Shift Equivalence of P-finite Sequences

Manuel Kauers*

Research Institute for Symbolic Computation
Johannes Kepler University
Altenbergerstraße 69
A4040 Linz, Austria
mkauers@risc.uni-linz.ac.at

Submitted: Aug 8, 2006; Accepted: Oct 19, 2006; Published: Nov 6, 2006
Mathematics Subject Classification: 68W30

Abstract

We present an algorithm which decides the shift equivalence problem for P-finite sequences. A sequence is called P-finite if it satisfies a homogeneous linear recurrence equation with polynomial coefficients. Two sequences are called shift equivalent if shifting one of the sequences s times makes it identical to the other, for some integer s. Our algorithm computes, for any two P-finite sequences, given via recurrence equation and initial values, all integers s such that shifting the first sequence s times yields the second.

1 Introduction

This paper is part of a long-term project concerning the development of a symbolic summation algorithm for finding closed forms of sums

$$\sum_{k=1}^{n} \operatorname{rat}(n, f_1(n), \dots, f_r(n)),$$

where $f_1(n), \dots, f_r(n)$ satisfy homogeneous linear recurrence equations with polynomial coefficients and rat is a multivariate rational function. The principal question is to decide whether there exists another rational function rat_1 such that the above sum is equal to $\operatorname{rat}_1(n, f_1(n), \dots, f_r(n))$ for $n \geq 1$, and if so, to compute one.

Already the case where the $f_i(n)$ satisfy linear recurrence equations with constant coefficients is unsolved. In a recent paper, Greene and Wilf [13] have provided a partial

*Partially supported by FWF grants SFB F1305 and P16613-N12

result by restricting the $f_i(n)$ to such sequences and assuming in addition that the summand involves these sequences only polynomially. For this situation, they have obtained a complete summation algorithm.

The solution to the shift equivalence problem is a step towards allowing nontrivial denominators in the summand expression. The problem is, for two given sequences to decide whether one of them can be matched to the other by shifting it an appropriate number of times. Formally, given $f, g\colon \mathbb{N} \to k$, we want to determine all $s \in \mathbb{Z}$ such that, for all possible n, $f(n) = g(n+s)$.

Several summation algorithms include a subroutine for deciding this problem for some classes of sequences. Gosper's algorithm [12, 21] for indefinite hypergeometric summation requires solving the shift equivalence problem for univariate polynomials, i.e., given $p, q \in \mathbb{Q}[n]$, to determine $s \in \mathbb{Z}$ with $p(n) = q(n+s)$. Also the computation of a greatest factorial factorisation (GFF) requires solving shift equivalence problems [21, 9, 10]. The problem can be solved for polynomials by observing that all possible solutions s must be among the integer roots of the polynomial $\operatorname{res}_n(p(n), q(n+s)) \in \mathbb{Q}[s]$, so in order to solve the problem it suffices to check all those roots. Alternative algorithms are available, we refer to [2, 19, 22] for further information about this case.

Karr's algorithm [14, 15] for simplifying nested sum and product expressions also includes an algorithm for deciding shift equivalence. In Karr's algorithm, sequences are represented as elements of certain types of difference fields (k, E) [7]. The shift equivalence algorithm is, roughly stated, based on finding the orbits in the multiplicative group $\{\, E(f)/f : f \in k \setminus \{0\} \,\}$. See [3, 24] for details.

In the present paper, we present a solution to the shift equivalence problem for sequences $f, g\colon \mathbb{N} \to k$ which are defined by homogeneous linear recurrence equations with polynomial coefficients (P-finite sequences). This is sufficiently general for solving the shift equivalence problems arising in summation. There, we are given multivariate polynomials p_1, p_2 and a tuple of P-finite sequences $f_1, \dots, f_r$ and we have to solve the shift equivalence problem for $f(n) := p_1(f_1(n), \dots, f_r(n))$ and $g(n) := p_2(f_1(n), \dots, f_r(n))$. As the set of P-finite sequences is closed under addition and multiplication [25], also f and g are P-finite and recurrence equations for them can be obtained algorithmically from p_1, p_2 and recurrence equations for $f_1, \dots, f_r$ [23, 18].

2 P-finite and C-finite Sequences

In all theoretical statements made in this paper, it is assumed that k is an arbitrary field of characteristic 0. For the algorithms, however, it is necessary to choose the field k such that certain problems can be solved in k. These are explained at the end of Section 3.2 below.

