Solving Difference Equations whose Coefficients are not Transcendental

Manuel Kauers [1]

Research Institute for Symbolic Computation
Johannes Kepler Universität
Altenberger Straße 69
A-4040 Linz, Austria, Europe

Abstract

We consider a large class of sequences, called admissible sequences, which are defined by systems of (possibly nonlinear) difference equations. A procedure for recursively enumerating the algebraic dependencies of such sequences is presented. Also a procedure for solving linear difference equations with admissible sequences as coefficients is proposed. The methods are illustrated on some problems arising in the literature on special functions and combinatorial sequences.

Key words: Difference Equations, Algebraic Dependencies, Symbolic Summation,

1 Introduction

A difference equation of order $r \in \mathbb{N}$ is an equation of the form

$$F(u(n), u(n+1), \dots, u(n+r), n) = 0 \qquad (n \geq 1), \tag{1}$$

where $F\colon k^{r+2} \to k$ is an explicitly given function. Any sequence $u\colon \mathbb{N} \to k$ which satisfies (1) is called a solution of that equation. If F is linear in the first $r+1$ arguments, i.e., if the equation reads

$$a_0(n)u(n) + a_1(n)u(n+1) + \dots + a_r(n)u(n+r) = g(n) \quad (n \geq 1) \tag{2}$$

Email address: `manuel.kauers@risc.uni-linz.ac.at` (Manuel Kauers).
[1] Partially supported by the Austrian Science Foundation (FWF) grants F1305 and P16613-N12.

for some sequences $a_0, \dots, a_r, g\colon \mathbb{N} \to k$, then we call it a *linear difference equation.* Depending on the class of functions from which the a_i and g are chosen and on the class of functions in which solutions u are to be found, there are various known algorithms for solving linear difference equations. In the simplest case, the a_i and g are independent of n. In this case, a closed form solution in terms of exponentials can always be found by classical means [14]. Less trivial is the case where the a_i and g are rational functions in n. There are algorithms due to Abramov [1] and van Hoeij [22] which find all rational function solutions u to such equations. Petkovšek's algorithm [15] can find hypergeometric solutions u of the same type of equations. Hendriks and Singer [8] define the notion of liouvillean sequences and propose an algorithm for computing such solutions of linear difference equations with rational coefficients. Schneider [19] has got an algorithm for the case where the a_i may involve complicated expressions of nested sums and products. This algorithm finds solutions in terms of nested sums and products.

All these algorithms have in common that the coefficient sequences a_i can be written as rational functions of some basis sequences $f_1, \dots, f_m$ which are algebraically independent (transcendental). In addition, these algorithms have the feature that they find *all* closed form solutions of a given equation in a finite number of steps. In the present paper, the focus is different. We aim at covering a very large class of linear difference equations, i.e., the class of functions, from which the functions a_i and g in (2) are chosen, is very rich. It contains all the above-mentioned classes as subclasses, plus a lot of additional sequences, including in particular sequences that may obey nontrivial algebraic dependencies (Section 2). We will describe how to determine algebraic dependencies of such sequences, and how to find solutions of linear difference equations (2) whose coefficients $a_0, \dots, a_r, a$ belong to this class. As we will argue, there is no reasonable hope that decision procedures for solving such problems exist. Instead, we will therefore propose procedures that recursively enumerate the set of all algebraic dependencies, resp. of all solutions to a given equation. Alternatively, these procedures may of course also be formulated as semi-decision procedures which, in a finite number of steps, find a solution if and only if there exists one, and otherwise run forever without producing any output.

The development of our algorithms was motivated by problems arising in the literature of special functions and combinatorial sequences, especially summation problems, which could up to now not be treated by symbolic computation. The overall goal is to devise methods for problems to which the classical algorithms are not applicable. Indeed, with the methods described in this paper, it is possible to solve some such problems. Example applications involving expression like F_{2^n} (the 2^n-th Fibonacci number) and other quantities, which are defined by nonlinear difference equations, are interspersed throughout the paper. In contrast to some of the algorithms mentioned in the beginning, our methods are conceptually simple and the mathematical background they are based on is rather moderate. Familiarity with basic facts of ideal theory and Gröbner basis techniques [3] is sufficient for reading this paper.

Not only are we interested in the theoretic aspects of solving difference equations. Rather, we claim that most of the proposed procedures are actually feasibly on modern computer architectures. We are not able to justify this claim by any sort of theoretical statements about the complexity of our procedures. In fact, the worst case runtime of most algorithms is so poor that it is of little interest to actually bounds for their time and memory requirements. However, as it is often the case with algorithms related to commutative algebra, we observed that the computational cost for many examples is reasonably low, and certain identities arising in the literature could indeed be tackled by an implementation of the methods described in this article [12].

2 Admissible Sequences

A sequence is a function $f\colon \mathbb{N} \to k$, where k is some fixed computable field of characteristic zero. The set of all sequences in k with pointwise addition and multiplication forms a commutative ring and is denoted by $\mathcal{S}$. Algorithms for sequences necessarily operate only on certain subsets of $\mathcal{S}$. The sequences to which our algorithms are applicable are called *admissible sequences.* They are defined by systems of difference equations of a certain form, called *admissible systems.* Admissible systems are defined as follows.

Definition 1 *Let $f_1, \ldots, f_m\colon \mathbb{N} \to k$ be solutions of a system*

$$S = \{\mathit{diffeq}_1, \ldots, \mathit{diffeq}_m\}$$

of difference equations where each diffeq_i takes the form

$$\begin{aligned} f_i(n+r_i) = \operatorname{rat}_i\Big(&f_1(n), \quad f_1(n+1), \quad \ldots, f_1(n+r_i-1), \quad f_1(n+r_i),\\ &\vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \vdots\\ &f_{i-1}(n), f_{i-1}(n+1), \ldots, f_{i-1}(n+r_i-1), f_{i-1}(n+r_i),\\ &f_i(n), \quad f_i(n+1), \quad \ldots, f_i(n+r_i-1),\\ &\vdots \qquad\qquad\qquad\qquad \vdots\\ &f_m(n), \quad f_m(n+1), \quad \ldots, f_m(n+r_i-1)\Big)\end{aligned}$$

with a rational function rat_i *over k whose numerator or denominator is constant. (In other words, either* $\operatorname{rat}_i = p$ *or* $\operatorname{rat}_i = 1/p$ *for some polynomial p.) Then the sequences $f_1, \ldots, f_m$ are called* admissible *and S is called an* admissible system *for $f_1, \ldots, f_m$.*

In the sequel, we understand that phrases like "admissible sequences $f_1, \ldots, f_m$ are given" mean that the following data is explicitly known:

- An admissible system S which has $f_1, \ldots, f_m$ as solutions
- The initial values $f_i(j) \in k$ for $j = 1, \ldots, r_i$ $(i = 1, \ldots, m)$

Given this data, any desired value $f_i(n)$ $(n \in \mathbb{N})$ can be computed by a linear number of field operations, by starting with the initial values and applying the recurrence equations in S a suitable number of times.

Example 2 *We list some admissible sequences.*

(1) The sequence $n \mapsto n$ is admissible. It is a solution of the admissible system $\{f(n+1) = f(n) + 1\}$ with initial value $f(1) = 1$.

