see [15], Theorem 3 (d). This differential equation has the order of `de1`.

(e) `re:=DEtoRE(de,F,z,s,n)`:
Transfer the differential equation `de` into the corresponding recurrence equation `re` for the coefficient s_n of $H(z)$, see [9], § 6.

Proof: Parts (a), (b) and (e) of the algorithm are described precisely in [9]. The rest follows from the Cauchy product representation

$$H(z) = F(z)\, G(z) = \sum_{n=0}^{\infty} \left(\sum_{k=0}^{n} f_k \, g_{n-k} \right) z^n = \sum_{n=0}^{\infty} \left(\sum_{k=0}^{n} f_k \right) z^n$$

of the product function $F(z)\, G(z)$. $\square$

As an example we consider the sum $\sum\limits_{k=0}^{n} L_k^{(\alpha)}(x)$. We get stepwise:

```
In[40]:= re=FindRecursion[LaguerreL[k,alpha,x],k]

Out[40]= (-1 + alpha + k) a[-2 + k] + (1 - alpha - 2 k + x) a[-1 + k] +

>      k a[k] == 0

In[41]:= de1=REtoDE[re,a,k,F,z]

                                                        2
Out[41]= (-1 - alpha + x + z + alpha z) F[z] + (-1 + z)  F'[z] == 0

In[42]:= de2=F[z]+(z-1)*F'[z]==0;

In[43]:= de=ProductDE[de1,de2,F,z]

                                                          2
Out[43]= (-2 - alpha + x + 2 z + alpha z) F[z] + (-1 + z)  F'[z] == 0

In[44]:= DEtoRE[de,F,z,s,n]

Out[44]= (2 + alpha + n) s[n] + (-4 - alpha - 2 n + x) s[1 + n] +

>      (2 + n) s[2 + n] == 0
```

or by a single statement

```
In[45]:= re=SymbolicSumRE[LaguerreL[k,alpha,x],k,n]

Out[45]= (2 + alpha + n) a[n] + (-4 - alpha - 2 n + x) a[1 + n] +

>     (2 + n) a[2 + n] == 0
```

and substituting n by $n - 2$

```
In[46]:= Simplify[re /. n->n-2]

Out[46]= (alpha + n) a[-2 + n] + (-alpha - 2 n + x) a[-1 + n] + n a[n] == 0
```

On the other hand, the calculation

```
In[47]:= FindRecursion[LaguerreL[n,alpha+1,x],n]

Out[47]= (alpha + n) a[-2 + n] + (-alpha - 2 n + x) a[-1 + n] + n a[n] == 0
```

shows that the left and right hand sides of the identity

$$\sum_{k=0}^{n} L_k^{(\alpha)}(x) = L_n^{(\alpha+1)}(x) \tag{36}$$

(see e. g. [24], VI (1.16)) satisfy the same recurrence equation.

In our example identity two initial values remain to be considered

$$L_0^{(\alpha)}(x) = L_0^{(\alpha+1)}(x) = 1 \qquad \text{and} \qquad L_0^{(\alpha)}(x) + L_1^{(\alpha)}(x) = L_1^{(\alpha+1)}(x) = 2 + \alpha - x$$

that trivially are established.

Thus Algorithm 9 shows the applicability of Algorithm 6 if in the expressions involved symbolic sums occur. This is summarized by

Algorithm 10 (Verification of identities) With Algorithms 6 and 9 identities involving symbolic sums can be verified. □

We like to mention that the function `FindRecursion` is successful for composite f_n as long as recurrence equations exist and are applied recursively. Here obviously no derivative rules are needed.

We note further that as a byproduct this algorithm in an obvious way can be generalized to sums $\sum\limits_{k=0}^{n} a_k\, b_{n-k}$ of the Cauchy product type. As an example, the algorithm generates the recurrence equation

$$2\,(1+2n)\,s_n - (1+n)\,s_{n+1} = 0 \tag{37}$$

for $s_n := \sum\limits_{k=0}^{n} \binom{n}{k}^2 = n!^2 \sum\limits_{k=0}^{n} \frac{1}{k!^2} \frac{1}{(n-k)!^2}$,

```
In[48]:= re=ConvolutionRESum[1/k!^2,1/k!^2,k,n]

                                         3
Out[48]= 2 (1 + 2 n) a[n] - (1 + n)  a[1 + n] == 0

In[49]:= ProductRE[re,FindRecursion[n!^2,n],a,n]

Out[49]= 2 (1 + 2 n) a[n] + (-1 - n) a[1 + n] == 0
```

compare [27]–[29].

Algorithm 9 may further be used to find a closed form representation of a symbolic sum in case the resulting term is hypergeometric:

Algorithm 11 (Closed forms of hypergeometric symbolic sums) Let $s_n := \sum\limits_{k=0}^{n} f_k$ be a hypergeometric term, i. e. $\frac{s_{n+1}}{s_n}$ be a rational function, then the following procedure generates a closed form representation for s_n:

(a) `re:=SymbolicSumRE(f,k,n)`:
Calculate the simple recurrence equation `re` of s_n using Algorithm 9.

(b) If `re` is of the hypergeometric type, then solve it by the hypergeometric coefficient formula, else apply Petkovsek's algorithm to find the hypergeometric solution s_n of `re`. □

This result should be compared with the Gosper algorithm [5]. Our procedure is an alternative decision procedure for the same purpose. Note that from the hypergeometricity of s_n the hypergeometricity of f_k follows [5], so that the first step of Algorithm 9 leads to a simple first order recurrence equation.