Definition 1 *[26] Let $f\colon \mathbb{N} \to k$ be a sequence.*

1. *f is called P-finite if there exist polynomials $a_0, \dots, a_r \in k[n]$ such that*

$$a_0(n)f(n) + a_1(n)f(n+1) + \dots + a_r(n)f(n+r) = 0 \quad (n \in \mathbb{N}).$$

2. f is called C-finite if there exist constants $a_0, \dots, a_r \in k$ such that

$$a_0 f(n) + a_1 f(n+1) + \cdots + a_r f(n+r) = 0 \quad (n \in \mathbb{N}).$$

In this section, we recall some known facts about P-finite and C-finite sequences that will be needed in the sequel.

2.1 Annihilating Operators

Let $k(n)$ be the field of univariate rational functions over k, and let $k(n)[E]$ be the univariate skew polynomial ring over $k(n)$ with the commutation rules $En = (n+1)E$ and $Ec = cE$ for each $c \in k$. This is a special instance of an Ore ring [20]. It acts on the ring $k^{\mathbb{N}}$ of sequences via

$$((a_0 + a_1 E + \cdots + a_r E^r) \cdot f)(n) := a_0(n) f(n) + a_1(n) f(n+1) + \cdots + a_r(n) f(n+r).$$

In view of this action, we will refer to the elements of $k(n)[E]$ as operators. If a sequence $f \colon \mathbb{N} \to k$ is P-finite, then there exists an operator $L \in k(n)[E]$ such that $L \cdot f = 0$. The set of all such operators forms a left ideal of $k(n)[E]$, the annihilating ideal of f. Occasionally we will allow also negative powers of E, naturally interpreting them as backwards shift. For $s < 0$, we understand that the sequence $E^s \cdot f$ is defined only for $n > -s$, but we prefer to suppress this detail in order to keep the notation simple.

Annihilating operators are heavily used in symbolic computation algorithms for special functions. For a thorough account on annihilating operators, we refer to [26, 6] and the references given there.

We write $\deg(L)$ for the degree of $L \in k(n)[E]$ with respect to E, i.e., the maximum index $r \in \mathbb{N}$ such that the coefficient of E^r in L is nonzero. Further we define $\deg(0) := -\infty$. In view of the operator interpretation, we shall use the words “order” and “degree” as synonyms for the degree of skew polynomials.

We need some elementary facts about the ring $k(n)[E]$.

Definition 2 *Let $A, B, D \in k(n)[E]$. If there exist $A', B' \in k(n)[E]$ such that $A = A'D$ and $B = B'D$ then D is called a common right divisor of A and B. If D is a common right divisor of maximum degree, then D is called a greatest common right divisor of A and B, written $D = \operatorname{gcrd}(A, B)$.*

The greatest common right divisor of two operators $A, B \in k(n)[E]$ is uniquely determined up to multiplication by elements of the ground field $k(n)$. The monic greatest common right divisor of A and B is called *the* greatest common right divisor (gcrd). The gcrd of any two specific operators can be computed by a modified version of the Euclidean algorithm [4, Sect. 3]. Also, by a modification of the extended Euclidean algorithm, one can compute for any $A, B \in k(n)[E]$ cofactor operators S, T with

$$SA + TB = \operatorname{gcrd}(A, B).$$

As in the commutative case, S and T can be chosen such that $\deg(S) \leq \deg(B)$ and $\deg(T) \leq \deg(A)$. Li [16, 17] has shown that the subresultant theory for efficient computation of gcds can be generalized to gcrds in $k(n)[E]$ as well. This generalizes earlier results of Chardin [5] for differential operators. We need here the following resultant criterion, which is classic for commutative polynomials, and which is contained in Li's work for skew polynomials.

Definition 3 *Let $A, B \in k(n)[E]$, with coefficients*

$$A = a_0(n) + a_1(n)E + \cdots + a_r(n)E^r, \quad B = b_0(n) + b_1(n)E + \cdots + b_s(n)E^s.$$

Then we call

$$\begin{vmatrix}
a_r(n+s-1) & 0 & \cdots & 0 & b_s(n+r-1) & 0 & \cdots & 0 \\
a_{r-1}(n+s-1) & a_r(n+s-2) & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\
\vdots & a_{r-1}(n+s-2) & \ddots & 0 & \vdots & & \ddots & 0 \\
\vdots & & \ddots & a_r(n) & b_1(n+r-1) & & & b_s(n) \\
a_0(n+s-1) & & & a_{r-1}(n) & b_0(n+r-1) & \ddots & & \vdots \\
0 & a_0(n+s-2) & & \vdots & 0 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & b_1(n) \\
0 & \cdots & 0 & a_0(n) & 0 & \cdots & 0 & b_0(n)
\end{vmatrix}$$

the resultant *of A and B (with respect to E), and denote the value of that determinant* $\operatorname{res}(A, B)$.