(2) The sequence of Legendre polynomials $P_n(x)$ [2] is admissible via the admissible system

$$\Big\{f_1(n+1) = f_1(n) + 1, f_2(n+2) = \frac{2n+3}{n+2} x f_2(n+1) - \frac{n+1}{n+2} f_2(n)\Big\}$$

with initial values $f_1(1) = 1$ (so that $f_1(n) = n$ for all n) and $f_2(1) = x$, $f_2(2) = \frac{1}{2}(3x^2 - 1)$ (so that $f_2(n) = P_n(x)$ for all n). Similarly, many other orthogonal polynomials, in fact all univariate holonomic sequences (also called P-finite sequences [24]) are admissible.

(3) The sequence of Fibonacci numbers is admissible. Moreover, the sequence $n \mapsto F_{2^n}$ is admissible. To see this, recall that the Fibonacci numbers obey the addition theorems

$$\begin{aligned} F_{p+q} &= F_{p+1}F_q + F_pF_{q+1} - F_pF_q, \\ F_{p+q+1} &= F_pF_q + F_{p+1}F_{q+1}. \end{aligned}$$

Setting $p = q = 2^n$, we find

$$\begin{aligned} F_{2^{n+1}} &= F_{2^n+2^n} = F_{2^n+1}F_{2^n} + F_{2^n}F_{2^n+1} - F_{2^n}F_{2^n}, \\ F_{2^{n+1}+1} &= F_{2^n+2^n+1} = F_{2^n}F_{2^n} + F_{2^n+1}F_{2^n+1}, \end{aligned}$$

and consequently

$$\{f_1(n+1) = 2f_2(n)f_1(n) - f_1(n)^2,\ f_2(n+1) = f_1(n)^2 + f_2(n)^2\}$$

is a suitable admissible system for specifying the sequence $n \mapsto F_{2^n}$. By a similar construction, admissible systems for $n \mapsto 2^{F_n}$, $n \mapsto F_{F_n}$, and in fact for any sequence $n \mapsto f(g(n))$ where f and g satisfy homogeneous linear recurrences with constant coefficients and the coefficients in the recurrence of g are integral, can be obtained.

(4) For some fixed $a_1, \dots, a_r \in k$, a sequence C satisfying the equation

$$C(n+r)C(n) = a_1 C(n+r-1)C(n+1) + a_2 C(n+r-2)C(n+2) + \cdots \\ \cdots + a_r C(n+r-\lfloor r/2 \rfloor)C(n+\lfloor r/2 \rfloor)$$

is an (r-)Somos sequence [20,6].

Somos sequences are admissible; a suitable admissible system is

$$\Big\{ f_1(n+r) = f_2(n)\Big(a_1 f_1(n+r-1) f_1(n+1) + \cdots \\ \cdots + a_r f_1(n+r-\lfloor r/2 \rfloor) f_1(n+\lfloor r/2 \rfloor)\Big), \\ f_2(n) = 1/f_1(n)\Big\}$$

Observe that the sequence f_2 was introduced to fulfill the requirement of Def. 1 that numerators or denominators of the rational functions on the right hand side be constant.

It should be remarked at this point that for some admissible systems, not every choice of initial values yields well-defined sequences. This is because denominators might become zero for some points. For instance, the admissible system $\{f_1(n+1) = f_1(n) + 1,\ f_2(n) = 1/f_1(n)\}$ defines two admissible sequences f_1, f_2 once the initial value $f_1(1)$ is chosen. If a negative integer is chosen for $f_1(1)$, then $f_2(n)$ is undefined at $n = -f_1(1)$. For admissible systems and initial values which are supplied as input of our algorithms, we will always assume that this situation does not occur, i.e., that the input sequences are well-defined.

New admissible sequences can be composed out of known ones by using the following closure properties of the class of admissible sequences.

Theorem 3 *[11, Thms. 3.5, 3.7] Let f and g be admissible sequences, $a \in \mathbb{N}$ and $\alpha \in k$. Then*

(1) αf, $f + g$ and $f \cdot g$ and, if $g(n) \neq 0$ for all n, f/g are admissible,
(2) $n \mapsto \sum_{i=1}^n f(i)$, $n \mapsto \prod_{i=1}^n f(i)$, and, if $f(n) \neq 0 \neq g(n)$ for all n,

$$n \mapsto f(1) + \cfrac{g(2)}{f(2) + \cfrac{g(3)}{\cdots + \cfrac{g(n)}{f(n)}}}$$

are admissible,
(3) $n \mapsto f(n+a)$, $n \mapsto f(an)$, $n \mapsto f(\lfloor n/a \rfloor)$ are admissible,

and admissible systems for these sequences can be effectively computed from admissible systems for f and g.

Using this theorem, it is possible to automatically transform an expression involving sums and products into a corresponding defining admissible system. Rather than giving a formal proof, we illustrate the theorem with an example.

Example 4 *The expression*

$$\sum_{i=1}^{n} \frac{1}{i} \prod_{j=1}^{i} \frac{F_j^2 - 1}{F_j^2 + 1}$$

constitutes an admissible sequence. A suitable admissible system can be constructed by first considering the innermost subexpressions and then building up the entire expression step by step:

$$\begin{aligned}
\{f_1(n+2) &= f_1(n+1) + f_1(n), && (\, f_1(n) \sim F_n \,)\\
f_2(n) &= 1/(f_1(n)^2 + 1), && (\, f_2(n) \sim 1/(F_n^2 + 1)\,)\\
f_3(n+1) &= f_3(n) f_2(n+1)(f_1(n+1)^2 - 1), && (\, f_3(n) \sim \Pi \,)\\
f_4(n+1) &= f_4(n) + 1, && (\, f_4(n) \sim n \,)\\
f_5(n) &= 1/f_4(n), && (\, f_5(n) \sim 1/n \,)\\
f_6(n+1) &= f_6(n) + f_4(n+1) f_3(n+1)\,\} && (\, f_6(n) \sim \Sigma \,)
\end{aligned}$$

It is easily seen that the class of admissible sequences properly includes many of the classes known for the algorithms mentioned in the introduction, such as the (univariate) holonomic sequences and Karr's $\Pi\Sigma$ sequences [9], for instance. In addition, sequences like $n \mapsto F_{F_n}$, which are admissible, do not belong to any class of sequences that can be handled by a known algorithm. Yet there are — of course — still sequences which are not covered. For instance, we can show [11, Section 4.3] that the sequences $n \mapsto (-1)^{\lfloor \log n \rfloor}$ and $n \mapsto 2^{n!}$ cannot be defined via an admissible system.

3 Reduction to Polynomial Algebra

Let $f_1, \ldots, f_m$ be admissible sequences, and consider the ring homomorphism $\phi\colon k[x_1, \ldots, x_m] \to \mathcal{S}$ that maps x_i to f_i and each $c \in k$ to the constant sequence $(c, c, c, \ldots)$. The homomorphism theorem asserts that the factor ring $k[x_1, \ldots, x_m]/\ker\phi$ is isomorphic to $\operatorname{im}\phi$, which is the smallest subring of $\mathcal{S}$ containing $f_1, \ldots, f_m$ and all constant sequences. As $\ker\phi$ is just a polynomial ideal, the computational treatment of the ring $k[x_1, \ldots, x_m]/\ker\phi$ is well understood. The theory of Gröbner bases [3] provides an algebraic framework for solving problems in such domains. By the isomorphism, there is a one-to-one correspondence between $k[x_1, \ldots, x_m]/\ker\phi$ and $\operatorname{im}\phi$, so that results obtained in the former ring (by Gröbner basis or other means) can be directly translated into results in the latter ring, which is in our interest.