Applying our algorithm to our example case $s_n = \sum\limits_{k=0}^{n} \binom{n}{k}^2$, we get from (37), and the initial value $s_0 = 1$ the representation

$$s_n = 4^n \frac{\left(\frac{1}{2}\right)_n}{n!} = \frac{(2n)!}{n!^2} = \binom{2n}{n} .$$

On the other hand, for $s_n = \sum\limits_{k=0}^{n} \binom{n}{k}^3$, our procedure gives

```
In[50]:= re=ConvolutionRESum[1/k!^3,1/k!^3,k,n]

                                                  2
Out[50]= 8 a[n] + (1 + n) (16 + 21 n + 7 n ) a[1 + n] -

                          5
>      (1 + n) (2 + n)  a[2 + n] == 0

In[51]:= ProductRE[re,FindRecursion[n!^3,n],a,n]

                      2                        2                      2
Out[51]= 8 (1 + n)  a[n] + (16 + 21 n + 7 n ) a[1 + n] - (2 + n)  a[2 + n] ==

>     0
```

and by Petkovsek's algorithm it turns out that s_n is no hypergeometric term.

References

[1] Abramowitz, M.: Evaluation of the integral $\int\limits_0^\infty e^{-u^2-x/u}\,du$, J. Math. Phys. **32** (1953), 188–192.

[2] Abramowitz, M. and Stegun, I. A.: Handbook of Mathematical Functions. Dover Publ., New York, 1964.

[3] Axiom: User Guide. The Numerical Algorithms Group Ltd., 1991.

[4] Bateman, H.: The k-function, a particular case of the confluent hypergeometric function. Trans. Amer. Math. Soc. **33** (1931), 817–831.

[5] Gosper Jr., R. W.: Decision procedure for indefinite hypergeometric summation. Proc. Natl. Acad. Sci. USA **75** (1978) 40–42.

[6] Gruntz, D. and Koepf, W.: Formal power series. Konrad-Zuse-Zentrum Berlin (ZIB), Preprint SC 93-31, 1993.

[7] Hansen, E.R.: A table of series and products. Prentice-Hall, Englewood Cliffs, NJ, 1975.

[8] Hearn, A.: Reduce User's manual, Version 3.4. The RAND Corp., Santa Monica, CA, 1987.

[9] Koepf, W.: Power series in Computer Algebra. J. Symb. Comp. **13**, 1992, 581–603.

[10] Koepf, W.: Algorithmic development of power series. In: Artificial intelligence and symbolic mathematical computing, ed. von J. Calmet und J. A. Campbell, International Conference AISMC-1, Karlsruhe, Germany, August 1992, Proceedings, Lecture Notes in Computer Science **737**, Springer-Verlag, Berlin–Heidelberg, 1993, 195–213.

[11] Koepf, W.: Examples for the algorithmic calculation of formal Puiseux, Laurent and power series. SIGSAM Bulletin **27**, 1993, 20–32.

[12] Koepf, W.: A package on formal power series. Mathematica Journal, 1994, to appear. Konrad-Zuse-Zentrum Berlin (ZIB), Preprint SC 93-27.

[13] Koepf, W.: On families of iterated derivatives, in preparation.

[14] Koepf, W. and Schmersau, D.: Bounded nonvanishing functions and Bateman functions. Complex Variables, 1994, to appear.

[15] Koepf, W. and Schmersau, D.: Spaces of functions satisfying simple differential equations. Konrad-Zuse-Zentrum Berlin (ZIB), Technical Report TR 94-2, 1994.

[16] Macsyma: Reference Manual, Version 13. Macsyma Inc, Arlington, MA 02174, USA.

[17] Maple: Reference Manual, fifth edition. Watcom publications, Waterloo, 1988.

[18] Nikiforov, A. F. and Uvarov, V. B.: Special functions of mathematical physics. Translated from the Russian and with a preface by Ralph P. Boas. Birkhäuser Verlag, Basel–Boston, 1988.

[19] Petkovsek, M.: Hypergeometric solutions of linear recurrences with polynomial coefficients. J. Symbolic Comp. **14**, 1992, 243–264.

[20] Risch, R.: Algebraic properties of the elementary functions of Analysis. Amer. J. of Math. **101**, 1979, 743–759.

[21] Salvy, B. and Zimmermann, P.: GFUN: A package for the manipulation of generating and holonomic functions in one variable. Rapports Techniques **143**, INRIA, Rocquencourt, 1992.

[22] Stanley, R. P.: Differentiably finite power series. Europ. J. Combinatorics **1**, 1980, 175–188.

[23] Szegö, G.: Orthogonal Polynomials. Amer. Math. Soc. Coll. Publ. Vol. **23**, New York City, 1939.

[24] Tricomi, F. G.: Vorlesungen über Orthogonalreihen. Grundlehren der Mathematischen Wissenschaften **76**, Springer-Verlag, Berlin–Göttingen–Heidelberg, 1955.

[25] Truesdell, C.: An essay toward a unified theory of special functions. Princeton University Press, 1948.

[26] Wolfram, St.: MATHEMATICA. A system for doing mathematics by Computer. Addison-Wesley Publ. Comp., Redwood City.

[27] Zeilberger, D.: A holonomic systems approach to special functions identities. J. Comput. Appl. Math. **32**, 1990, 321-368.

[28] Zeilberger, D.: A fast algorithm for proving terminating hypergeometric identities. Discrete Math. **80**, 1990, 207-211.

[29] Zeilberger, D.: The method of creative telescoping. J. Symbolic Computation **11**, 1991, 195–204.