The resultant of two operators $A, B \in k(n)[E]$ belongs to $k(n)$. Note that no noncommutative arithmetic is required for its computation.

Proposition 1 *[17, Prop. 9.1(1,2)] Let $A, B \in k(n)[E] \setminus k(n)$. Then* $\operatorname{res}(A, B) = 0$ *if and only if* $\deg(\operatorname{gcrd}(A, B)) > 0$.

If $L \in k(n)[E]$ is an annihilating operator of a sequence $f\colon \mathbb{N} \to k$, then so is AL for any $A \in k(n)[E]$. In particular, by choosing an appropriate $A \in k(n)$, we can always replace L by an equivalent operator whose coefficients belong to $k[n]$ instead of $k(n)$. If $L \in k(n)[E]$ is such an operator, i.e.,

$$L = l_0(n) + l_1(n)E + \cdots + l_r(n)E^r$$

with $l_0, \ldots, l_r \in k[n]$, then f is uniquely defined by L and sufficiently many initial values. The number of initial values necessary to define f is given by $\max(0, n_0) + r$, where n_0 is the greatest integer root of l_r. (Set $n_0 := 0$ if l_r does not have any integer roots.) Given this data, many questions about f can be answered algorithmically [23, 18], in particular, it can be decided whether already a right divisor D of L annihilates f.

Proposition 2 *Let $f\colon \mathbb{N} \to k$ be annihilated by $L \in k[n][E]$, and let $A = a_0(n) + a_1(n)E + \cdots + a_r(n)E^r \in k[n][E]$, $B = b_0(n) + b_1(n)E + \cdots + b_s(n)E^s \in k[n][E]$ be such that $L = AB$. Then $B \cdot f = 0$ if and only if $(B \cdot f)(n) = 0$ for $n = 0, \dots, \max(0, n_0) + r$, where n_0 is the greatest integer root of a_r.*

Proof. First of all, we have $(A \cdot g)(n) = 0$ for $n = 0, \dots, \max(0, n_0) + r$ if and only if g is the zero sequence. For $n > \max(0, n_0)$, this can be seen by induction:

$$\begin{aligned} a_r(n)g(n+r) &= a_0(n)g(n) + a_1(n)g(n+1) + \cdots + a_{r-1}(n)g(n+r-1) \\ &= a_0(n)0 + a_1(n)0 + \cdots + a_{r-1}(n)0 = 0, \end{aligned}$$

hence, since $a_r(n) \neq 0$, we must have $g(n+r) = 0$. Now take $g = B \cdot f$. Then $A \cdot (B \cdot f) = (AB) \cdot f = L \cdot f = 0$ implies the claim. $\square$

Note that A can be computed from L and B by right division, if it is not given. Also note that more generally, we can test for any $L' \in k(n)[E]$ whether it annihilates f by applying the proposition to $B := \operatorname{gcrd}(L, L')$.

2.2 Characteristic Polynomial and Companion Matrix

It will be convenient to adopt matrix notation for C-finite operators. If $f\colon \mathbb{N} \to k$ is C-finite, say $L \cdot f = 0$ for some $L = E^r - a_0 - a_1E - \cdots - a_{r-1}E^{r-1} \in k[E]$, then we have the matrix identity

$$\begin{pmatrix} f(n+1) \\ \vdots \\ \vdots \\ f(n+r-1) \\ f(n+r) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ a_0 & a_1 & \cdots & \cdots & a_{r-1} \end{pmatrix} \begin{pmatrix} f(n) \\ \vdots \\ \vdots \\ f(n+r-2) \\ f(n+r-1) \end{pmatrix}$$

for every $n \in \mathbb{N}$. The $r \times r$ matrix in this equation is called the *companion matrix* of L.

Iterating the above equation n times, it follows that

$$\begin{pmatrix} f(n+1) \\ \vdots \\ \vdots \\ f(n+r-1) \\ f(n+r) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ a_0 & a_1 & \cdots & \cdots & a_{r-1} \end{pmatrix}^n \begin{pmatrix} f(0) \\ \vdots \\ \vdots \\ f(r-2) \\ f(r-1) \end{pmatrix},$$

thus any value of f can be obtained by multiplying the vector of initial values by a suitable power of the companion matrix.

The characteristic polynomial of the companion matrix is precisely L. For this reason, L is also called the *characteristic polynomial* of the sequence f. We can always assume that $a_0 \neq 0$ by changing to an operator of lower order, if necessary. In this case, the companion matrix will not have 0 as an eigenvalue.