There is only a slight obstacle here: in order to perform computations in the ring $k[x_1, \ldots, x_m]/\ker\phi$, we have to know the ideal $\ker\phi$ explicitly, i.e., we have to know a list of ideal generators $p_1, \ldots, p_s \in k[x_1, \ldots, x_m]$ such that $\ker\phi = \langle p_1, \ldots, p_s \rangle$. Ideally, we would like to be able to compute such generators $p_1, \ldots, p_s$ from a given admissible system and initial values for the $f_1, \ldots, f_m$. No algorithms are known for this problem. However, in an earlier paper [10] we have shown that the membership problem for $\ker\phi$ (given $p \in k[x_1, \ldots, x_m]$, decide $p \in \ker\phi$) is decidable. The remarkable aspect of this algorithm is that generators of $\ker\phi$ need not be known; only an admissible system for the $f_1, \ldots, f_m$ and initial values are required as input.

Note that $p \in \ker\phi$ just means that $F := p(f_1, \ldots, f_m)$ is the zero sequence. Regardless of whether this is the case, F is for sure an admissible sequence, because the f_i are admissible (Thm. 3). Our decision procedure [10] is hence an algorithm for deciding zero equivalence of admissible sequences. It works by constructing an induction proof for the sequence F to be zero. In a first phase, it computes a number $N \in \mathbb{N}$ with the property that

$$\forall n \in \mathbb{N} : P(n) = P(n+1) = \cdots = P(n+N-1) = 0 \Rightarrow P(n+N) = 0.$$

This N provides the induction step. In a second phase, the algorithm evaluates $P(1), P(2), \ldots, P(N)$ and determines whether they are all zero. (It is assumed that the ground field k is such that zero equivalence in the ground field can be decided.) This either supplies the base of the induction and thus a decision that indeed $F \equiv 0$, or it leads to an index $n \in \{1, \ldots, N\}$ with $P(n) \neq 0$ and thus a decision that $F \not\equiv 0$. The algorithm reveals the following theorem.

Theorem 5 *[10,11] There exists an algorithm which for a given admissible system S and $p \in k[x_1, \ldots, x_m]$ computes a number $N \in \mathbb{N}$ such that for all solutions $f_1, \ldots, f_m$ of S:*

$$\forall n \in \mathbb{N} : P(n) = 0 \iff P(1) = P(2) = \cdots = P(N) = 0,$$

where $P = p(f_1, \ldots, f_m)$. □

As an example, it can be shown with this algorithm that the expression of Example 4 represents the zero sequence. It must be remarked that the number N asserted by the above theorem only depends on the admissible system S, but not on the initial values of the $f_1, \ldots, f_m$. For further details about this algorithm and more examples, we refer to [10,11].

4 Recursive Enumeration of a Basis for the Kernel

Our goal is to find elements of the ideal $\ker\phi$. There is little hope that an algorithm could be found which computes a basis of $\ker\phi$ from an admissible system and initial values for the $f_1,\dots,f_m$. This is because of the following reduction.

Theorem 6 *If there exists an algorithm which for given admissible sequences $f_1,\dots,f_m$ computes a basis of $\ker\phi$, then there exists an algorithm which for a given admissible sequence f decides whether there exists an index n with $f(n)=0$, and, if yes, delivers the smallest such n.*

Proof. Let f be an admissible sequence, say f is defined via an admissible system for $f_1,\dots,f_m$ and $f=f_i$. If we define $f_0(n):=n$ and $f_{m+1}(n)=\prod_{j=1}^n f_i(j)$, then f_0 and f_{m+1} are admissible, too, by Theorem 3.

Let $\phi\colon k[x_0,\dots,x_{m+1}]\to\mathcal{S}$ be defined via $\phi(x_i)=f_i$ $(i=0,\dots,m+1)$. Then $f(n_0)=0$ for some $n_0\in\mathbb{N}$ if and only if

$$x_{m+1}(x_0-1)(x_0-2)\cdots(x_0-n_0)\in\ker\phi.$$

Suppose now that a basis $\ker\phi=\langle p_1,\dots,p_m\rangle$ is known. Then we can compute a Gröbner basis G for this ideal with respect to the lexicographic order $x_{m+1}>x_m>\cdots>x_1>x_0$. This G contains the polynomial $x_{m+1}(x_0-1)(x_0-2)\cdots(x_0-n_0)$ indicating the smallest root n_0 of f — or no polynomial of this form if f does not have any roots. □

According to this theorem, the computation of a basis for $\ker\phi$ is at least as difficult as finding a root of an admissible sequence. Unfortunately, finding roots of sequences is a very difficult problem. Already for the class of sequences satisfying homogeneous linear recurrence equations with constant coefficients (also called C-finite sequences [24]), it is an open problem whether the question about the existence of a root is decidable [5, Sec. 2.3]. As this class is only a ridiculously small subclass of the class of admissible sequences, it does not seem reasonable to look for an algorithm that is capable of solving this problem for the much larger class. We want to describe instead a procedure by which a basis of the kernel $\ker\phi$ can be recursively enumerated.

4.1 Linear Dependencies of Admissible Sequences

As a subalgorithm for the enumeration procedure, we need an algorithm for computing linear dependencies of admissible sequences. Given a finite set of polynomials $P=\{p_1,\dots,p_l\}\subseteq k[x_1,\dots,x_m]$, this algorithm computes a basis of the vector space

$$V_P := (p_1 k + p_2 k + \cdots + p_l k) \cap \ker \phi \subseteq k[x_1, \ldots, x_m].$$

The algorithm proceeds by making an undetermined ansatz, obtaining candidates for the desired relations by comparing them with the initial values and solving a linear system. The candidates can be validated by the algorithm of Theorem 5. If necessary, the ansatz is refined more and more, until all candidates actually belong to the kernel.

Algorithm 7 ***Input:*** *Admissible sequences* $f_1, \ldots, f_m \colon \mathbb{N} \to k$, *a set* $P = \{p_1, \ldots, p_l\} \subseteq k[x_1, \ldots, x_m]$
Output: *A basis of the vector space* V_P.

1 *Define* $g_i := p_i(f_1, \ldots, f_m)$ $(i = 1, \ldots, m)$
2 $N = l$;
3 **repeat**
4 $\quad N = N + 1$
5 $\quad$ *Compute a basis* $B \subseteq k^l$ *for the solution space of the linear system*

$$\begin{pmatrix} g_1(1) & \cdots & g_l(1) \\ \vdots & \ddots & \vdots \\ g_1(N) & \cdots & g_l(N) \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_l \end{pmatrix} = 0$$

6 **until** $b_1 g_1(n) + b_2 g_2(n) + \cdots + b_l g_l(n) = 0$ $(n \in \mathbb{N})$ *for all* $(b_1, \ldots, b_l) \in B$
7 **return** $\{\, b_1 p_1 + b_2 p_2 + \cdots + b_l p_l : (b_1, \ldots, b_l) \in B \,\}$

Observe that the g_i are admissible sequences by Theorem 3.(1). Therefore, the condition in line 6 can be decided according to Thereom 5.

Theorem 8 *Algorithm 7 is correct, i.e., if B is the set of polynomials returned by the algorithm, then B is a basis of the vector space* V_P.

Proof. Obviously, each element of B belongs to V_P by line 6, and by construction the elements of B are linearly independent. It only remains to show that every $c_1 p_1 + \cdots + c_l p_l \in V_P$ is a linear combination of the vectors in B. For every vector $c_1 p_1 + \cdots + c_l p_l \in V_P$ the identity

$$c_1 g_1(n) + \cdots + c_l g_l(n) = 0$$

holds for all $n \geq 1$, by definition of V_P. In particular it holds for $n = 1, \ldots, N$, and hence $(c_1, \ldots, c_l)$ belongs to the solution space of the linear systems in line 5. □

Theorem 9 *Algorithm 7 terminates, i.e., in the notation of the algorithm, for sufficiently large N, all elements of B will give rise to kernel elements.*

Proof. Let S be an admissible system defining $f_1, \ldots, f_m$, and consider the admissible system

$$S' := S \cup \{ f_{m+1}(n+1) = f_{m+1}(n), \ldots, f_{m+l}(n+1) = f_{m+l}(n) \}$$

for $f_1, \ldots, f_{m+l}$. Let $P := f_{m+1}p_1(f_1, \ldots, f_m) + \cdots + f_{m+l}p_l(f_1, \ldots, f_m)$. By Theorem 5, there exists a number $N \in \mathbb{N}$ such that $P \equiv 0$ if and only if $P(1) = P(2) = \cdots = P(N) = 0$. This N bounds the number of iterations in the loop in Algorithm 7. □

4.2 Algebraic Dependencies of Admissible Sequences

It is a simple matter to extend Algorithm 7 to the desired enumeration procedure: just apply the algorithm in turn to find all linear dependencies of all the polynomials with total degree $d = 1, 2, 3, \ldots$. The union of the outputs for all $d \in \mathbb{N}$ is obviously a k vector space basis for $\ker \phi$. Unless $\ker \phi = \{0\}$, this basis will be infinite. The vector space basis is also an ideal basis, but a rather redundant one. In order to obtain an irredundand ideal basis, we should restrict the set P in the input of Algorithm 7 in such a way that solutions of the linear system are not already consequences of the dependencies accumulated for degrees smaller than d. This can be done as follows.

Procedure 10 ***Input:*** *Admissible sequences* $f_1, \ldots, f_m \colon \mathbb{N} \to k$
Output: *An ideal basis of* $\ker \phi$

$G = \emptyset; d = 0$
<u>**repeat**</u>
  Let P be a vector space basis of

$$\{ p \in k[x_1, \ldots, x_m] : \deg p \leq d \}$$

  Delete from P all elements p with $\mathrm{LT}(g) \mid \mathrm{LT}(p)$ *for some* $g \in G$
  Apply Algorithm 7 to $f_1, \ldots, f_m$ and P, obtaining B
  Output the elements of B
  $G = \text{GröbnerBasis}(G \cup B)$
  $d = d + 1$

To be specific, assume that $k[x_1, \ldots, x_m]$ is equipped with a total degree term order. Any other admissible term order [3, Def. 4.59] could be taken instead. By $\mathrm{LT}(p)$ we mean the leading term of a polynomial $p \in k[x_1, \ldots, x_m]$ with respect to that order.

Theorem 11 *Procedure 10 is correct, i.e., the polynomials it outputs generate* $\ker\phi$ *as an ideal.*

Proof. Without line 4, the Theorem would be evident. We have to show that no relations are lost in line 4. In other words, if $\mathfrak{a}$ denotes the ideal generated by the output of the procedure, we have to show that $\mathfrak{a} \subseteq \ker\phi$.

Suppose for the contrary that there exists $p \in \ker\phi \setminus \mathfrak{a}$. Then, because Algorithm 7 is complete, $\mathrm{LT}(p)$ must be a multiple of $\mathrm{LT}(a)$ for some $a \in \mathfrak{a}$. The leading term of $p' := p - \frac{\mathrm{LC}(p)}{\mathrm{LC}(a)}a$ (with LC being the leading coefficient) is smaller than that of p, and p' also belongs to $\ker\phi \setminus \mathfrak{a}$, because $p \notin \mathfrak{a}$ and $a \in \mathfrak{a} \subseteq \ker\phi$. Repeating the argument, we find $p'' \in \ker\phi \setminus \mathfrak{a}$ with a leading term smaller than that of p', and so on. This leads to an infinite descending chain of terms $\mathrm{LT}(p), \mathrm{LT}(p'), \mathrm{LT}(p''), \dots$, which by the admissibility of the term order cannot exist. $\square$

Example 12 *Consider the sequences* f_1, f_2, f_3 *defined by*

$$f_1(n) = F_{2^{n+a}}, \quad f_2(n) = F_{2^{n+a}+1}, \quad f_3(n) = \sum_{k=0}^{n} \frac{1}{F_{2^{a+k}}}.$$

Applying Procedure 10, we find no algebraic dependencies of total degrees $0, 1, 2$ *between these sequences, i.e., there does not exists a polynomial* $p \in \mathbb{Q}[x, y, z]$ *of total degree at most* 2 *with* $p(f_1, f_2, f_3) \equiv 0$*. For degree* 3*, the procedure delivers the relations*

$$\begin{aligned} 0 &= F_{2^{n+a}} F_{2^{n+a}+1}^2 - F_{2^{n+a}+1}^3 + F_{2^{n+a}+1}^2 + F_{2^{n+a}}^2 F_{2^{n+a}+1} \\ &\quad - F_{2^{n+a}} F_{2^{n+a}+1} + F_{2^{n+a}+1} - F_{2^{n+a}}^2 - 1, \\ 0 &= F_{2^{n+a}}^3 - 2F_{2^{n+a}+1}^2 F_{2^{n+a}} + F_{2^{n+a}} + F_{2^{n+a}+1}^3 - F_{2^{n+a}+1} \end{aligned}$$

therefore $\langle xy^2 - y^3 + +y^2 + x^2y - xy + y - x^2 - 1, x^3 - 2y^2x + x + y^3 - y \rangle \subseteq \ker\phi$*, with* $\phi\colon \mathbb{Q}[x, y, z] \to \mathcal{S}$ *such that* $\phi(x) = f_1, \phi(y) = f_2, \phi(z) = f_3$*. For total degree* $4, 5, 6, \dots, 10$*, there are no further relations.*

Procedure 10 does not terminate. However, after a finite number of steps, it will have output a complete basis of $\ker\phi$. This follows from Hilbert's basis theorem: If $\mathfrak{a}_d$ denotes the ideal generated by the output of the first d iterations ($d = 0, 1, 2, \dots$), then

$$\mathfrak{a}_0 \subseteq \mathfrak{a}_1 \subseteq \mathfrak{a}_2 \subseteq \cdots$$

is an infinite ascending chain of polynomial ideals, so there must be some index d_0 such that $\mathfrak{a}_d = \mathfrak{a}_{d_0}$ for all $d \geq d_0$. As the procedure in each iteration $d > 0$ only outputs elements of $\mathfrak{a}_d \setminus \mathfrak{a}_{d-1}$ (this is easy to see), it follows that for $d \geq d_0$, no further output will happen. What misses for a full algorithm is a way to compute a suitable upper bound for the value d_0.

A finite algorithm for computing a basis can be obtained by restricting the attention to smaller classes of sequences. For instance, Karr's summation algorithm [9] includes as a subroutine an algorithm for computing the algebraic dependencies of sequences which can be expressed in terms of nested sums and products. Also the algebraic dependencies of sequences which satisfy homogeneous linear difference equations with constant coefficients can be effectively computed [13,11]. It would be desirable to further investigate for which classes of sequences the algebraic dependencies are computable.

5 Applications

The procedures introduced in the previous section enable us to solve certain problems appearing in the literature of special functions and combinatorial sequences automatically. In this section, we want to illustrate some applications with concrete examples.