3 Shift Equivalence of C-finite Sequences

We now introduce an algorithm for solving the shift equivalence problem for two C-finite sequences. The algorithm for the P-finite case calls this algorithm as a subroutine.

Let $f_1, f_2\colon \mathbb{N} \to k$ be C-finite sequences, and suppose that $L_1, L_2 \in k[E]$ are given with $L_1 \cdot f_1 = L_2 \cdot g_2 = 0$. We want to determine all $s \in \mathbb{Z}$ such that $f_1 = E^s \cdot f_2$.

Lemma 1 *Let $f_1, f_2\colon \mathbb{N} \to k$ be annihilated by $L_1, L_2 \in k[E]$, respectively.*

1. *For all $s \in \mathbb{Z}$ and all $L \in k[E]$, we have $L \cdot f_1 = 0$ if and only if $L \cdot (E^s \cdot f_1) = 0$.*
2. *If there exists some $s \in \mathbb{Z}$ with $f_1 = E^s \cdot f_2$, then $L \cdot f_1 = L \cdot f_2 = 0$ for $L := \gcd(L_1, L_2)$.*

Proof.

1. Let $s \in \mathbb{Z}$ and $L = l_0 + l_1 E + \cdots + l_r E^r \in k[E]$. Then

$$\begin{aligned} L \cdot f_1 = 0 &\iff \forall n \in \mathbb{N} : l_0 f_1(n) + l_1 f_1(n+1) + \cdots + l_r f_1(n+r) = 0 \\ &\iff \forall n \in \mathbb{N} : l_0 f_1(n+s) + l_1 f_1(n+s+1) + \cdots + l_r f_1(n+s+r) = 0 \\ &\iff \forall n \in \mathbb{N} : l_0 (E^s f_1)(n) + l_1 (E^s f_1)(n+1) + \cdots + l_r (E^s f_1)(n+r) = 0 \\ &\iff L \cdot (E^s \cdot f_1) = 0. \end{aligned}$$

2. Let $s \in \mathbb{Z}$ be such that $f_1 = E^s f_2$. Then, by part 1, $L_2 \cdot f_1 = 0$. By assumption, $L_1 \cdot f_1 = 0$, hence $(SL_1 + TL_2) \cdot f_1 = 0$ for any $S, T \in k[E]$. As it is possible to choose S, T such that $SL_1 + TL_2 = L$, it follows that $L \cdot f_1 = 0$. For the same reason, $L \cdot f_2 = 0$. $\square$

In order to solve the shift equivalence problem for f_1, f_2, we check in a preprocessing step whether these sequences are annihilated by the same recurrence. Computing $L = \gcd(L_1, L_2)$, we need to check whether $L \cdot f_1 = 0$ and $L \cdot f_2 = 0$, which is possible by Prop. 2. If one or both of the two sequences is not annihilated by L, then there is no solution to the shift equivalence problem, and we return the empty set. Otherwise, we proceed as described in the remainder of this section. From now on, we may assume that $L \in k[E]$ monic with $L \cdot f_1 = L \cdot f_2 = 0$ is given.

3.1 Reduction to a Matrix Equation

Let $r = \deg(L)$ and let $C \in k^{r \times r}$ be the companion matrix of L. Writing

$$F_1(n) := \begin{pmatrix} f_1(n) \\ f_1(n+1) \\ \vdots \\ f_1(n+r-1) \end{pmatrix} \qquad \text{and} \qquad F_2(n) := \begin{pmatrix} f_2(n) \\ f_2(n+1) \\ \vdots \\ f_2(n+r-1) \end{pmatrix},$$

we then have the matrix identities

$$F_1(n) = C^n F_1(0) \quad \text{and} \quad F_2(n) = C^n F_2(0)$$

for all $n \in \mathbb{N}$.

Lemma 2 *In the notation above, we have $f_1 = E^s f_2$ if and only if*

$$F_1(0) = C^s F_2(0), \tag{1}$$

for any $s \in \mathbb{Z}$.

Proof. Let $s \in \mathbb{Z}$. Then

$$\begin{aligned} f_1 = E^s f_2 &\iff \forall n \in \mathbb{N} : F_1(n) = F_2(n+s) \iff \forall n \in \mathbb{N} : C^n F_1(0) = C^{n+s} F_2(0) \\ &\iff F_1(0) = C^s F_2(0), \end{aligned}$$

as claimed. $\square$

Thus in order to solve the shift equivalence problem for f_1, f_2, it remains to solve the matrix equation (1).

3.2 Solution of the Matrix Equation

Let $C \in k^{r\times r}$ be invertible, and $u, v \in k^r$. We seek all $s \in \mathbb{Z}$ satisfying the matrix equation $u = C^s v$. Consider the Jordan decomposition of C, i.e., let $T, J \in \bar{k}^{r\times r}$ be invertible such that $C = T^{-1} J T$ and J is of the form

$$J = \begin{pmatrix} \boxed{J_1} & 0 & \cdots & 0 \\ 0 & \boxed{J_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \boxed{J_m} \end{pmatrix} \quad \text{with} \quad J_i = \begin{pmatrix} \alpha_i & 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & 1 \\ 0 & \cdots & \cdots & 0 & \alpha_i \end{pmatrix} \quad (i = 1, \ldots, m),$$

where each α_i is an eigenvalue of C. Owing to the cancellation of T^{-1} with T, we have $C^s = T^{-1} J^s T$, and so we are done if we find all $s \in \mathbb{Z}$ such that $\bar{u} = J^s \bar{v}$, where $\bar{u} := Tu$ and $\bar{v} := Tv$.

Since

$$J^s = \begin{pmatrix} \boxed{J_1}^s & 0 & \cdots & 0 \\ 0 & \boxed{J_2}^s & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \boxed{J_m}^s \end{pmatrix} \quad (s \in \mathbb{Z}),$$

we can solve the problem for each Jordan block separately. The intersection of the individual solution sets gives the set of all solutions:

Algorithm 1
INPUT: A matrix $C \in k^{r\times r}$, vectors $u = (u_1, \ldots, u_r), v = (v_1, \ldots, v_r) \in k^r$
OUTPUT: All $s \in \mathbb{Z}$ such that $u = C^s v$

function solveMatrixEquation$(C, u, v; k)$
  Compute $J, T \in \bar{k}^{r\times r}$ such that $C = T^{-1}JT$ and J is in Jordan form
  $\bar{u} := Tu; \bar{v} := Tv$
  $S := \mathbb{Z}$
  foreach Jordan block J_i of J **do**
    Let r_0, r_1 be the index of the first and last row of J_i in J, respectively
    $S := S \cap \text{solveMESingleJordanBlock}(J_i, (\bar{u}_{r_0}, \ldots, \bar{u}_{r_1}), (\bar{v}_{r_0}, \ldots, \bar{v}_{r_1}); \bar{k})$ // Alg. 2
  return S

Now assume that $J \in k^{r\times r}$ consists of a single Jordan block, and let $\alpha \neq 0$ be its eigenvalue. We can assume without loss of generality that $\bar{u}_r \neq 0 \neq \bar{v}_r$. (Otherwise: If $\bar{u} = \bar{v} = 0$, the solution set is $\mathbb{Z}$. If $\bar{u}_r = \bar{v}_r = 0$ we can drop the last entries of $\bar{u}, \bar{v}$ and the last row and the last column from J, and iterate if necessary. If $\bar{u}_r = 0$ and $\bar{v}_r \neq 0$ or $\bar{u}_r \neq 0$ and $\bar{v}_r = 0$, then the solution set is $\emptyset$.) As can be shown easily by induction, we have

$$J^s = \begin{pmatrix} \alpha & 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & 1 \\ 0 & \cdots & \cdots & 0 & \alpha \end{pmatrix}^s = \begin{pmatrix} \alpha^s & s\alpha^{s-1} & \binom{s}{2}\alpha^{s-2} & \cdots & \binom{s}{r-1}\alpha^{s-(r-1)} \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \binom{s}{2}\alpha^{s-2} \\ \vdots & & \ddots & \alpha^s & s\alpha^{s-1} \\ 0 & \cdots & \cdots & 0 & \alpha^s \end{pmatrix} \qquad (s \in \mathbb{Z}).$$

If $r = 1$, the solution set for $\bar{u} = J\bar{v}$ is simply given by

$$\{\, s \in \mathbb{Z} : \bar{u}_r/\bar{v}_r = \alpha^s \,\}.$$

If $r > 1$, then the last two rows of the matrix equation yield

$$\begin{aligned} \bar{u}_{r-1} = \alpha^s \bar{v}_{r-1} + s\alpha^{s-1}\bar{v}_r &= \frac{\bar{u}_r}{\bar{v}_r}\bar{v}_{r-1} + \frac{s}{\alpha}\frac{\bar{u}_r}{\bar{v}_r}\bar{v}_r \\ \Longrightarrow \quad s &= \alpha\Big(\frac{\bar{u}_{r-1}}{\bar{u}_r} - \frac{\bar{v}_{r-1}}{\bar{v}_r}\Big) \end{aligned}$$

as a unique solution candidate. If this s is not an integer, or it does not satisfy $\bar{u} = J^s\bar{v}$, then the solution set is $\emptyset$, otherwise it is $\{s\}$. This gives the following algorithm.