Example 13 *Consider the* 4*-Somos sequence* $C(n)$ *defined by*

$$\begin{aligned} &C(n+2) = (C(n-1)C(n+1) + C(n)^2)/C(n-2) \qquad (n \in \mathbb{Z}), \\ &C(-2) = C(1) = C(0) = C(1) = 1. \end{aligned}$$

It is of interest to know whether $C(n)$ *also satisfies a* r*-Somos recurrence for some* $r \neq 4$ *(cf. [21]).*

For any given r*, say* $r = 8$*, this question can be answered using Algorithm 7: compute a vector space basis* $\{b_1, \ldots, b_s\}$ *of*

$$\ker \phi \cap \{\, p \in \mathbb{Q}[x_{-r}, x_{-r+1}, \ldots, x_r] : \deg p \leq 2 \,\},$$

where $\phi\colon Q[x_{-r}, \ldots, x_r] \to \mathcal{S}$ *maps* x_i *to the sequence* $n \mapsto C(n+i)$*. For* $r = 8$*, this basis is lengthy and not reproduced here.*

We have to find out whether the b_i *can be combined to a relation of the desired form. One way to do so is to make an ansatz*

$$a_4 C(n-4)C(n+4) + a_3 C(n-3)C(n+3) + \cdots + a_0 C(n)^2 = 0$$

for the coefficients a_i*, compute the normal form of the polynomial* $a_4 x_{-4} x_4 + a_3 x_{-3} x_3 + a_2 x_{-2} x_2 + a_1 x_{-1} x_1 + a_0 x_0^2$ *with respect to the ideal* $\langle b_1, \ldots, b_s \rangle$ *(and some term order), equate the coefficients of that normal form to zero and solve the resulting linear system for the* a_i*. The solutions of this system are precisely the desired values for the coefficients.*

In this way, we have found the relations

$$\begin{aligned}C(n+2)C(n-2) &= C(n+1)C(n-1)+C(n)^2\\ C(n+3)C(n-3) &= C(n+1)C(n-1)+5C(n)^2\\ C(n+4)C(n-4) &= 25C(n+1)C(n-1)-4C(n)^2,\end{aligned}$$

and by an analogous ansatz for odd r *the relations*

$$\begin{aligned}C(n+3)C(n-2) &= 5C(n+1)C(n)-C(n+2)C(n-1)\\ C(n+4)C(n-3) &= C(n+1)C(n-1)+5C(n)^2.\end{aligned}$$

The first three relations were also given by van der Poorten [21], the last one is new. By Theorem 8, we can be sure that every other r*-Somos recurrence of* C *for* $r \leq 8$ *is a linear combinations of those given above.*

Example 14 *Certain nonlinear difference equations can be solved using Procedure 10. For instance, Rabinowitz [17] has asked for a solution of*

$$u(n+1)=\frac{3u(n)+1}{5u(n)+3}\quad (n\geq 1),\qquad u(1)=1$$

in terms of Fibonacci numbers. If there exists a rational function $r=p/q$ *with* $p,q\in\mathbb{Q}[x,y]$ *such that* $u(n)=r(F_n,F_{n+1})$ *for all* $n\geq 1$*, then* $q(x,y)z-p(x,y)\in\ker\phi$*, where* $\phi\colon\mathbb{Q}[x,y,z]\to\mathcal{S}$ *maps* x *to the Fibonacci sequence* F*,* y *to the shifted Fibonacci sequence* $n\mapsto F_{n+1}$*, and* z *to* u*.*

In order to find a solution, we apply Procedure 10. After each iteration, we compute a lexicographic Gröbner basis G *of* $\mathfrak{a}$ *with respect to* $z>y>x$ *and check whether* G *contains a polynomial linear in* z*. Each such polynomial supplies a solution, and if no such polynomial appears in* G *then no such polynomial is contained in* $\mathfrak{a}$*, and we increase the degree.*

This procedure will eventually reveal any solution of the difference equation in terms of Fibonacci numbers—if such a solution exists at all. Otherwise, the procedure will run forever. In the present situation, we find

$$u(n)=-\frac{2F_n^2-2F_nF_{n+1}+F_{n+1}^2}{4F_n^2-6F_nF_{n+1}+F_{n+1}^2}\qquad (n\geq 1).$$

By leaving the initial value $u(1)$ *symbolic during the computation, the more general solution*

$$u(n)=\frac{(3-7u(1))F_n^2-4(2u(1)-1)F_nF_{n+1}+(1-3u(1))F_{n+1}^2}{(15u(1)-7)F_n^2+4(2-5u(1))F_nF_{n+1}+(5u(1)-3)F_{n+1}^2}$$

can be found.

Example 15 *Linear difference equation can be treated as explained for non-linear equations in the previous example. This includes indefinite summation as a special case. The important aspect here is that any admissible sequence may occur in the summand. Identities like*

$$\sum_{k=0}^{n} \frac{1}{F_{2^k}} = 4 - \frac{F_{2^n+1}}{F_{2^n}}, \quad \sum_{k=0}^{n} \frac{1}{F_{3\cdot 2^k}} = \frac{9}{4} - \frac{F_{3\cdot 2^n+1}}{F_{3\cdot 2^n}} \qquad (n \geq 1)$$

[7, Ex. 6.61] can thus be found automatically. In contrast, no closed form for $\sum_{k=0}^{n} 1/F_{2^{k+a}}$ *in terms of* $F_{2^{n+a}}$ *and* $F_{2^{n+a}+1}$ *is found (cf. Example 12).*

6 Linear Difference Equations

Linear difference equations deserve special attention because of their importance in practice. Although we could find solutions of linear difference equations by means of Procedure 10 just as explained before for nonlinear difference equations, we would like to describe an alternative method for this special case. Let us consider an equation of the form

$$a_r(n)u(n+r) + a_{r-1}(n)u(n+r-1) + \cdots + a_0(n)u(n) = g(n), \tag{3}$$

where $a_0, \dots, a_r, g$ are known admissible sequences and u is unknown. If a_r has finitely many roots only, then u is determined uniquely by finitely many initial values, and in particular every solution is admissible. Otherwise, if $a_r(n)$ has infinitely many solutions, then the equation has a continuum of solutions.

What interests us here is not the general solution of equation 3, but solutions of a prescribed form. We will assume that admissible sequences $f_1, \dots, f_m$ are given and that solutions of (3) are to be computed which have the form $p(f_1, \dots, f_m)$ for some polynomial p.

6.1 The Homogeneous Equation

Let us first consider the case of a homogeneous equation, $g(n) = 0$. In order to find polynomials p such that $u := p(f_1, \dots, f_m)$ satisfies (3), we first use Procedure 10 to compute generators of the ideal $\ker \phi$ where

$$\phi\colon \underbrace{k[x_{1,0}, \dots, x_{m,0}, \dots\dots, x_{1,r}, \dots, x_{m,r}, y_0, \dots, y_r]}_{=:R} \to \mathcal{S}$$

maps $x_{i,j}$ to $n \mapsto f_i(n+j)$, y_i to $n \mapsto a_i(n)$, and constants to constants. Assuming that $\ker \phi$ is known, we then compute a basis of the syzygy module

$$S := \operatorname{Syz}(y_0, \ldots, y_r) = \{\, (p_0, \ldots, p_r) : p_0 y_0 + \cdots + p_r y_r = 0 \,\} \subseteq (R/\ker \phi)^{r+1}.$$

It is well known how to compute the syzygy module over a polynomial ring [3, Sec. 6.1], and it is straightforward to generalize this algorithm to the case of a factor ring $k[X]/\mathfrak{a}$ [11, Thm. 2.9]. (For simplicity of notation, we will not distinguish polynomials and their residue classes modulo $\ker \phi$.) Now observe that for every polynomial $p \in k[x_1, \ldots, x_m]$ we have

$$\begin{gathered} p(f_1(n), \ldots, f_m(n)) \text{ solves (3)} \\ \Longleftrightarrow \\ \big(p(x_{1,0}, \ldots, x_{m,0}), \ldots\ldots, p(x_{1,r}, \ldots, x_{m,r})\big) \in S \end{gathered} \tag{4}$$

Hence, we can find polynomials p with the desired property as follows. If we make an ansatz $p = \sum_{i_1,\ldots,i_m} a_{i_1,\ldots,i_m} x_1^{i_1} \cdots x_m^{i_m}$ for the solution polynomial and compute the normal form of the vector

$$\big(p(x_{1,0}, \ldots, x_{m,0}), \ldots\ldots, p(x_{1,r}, \ldots, x_{m,r})\big)$$

for this general p with respect to a Gröbner basis of S, then we will end up with some vector $(\bar{p}_0, \ldots, \bar{p}_r)$ where each $\bar{p}_i$ is a polynomial whose coefficients are linear combinations of the as yet undetermined $a_{i_1,\ldots,i_m}$. The ansatz polynomial p represents a solution precisely for those values $a_{i_1,\ldots,i_m}$ that make all $\bar{p}_i$ vanish, because of (4) and the fact that normal forms are zero precisely for vectors which belong to the module. Comparing the coefficients of the $\bar{p}_i$ to zero gives rise to a linear system over k for the coefficients $a_{i_1,\ldots,i_m}$ which can be solved.

Algorithm 16 ***Input:*** *A Göbner basis G of a module $S \subseteq (R/\mathfrak{a})^s$, a set $P = \{p_1, \ldots, p_l\} \subseteq (R/\mathfrak{a})^s$*
Output: *A basis for the vector space of all linear combinations p of $p_1, \ldots, p_l$ with $p \in S$*

Make an ansatz $p = a_1 p_1 + \cdots + a_l p_l$
Compute the normal form $(q_1, \ldots, q_s)$ of p w.r.t. G
Let $c_i(a_1, \ldots, a_l)$ $(i \in I)$ be the coefficients of $q_0, \ldots, q_s$
Compute a basis B of the space $\{\, (a_1, \ldots, a_l) : c_i(a_1, \ldots, a_l) = 0\ (i \in I) \,\}$
return $\{\, a_1 p_1 + \cdots + a_l p_l : (a_1, \ldots, a_l) \in B \,\}$

Termination of this algorithm is obvious, and its correctness follows from the discussion above. Applying the algorithm in turn to bigger and bigger ansatz polynomials, we obtain a procedure that recursively enumerates a basis for the solution space of (3). We cannot hope for a termination criterion (such as, e.g., a degree bound) here either, because the solution space may have infinite dimension. As in Procedure 10, we can discard leading terms to avoid redundant solutions and to keep the linear systems small. Cancellation of leading terms also ensures the output solutions are linearly independent.

Procedure 17 ***Input:*** *Admissible sequences* $a_0, \dots, a_r$ *and* $f_1, \dots, f_m$ *and a basis of* $\ker \phi$ *with* $\phi\colon R \to \mathcal{S}$ *as defined above.*
Output: *A basis of the vector space of all solutions* u *of*

$$a_0(n)u(n) + \cdots + a_r(n)u(n+r) = 0$$

which depend polynomially on $f_1, \dots, f_m$*.*

```
 1  B = ∅; d = 0
 2  Let G be a Gröbner basis of ker φ
 3  Let S be a Gröbner basis of Syz(y_0, ..., y_r) ⊆ (R/ker φ)^(r+1)
 4  repeat
 5     Let P be a vector space basis of { p ∈ k[x_1, ..., x_m] : deg p ≤ d }
 6     Delete from P all elements p with Lt(g) | Lt(p) for some g ∈ G
 7     Delete from P all elements p with Lt(b) = Lt(p) for some b ∈ B
 8     P := { (p(x_{1,0}, ..., x_{m,0}), ..., p(x_{1,r}, ..., x_{m,r})) : p ∈ P }
 9     Apply Algorithm 16 to P and S, obtaining B_0
10     B_0 := { q_0 : (q_0, ..., q_r) ∈ B_0 }
11     Output the elements of B_0
12     B := B ∪ B_0; d = d + 1
```

Theorem 18 *Procedure 17 is correct, i.e., its output constitutes a vector space basis of the* k *vector space of all solutions* u *of* (3) *which can be written polynomially in* $f_1, \dots, f_m$*.*

Proof. First of all, it is clear that every output polynomial really gives rise to a solution. We have to show that (a) no solutions are overlooked due to lines 6 and 7 and (b) the output solutions are linearly independent over k.

(a) Let p be a polynomial that corresponds to a solution u of the difference equation. The polynomial p is equivalent modulo $\ker \phi$ to a polynomial p' that does not contain terms which are multiples of leading terms in G. This polynomial p' corresponds to the same solution u so it suffices to take the terms into account that may possibly occur in p'.

Secondly, if a solution polynomial p involves a term τ which appears as a leading term of some solution $b \in B$ which was found before, then $p' := p - \alpha b$ for a suitable constant $\alpha \in k$ is another solution which does not involve τ. Restricting the ansatz such that only p' is found is just fine, because p is a linear combination of p' and b.

(b) Induction to d. For $d = 0$, $B = \emptyset$ is linearly independent. Now suppose that B is linearly independent at iteration d and assume $b_1, \dots, b_v \in B$, $c_1, \dots, c_w \in B_0$, and $\beta_1, \dots, \beta_v, \gamma_1, \dots, \gamma_w \in k$ are such that

$$\beta_1 b_1 + \cdots + \beta_v b_v + \gamma_1 c_1 + \cdots + \gamma_w c_w = 0. \tag{5}$$

If $\beta_1 b_1 + \cdots + \beta_v b_v$ is not the zero polynomial, then we must have

$$\mathrm{LT}(\beta_1 b_1 + \cdots + \beta_v b_v) = \mathrm{LT}(\gamma_1 c_1 + \cdots + \gamma_w c_w),$$

which is excluded by line 7. It follows that $\beta_1 b_1 + \cdots + \beta_v b_v = 0$, and hence by (5) also $\gamma_1 c_1 + \cdots + \gamma_w c_w = 0$. Now using the linear independence of B and B_0, respectively, we obtain $\beta_1 = \cdots = \beta_v = \gamma_1 = \cdots = \gamma_w = 0$, as desired. □

Procedure 17 requires knowledge of $\ker \phi$ as input, but we only know a recursive enumeration procedure for computing $\ker \phi$ (Procedure 10). If in practice that procedure is aborted after a while, then it is not clear whether the ideal $\mathfrak{a}$ generated by the output produced before abortion already generates the whole ideal $\ker \phi$. It is therefore interesting to know to what extend Procedure 17 remains correct if it is applied to some ideal $\mathfrak{a} \subsetneq \ker \phi$ in place of $\ker \phi$. It is quite easy to see that its output will still be correct and complete, but the output might be redundant. The sequence of polynomials it produces will also continue to be linearly independent, but this need no longer be true for the sequence of solutions u that these polynomials represent.