Algorithm 2
INPUT: A Jordan block $J \in k^{r\times r}$, and vectors $u = (u_1, \ldots, u_r)$, $v = (v_1, \ldots, v_r) \in k^r$
OUTPUT: All $s \in \mathbb{Z}$ such that $u = J^s v$

function solveMESingleJordanBlock$(J, u, v; k)$
  if $u = v = (0, \ldots, 0)$ **then** **return** $\mathbb{Z}$

3 **while** $v_r = 0$ **do**
4   **if** $u_r = 0$ **then** $r := r - 1$ **else** **return** $\emptyset$
5 // now $v_r \neq 0$
6 **if** $u_r = 0$ **then** **return** $\emptyset$
7 // now $u_r \neq 0 \neq v_r$
8 Let $\alpha \in k$ be the diagonal element of J
9 **if** $r = 1$ **then** **return** $\{s \in \mathbb{Z} : u_r/v_r = \alpha^s\}$
10 $s := \alpha(u_{r-1}/u_r - v_{r-1}/v_r)$
11 **if** $s \in \mathbb{Z}$ **and** $u = J^s v$ **then** **return** $\{s\}$ **else** **return** $\emptyset$

The correctness of Algorithms 1 and 2 should be clear by the above discussion. Several restrictions, however, have to be made for the field k in order that every step in these algorithms can be carried out algorithmically. Of course, it is necessary that k is a computable, i.e., that every element has a finite representation, that the arithmetic operations $+, -, \cdot, /$ are computable, and that zero equivalence can be decided. Furthermore, for the computation of a Jordan decomposition (Line 2 in Alg. 1), we need to be able to compute absolute factorizations of univariate polynomials in $k[X]$. The algebraic closure $\bar{k}$ also has to be a computable field. Line 11 of Algorithm 2 requires to decide whether an element of $\bar{k}$ is an integer. All these requirements can be accommodated for most fields k that might be of interest. More restrictive is the final requirement, originating from line 9: We have to be able to compute the set $\{s \in \mathbb{Z} : a = b^s\}$ for given $a, b \in \bar{k}$. An algorithm for this purpose was given by Abramov and Bronstein [1]. This algorithm is applicable whenever k is such that it can be decided for any given $x \in k$ whether x is transcendental or algebraic over $\mathbb{Q}$, and that for any two elements $x, y \in k$ it can be decided whether these elements are algebraically independent over $\mathbb{Q}$. Ge's algorithm [8] gives rise to an efficient alternative if k is a single algebraic extension of $\mathbb{Q}$, i.e., if $k = \mathbb{Q}(\alpha)$ for some algebraic number α.

3.3 Summary

Lemma 2 reduces the shift equivalence problem for C-finite sequence to solving a matrix equation, and this matrix equation can be solved by means of Algorithm 1. Putting things together, we thus obtain the following algorithm for solving the shift equivalence problem for C-finite sequences.

Algorithm 3
INPUT: $f_1, f_2 \colon \mathbb{N} \to k$ C-finite, specified by annihilating operators $L_1, L_2 \in k[E]$ and initial values
OUTPUT: all $s \in \mathbb{Z}$ such that $f_1 = E^s f_2$

function cfiniteSE(f_1, f_2)
   $L := \gcd(L_1, L_2) \in k[E]$
   if $L \cdot f_1 \neq 0$ **or** $L \cdot f_2 \neq 0$ **then** **return** $\emptyset$

Let $r := \deg(L)$ and $C \in k^{r\times r}$ be the companion matrix of L
return solveMatrixEquation$(C, (f_1(0), \ldots, f_1(r-1)), (f_2(0), \ldots, f_2(r-1)))$

3.4 Examples

Example 1 *Let $f_1, f_2 \colon \mathbb{Z} \to \mathbb{Q}$ be defined by*

$$\begin{aligned} f_1(n+3) &= 5f_1(n+2) - 8f_1(n+1) + 4f_1(n), && f_1(0) = 0, f_1(1) = -16, f_1(2) = -64,\\ f_2(n+3) &= 2f_2(n+2) + 4f_2(n+1) - 8f_2(n), && f_2(0) = \tfrac{1}{4}, f_2(1) = \tfrac{7}{16}, f_2(2) = \tfrac{3}{4}. \end{aligned}$$

In operator notation, we have

$$\underbrace{(E^3 - 5E^2 + 8E - 4)}_{=:L_1} \cdot f_1 = 0, \qquad \underbrace{(E^3 - 2E^2 - 4E + 8)}_{=:L_2} \cdot f_2 = 0.$$

The greatest common divisor of these operators is

$$L := \gcd(L_1, L_2) = E^2 - 4E + 4 = (E-2)^2,$$

and it can be checked that $L \cdot f_1 = L \cdot f_2 = 0$.