Example 19 *Procedure 17 applied to the difference equation*

$$(F_n - 2F_{n+1})u(n+2) + (3F_n + 2F_{n+1})u(n+1) - F_n u(n) = 0$$

with the assumption $\ker \phi = \{0\}$ *gives the infinite output*

$$\begin{array}{l} F_n^2 \\ F_n^2(F_n^2 + F_nF_{n+1} - F_{n+1}^2)^2 \\ F_n^2(F_n^2 + F_nF_{n+1} - F_{n+1}^2)^4 \\ F_n^2(F_n^2 + F_nF_{n+1} - F_{n+1}^2)^6 \\ F_n^2(F_n^2 + F_nF_{n+1} - F_{n+1}^2)^8 \\ \vdots \end{array}$$

All these solutions are correct. However, they are not linearly independent as sequences. Indeed, because of the identity $(F_n^2 + F_nF_{n+1} - F_{n+1}^2)^2 = 1$ *they all represent the same solution. If we take* $\ker \phi = \langle (x_0^2 + x_0x_1 + x_1^2)^2 - 1\rangle$, *assuming that* x_0 *and* x_1 *are the variables encoding* F_n *and* F_{n+1}, *respectively, we get the single solution*

$$F_n^2$$

as output. There ought to be a second solution to the equation, linearly independent of F_n^2. This second solution is

$$F_n^2 \sum_{k=1}^{n-1} \frac{1}{F_k^2 F_{k+1}^2} \prod_{i=2}^{k} \frac{F_{i-1}}{3F_i - F_{i+1}}$$

and cannot be expressed as a rational function in F_n and F_{n+1}. This is the reason why only one solutions is output by Procedure 17.

6.2 The Inhomogeneous Equation

Extension of Procedure 17 to the inhomogeneous equation (3) is straightforward. If an additional variable z is introduced to represent the inhomogeneous part $g(n)$, then we have

$$p(f_1(n), \ldots, f_m(n)) \text{ solves } (3)$$
$$\Longleftrightarrow$$
$$\Big(p(x_{1,0}, \ldots, x_{m,0}), \ldots\ldots, p(x_{1,r}, \ldots, x_{m,r}), -1\Big) \in \mathrm{Syz}(y_0, \ldots, y_r, z)$$

Modifying lines 3 and 8 of Procedure 17 accordingly, we obtain a method to find the solutions of the inhomogeneous equation.

This leads us to an alternative procedure for indefinite summation of admissible sequences. In order to find a closed form for $\sum_{k=1}^{n} f(k)$ in terms of some other given admissible sequences $f_1(n), \ldots, f_m(n)$, solve the telescoping equation

$$u(n+1) - u(n) = f(n+1)$$

using the inhomogeneous extension of Procedure 17. If $u(n)$ is a solution, then $\sum_{k=1}^{n} f(k) = u(n) - u(0)$.

Example 20 *For the Legendre polynomials $P_n(x)$ we find the summation identity*

$$\sum_{k=0}^{n} (2k+1) P_k(x) P_k(y) = \frac{n+1}{x-y} (P_n(y) P_{n+1}(x) - P_n(x) P_{n+1}(y))$$

by solving the telescoping equation

$$u(n+1) - u(n) = (2n+3) P_{n+1}(x) P_{n+1}(y)$$

in terms of $n, P_n(x), P_{n+1}(x), P_n(y), P_{n+1}(y)$.

As opposed to an indefinite sum $\sum_{k=1}^{n} f(k)$, by a definite sum we understand a sum $\sum_{k=1}^{n} f(n,k)$ where the summand sequence $f(n,k)$ need not be independent of the summation bound n. Most summation problems arising in practice are of this type. Usually, closed forms for definite sums cannot be obtained by solving the telescoping equation as in the previous example.

Zeilberger [23,25,16] has extended Gosper's algorithm for hypergeometric indefinite summation such as to find recurrence relations for definite sums. His extension is known as the method of creative telescoping, and it has been applied also by Chyzak [4] and Schneider [18,19] for non-hypergeometric summation problems. The idea is as follows. Given a definite sum $F(n) = \sum_{k=1}^{n} f(n,k)$, find $c_0(n),\ldots,c_s(n)$, independent of k, such that $c_0(n)f(n,k) + \cdots + c_s(n)f(n+s,k)$ is summable with respect to k, i.e., such that there exists $u(n,k)$ with

$$c_0(n)f(n,k) + \cdots + c_s(n)f(n+s,k) = u(n,k+1) - u(n,k),$$

for then upon summing this equation over $k = 1,\ldots,n+s$ gives

$$\begin{aligned} c_0(n)F(n) + \cdots + c_s(n)F(n+s) &= u(n,n+s+1) - u(n,1) \\ &\quad + \sum_{i=0}^{s-1} c_i(n) \sum_{j=i+1}^{s} f(n+i,n+j). \end{aligned}$$

We thus obtained a recurrence equation for the definite sum $F(n)$. (Note that s is an explicit number at runtime, so that the double sum on the right hand side is not a symbolic sum but just a polynomial in the $c_i(n)$ and $f(n+i,n+j)$.)

Our procedure for solving linear difference equations can be adapted such as to solve the creative telescoping problem as well. In general, let admissible sequences $a_0(k),\ldots,a_r(k)$ and $g_0(k),\ldots,g_s(k)$ be given. We seek a sequence $u(k)$ and constants $c_0,\ldots,c_s$ such that

$$a_0(k)u(k) + \cdots + a_r(k)u(k+r) = c_0 g_0(k) + \cdots + c_s g_s(k). \tag{6}$$

(For creative telescoping, set $g_i(k) = f(n+i,k)$ and regard n as a constant.) As usual, we assume that admissible sequences $f_1(k),\ldots,f_m(k)$ are specified upon which $u(k)$ is expected to depend polynomially.

Let R be a polynomial ring containing variables $x_{i,j}, y_i, z_i$ for each of the sequences $f_i(k+j), a_i(k), g_i(k)$ $(j = 0,\ldots,r)$, and let $\phi\colon R \to \mathcal{S}$ be the ring homomorphism that maps each variable to the corresponding sequence. Then, like before,

$$(p(f_1(k),\dots,f_m(k)),c_0,\dots,c_s) \text{ solves } (6)$$
$$\iff$$
$$\bigl(p(x_{1,0},\dots,x_{m,0}),\dots\dots,p(x_{1,r},\dots,x_{m,r}),-c_0,\dots,-c_s\bigr)$$
$$\in \operatorname{Syz}(y_0,\dots,y_r,z_0,\dots,z_s).$$

Procedure 17 thus extends as follows.

Procedure 21 ***Input:*** *Admissible sequences* $a_0,\dots,a_r$, $g_0,\dots,g_s$, *and* $f_1,\dots,f_m$ *and a basis of* $\ker\phi$ *with* $\phi\colon R\to\mathcal{S}$ *as defined above.*
Output: *A basis of the vector space of all solutions* $(u,c_0,\dots,c_s)\in\mathcal{S}\times k^{s+1}$ *of*

$$a_0(k)u(k)+\cdots+a_r(k)u(k+r)=c_0g_0(k)+\cdots+c_sg_s(k)$$

in which u *depends polynomially on* $f_1,\dots,f_m$.