Computing the Jordan decomposition of the companion matrix, we find

$$C := \begin{pmatrix} 0 & 1 \\ -4 & 4 \end{pmatrix} = \begin{pmatrix} 0 & 1/2 \\ -2 & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1/2 \\ -2 & 1 \end{pmatrix} =: T^{-1}JT.$$

Applying T to the vectors of initial values leads to

$$\bar{u} = \begin{pmatrix} 0 & 1/2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ -16 \end{pmatrix} = \begin{pmatrix} -8 \\ -16 \end{pmatrix}, \qquad \bar{v} = \begin{pmatrix} 0 & 1/2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1/4 \\ 7/16 \end{pmatrix} = \begin{pmatrix} 7/32 \\ -1/16 \end{pmatrix}.$$

It remains to determine $s \in \mathbb{Z}$ such that

$$\begin{pmatrix} -8 \\ -16 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix}^s \begin{pmatrix} 7/32 \\ -1/16 \end{pmatrix}. \tag{2}$$

Since J consists of a single Jordan block of size two, we have a unique solution candidate:

$$s = 2\Big(\frac{-8}{-16} - \frac{7/32}{-1/16}\Big) = 8$$

Indeed, (2) is fullfilled for $s = 8$, and it follows that $f_1 = E^s f_2$ if and only if $s = 8$.

Example 2 *Consider $f_1, f_2 \colon \mathbb{N} \to \mathbb{Q}$ defined via*

$$\begin{aligned} f_1(n+3) &= -f_1(n+2) + f_1(n+1) + f_1(n), && f_1(0) = 0, f_1(1) = 0, f_1(2) = 4,\\ f_2(n+3) &= -f_2(n+2) + f_2(n+1) + f_2(n), && f_2(0) = 8, f_2(1) = 8, f_2(2) = 4. \end{aligned}$$

We have $L \cdot f_1 = L \cdot f_2 = 0$ *for*

$$L = E^3 + E^2 - E - 1 = (E+1)(E-1)^2 \in k[E].$$

Computing the Jordan decomposition of the companion matrix, we find

$$\begin{aligned} C &:= \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1/4 & -1/2 & 1/4 \\ -1/4 & 1/2 & 3/4 \\ -1/2 & 0 & 1/2 \end{pmatrix}^{-1} \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1/4 & -1/2 & 1/4 \\ -1/4 & 1/2 & 3/4 \\ -1/2 & 0 & 1/2 \end{pmatrix} =: T^{-1}JT. \end{aligned}$$

Applying T *to the vectors of initial values leads to*

$$\bar{u} = T \begin{pmatrix} 0 \\ 0 \\ 4 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix}, \quad \bar{v} = T \begin{pmatrix} 8 \\ 8 \\ 4 \end{pmatrix} = \begin{pmatrix} -1 \\ 5 \\ -2 \end{pmatrix}.$$

It remains to find $s \in \mathbb{Z}$ *such that*

$$\begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}^{s} \begin{pmatrix} -1 \\ 5 \\ -2 \end{pmatrix}.$$

The matrix J *consists of two Jordan blocks which have to be considered separately. The first block has length* 1*, and it restricts the solutions to the set*

$$S_1 := \{\, s \in \mathbb{Z} : \frac{1}{-1} = (-1)^s \,\} = 1 + 2\mathbb{Z}$$

of all odd integers. The second block has length 2*, so it leads to the unique solution candidate*

$$s = 1(\tfrac{3}{2} - (-\tfrac{5}{2})) = 4.$$

Since $S_1 \cap \{4\} = \emptyset$*, it follows that the two sequences* f_1 *and* f_2 *are not shift equivalent.*

4 Shift Equivalence of P-finite Sequences

The algorithm for the P-finite case consists of a case distinction: either the question can be reduced to a shift equivalence problem for C-finite sequences, and then Algorithm 3 above can be applied, or it is possible to determine a finite number of candidate solutions s, which can be checked one ofter the other.

Contrary to the C-finite case, for a general operator $L \in k(n)[E]$, it is no longer the case that $L \cdot f = 0 \iff L \cdot (E^s \cdot f) = 0$. The following definition is made in order to repair this deficiency.