1 $B=\emptyset; d=0$
2 *Let* G *be a Gröbner basis of* $\ker\phi$
3 *Let* S *be a Gröbner basis of* $\operatorname{Syz}(y_0,\dots,y_r,z_0,\dots,z_s)\subseteq(R/\ker\phi)^{r+s+2}$
4 **repeat**
5 *Let* P *be a vector space basis of* $\{\,p\in k[x_1,\dots,x_m]:\deg p\le d\,\}$
6 *Delete from* P *all elements* p *with* $\mathrm{LT}(g)\mid\mathrm{LT}(p)$ *for some* $g\in G$
7 $P:=\{\,(p(x_{1,0},\dots,x_{m,0}),\dots,p(x_{1,r},\dots,x_{m,r}),0,\dots,0):p\in P\,\}$
8 $P:=P\cup\{e_i:i=r+1,\dots,r+s+2\}$,
where $e_i=(0,\dots,0,1,0,\dots,0)$ *with* 1 *at position* i
9 *Delete from* P *all elements* p *with* $\mathrm{LT}(b)=\mathrm{LT}(p)$ *for some* $b\in B$
10 *Apply Algorithm 16 to* P *and* S, *obtaining* B_0
11 *Output* $(q_0,q_{r+1},\dots,q_{r+s+2})$ *for each* $(q_0,q_1,\dots,q_{r+s+2})\in B_0$
12 $B:=B\cup B_0; d=d+1$

The technical difference to Procedure 17 is that the accumulated basis elements are now stored as a set of vectors $B\subseteq(R/\ker\phi)^{s+r+2}$ rather than as a set of polynomials $B\subseteq R/\ker\phi$. Consequently, in line 9 we understand by $\mathrm{LT}(p)$ the leading term in the sense of a module term order. This modification is necessary to keep the output linearly independent. Besides this change, the correctness argument of Theorem 18 applies also here, and need not be repeated. Also the remarks of Section 6.1 about the usage of a subideal $\mathfrak{a}\subsetneq\ker\phi$ instead of $\ker\phi$ apply for Procedure 21 as well.

Example 22 *Let us illustrate how to compute a closed form for the definite sum*

$$f(n)=\sum_{k=0}^{n}\frac{1}{F_{2^{n+k}}}.$$

First of all, recall that this sum is not indefinitely summable (Ex. 12). We apply creative telescoping: Find c_0, c_1*, independent of* k*, and* $u(k)$ *such that*

$$u(k+1) - u(k) = \frac{c_0}{F_{2^{n+k}}} + \frac{c_1}{F_{2^{(n+1)+k}}}.$$

Procedure 21 finds the solution $u(k) = 1/F_{2^{n+k}}$*,* $c_0 = -1$*,* $c_1 = 1$*, which we could have spotted right away by inspection:*

$$\frac{1}{F_{2^{n+(k+1)}}} - \frac{1}{F_{2^{n+k}}} = -\frac{1}{F_{2^{n+k}}} + \frac{1}{F_{2^{(n+1)+1}}}.$$

Now summing over $k = 0, \ldots, n$ *gives*

$$\frac{1}{F_{2^{2n+2}}} - \frac{1}{F_{2^n}} = -f(n) - \frac{1}{F_{2^{2n+1}}} + f(n+1), \text{ i.e.,}$$
$$f(n+1) - f(n) = \frac{1}{F_{2^{2n+2}}} + \frac{1}{F_{2^{2n+1}}} - \frac{1}{F_{2^n}}.$$

This recurrence is now suitable as a definition for $f(n)$ *in an admissible system. Procedure 10 can thus be applied to express* $f(n)$ *as a rational function of* $F_{2^n}, F_{2^n+1}, F_{2^{2n}}, F_{2^{2n}+1}$*, say. It finds*

$$f(n) = \sum_{k=0}^{n} \frac{1}{F_{2^{n+k}}} = 1 + \frac{1 - F_{2^n} + F_{2^n+1}}{F_{2^n}} - \frac{F_{2^{2n}+1}}{F_{2^{2n}}}.$$

Unfortunately, owing to the considerable complexity requirements of our procedures, we were not able to compute more sophisticated definite sums that could not already be handled by other algorithms.

7 Conclusion

Difference equations of quite complicated form can be solved algorithmically. In this paper, the focus was on quite a large class of univariate sequences that we called admissible. We have given an effective method for enumerating a basis of the ideal of all algebraic dependencies of a set of given admissible sequences, a problem for which a finite algorithm is not likely to be found. Applications related to combinatorial sequences and symbolic summation were indicated.

References

[1] Sergei A. Abramov. Rational solutions of linear difference and q-difference equations with polynomial coefficients. In *Proceedings of ISSAC '95*, July 1995.

[2] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions*. Dover Publications, Inc., 9 edition, 1972.

[3] Thomas Becker, Volker Weispfenning, and Heinz Kredel. *Gröbner Bases*. Springer, 1993.

[4] Frédéric Chyzak. An extension of Zeilberger's fast algorithm to general holonomic functions. *Discrete Mathematics*, 217:115–134, 2000.

[5] Graham Everest, Alf van der Poorten, Igor Shparlinski, and Thomas Ward. *Recurrence Sequences*, volume 104 of *Mathematical Surveys and Monographs*. American Mathematical Society, 2003.

[6] David Gale. The strange and surprising saga of the Somos sequences. *Mathematical Intelligencer*, 13(1):40–42, 1991.

[7] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics*. Addison-Wesley, second edition, 1994.

[8] P.A. Hendriks and M.F. Singer. Solving difference equations in finite terms. *Journal of Symbolic Computation*, 27(3):239–259, 1999.

[9] Michael Karr. Summation in finite terms. *Journal of the ACM*, 28:305–350, 1981.

[10] Manuel Kauers. An algorithm for deciding zero equivalence of nested polynomially recurrent sequences. Technical Report 2003-48, SFB F013, Johannes Kepler Universität, 2003. (submitted).

[11] Manuel Kauers. *Algorithms for Nonlinear Higher Order Difference Equations*. PhD thesis, RISC-Linz, Johannes Kepler Universität Linz, 2005.

[12] Manuel Kauers. SumCracker — a package for manipulating symbolic sums and related objects. Technical report, SFB F013, Johannes Kepler Universität, 2005. (in preparation).

[13] Manuel Kauers and Burkhard Zimmermann. Computing the algebraic dependencies of recurrent sequences. Technical report, SFB F013, Johannes Kepler Universität, 2005. (in preparation).

[14] L.M. Milne-Thomson. *The Calculus of Finite Differences*. Macmillan and Co., ltd., 1933.

[15] Marko Petkovšek. Hypergeometric solutions of linear recurrences with polynomial coefficients. *Journal of Symbolic Computation*, 14(2–3):243–264, 1992.

[16] Marko Petkovšek, Herbert Wilf, and Doron Zeilberger. $A = B$. AK Peters, Ltd., 1997.

[17] Stanley Rabinowitz. Problem B-951. *The Fibonacci Quarterly*, 40(1):84, 2003.

[18] Carsten Schneider. *Symbolic Summation in Difference Fields*. PhD thesis, RISC-Linz, Johannes Kepler Universität Linz, 2001.

[19] Carsten Schneider. Solving parameterized linear difference equations in terms of indefinite nested sums and products. *Journal of Difference Equations and Applications*, 11(9):799–821, 2005.

[20] Michael Somos. Problem 1470. *Crux Mathematicorum*, 15:208, 1989.

[21] Alf van der Poorten. Elliptic curves and continued fractions. *Journal of Integer Sequences*, 8(2):1–19, 2005.

[22] Mark van Hoeij. Rational solutions of linear difference equations. In *Proceedings of ISSAC'98*, pages 120–123, 1998.

[23] Doron Zeilberger. A fast algorithm for proving terminating hypergeometric identities. *Discrete Mathematics*, 80:207–211, 1990.

[24] Doron Zeilberger. A holonomic systems approach to special functions. *Journal of Computational and Applied Mathematics*, 32:321–368, 1990.

[25] Doron Zeilberger. The method of creative telescoping. *Journal of Symbolic Computation*, 11:195–204, 1991.