Definition 4 *For $A = a_0(n) + a_1(n)E + \cdots + a_r(n)E^r \in k(n)[E]$ and $s \in k$, we define*

$$A^{(s)} = a_0(n+s) + a_1(n+s)E + \cdots + a_r(n+s)E^r.$$

With this definition, we can formulate the following generalization of Lemma 1.

Lemma 3 *Let $f_1, f_2 \colon \mathbb{N} \to k$ be annihilated by $L_1, L_2 \in k(n)[E]$, respectively.*

1. *For all $s \in \mathbb{Z}$ and all $L \in k(n)[E]$, we have $L \cdot f_1 = 0$ if and only if $L^{(s)} \cdot (E^s \cdot f_1) = 0$.*
2. *If there exists some $s \in \mathbb{Z}$ with $f_1 = E^s \cdot f_2$, then $L \cdot f_1 = L \cdot f_2 = 0$ for $L := \operatorname{gcrd}(L_1, L_2^{(s)})$.*

Proof.

1. Let $s \in \mathbb{Z}$ and $L = l_0(n) + l_1(n)E + \cdots + l_r(n)E^r \in k(n)[E]$. Then

$$\begin{aligned} L \cdot f_1 = 0 &\iff \forall n \in \mathbb{N} : l_0(n)f_1(n) + \cdots + l_r(n)f_1(n+r) = 0 \\ &\iff \forall n \in \mathbb{N} : l_0(n+s)f_1(n+s) + \cdots + l_r(n+s)f_1(n+s+r) = 0 \\ &\iff \forall n \in \mathbb{N} : l_0(n+s)(E^s \cdot f_1)(n) + \cdots + l_r(n+s)(E^s \cdot f_1)(n+r) = 0 \\ &\iff L^{(s)} \cdot (E^s \cdot f_1) = 0. \end{aligned}$$

2. Let $s \in \mathbb{Z}$ such that $f_1 = E^s f_2$. Then, by part 1, $L_2^{(s)} \cdot f_1 = 0$. By assumption, $L_1 \cdot f_1 = 0$, hence $(SL_1 + TL_2^{(s)}) \cdot f_1 = 0$ for any $S, T \in k(n)[E]$. As it is possible to choose $S, T \in k(n)[E]$ such that $SL_1 + TL_2^{(s)} = L$, it follows that $L \cdot f_1 = 0$. For the same reason, $L \cdot f_2 = 0$. $\square$

4.1 The degenerate Case

Let $L_1, L_2 \in k(n)[E]$ be given. We may extend the ground field k by a new transcendental element s, commuting with E, and consider L_1, L_2 as elements of $k(s)(n)[E]$, with coefficients free of s. In this setting we can form $L_2^{(s)}$ for *symbolic* s and consider $L := \operatorname{gcrd}(L_1, L_2^{(s)})$. It turns out that the coefficients of L neither contain s nor n:

Lemma 4 *Let $L_1, L_2 \in k(n)[E]$.*

1. $\deg(\operatorname{gcrd}(L_1, L_2^{(s)})) > 0$ *for infinitely many $s \in \mathbb{Z}$ if and only if* $\deg(\operatorname{gcrd}(L_1, L_2^{(s)})) > 0$ *where $L_1, L_2^{(s)}$ are viewed as elements of $k(s)(n)[E]$.*
2. *If $L_1, L_2^{(s)}$ are viewed as elements of $k(s)(n)[E]$, then $L := \operatorname{gcrd}(L_1, L_2^{(s)})$ belongs to $k[E]$.*

Proof.

1. Consider the resultant $\operatorname{res}(L_1, L_2^{(s)}) \in k(s)(n)$. By Prop. 1, a nontrivial gcrd appears precisely for those values of s where the resultant vanishes. Since the resultant is a rational function in s over a field of characteristic zero, it can only have infinitely many integer roots if it is identically zero. Then, however, already the gcrd over $k(s)(n)$ must be nontrivial, again by Prop. 1.

2. Since L is a right divisor of L_1 and L_1 does not involve s, also L is free of s. Furthermore, we have that $L^{(-s)} = \operatorname{gcrd}(L_1^{(-s)}, (L_2^{(s)})^{(-s)}) = \operatorname{gcrd}(L_1^{(-s)}, L_2)$ is a right divisor of $L_2 \in k(n)[E]$ and therefore it is free of s, too. But L and $L^{(-s)}$ can be simulaniously free of s only if they are also free of n. $\square$

The degenerate case happens if $L := \operatorname{gcrd}(L_1, L_2^{(s)})$ (computed in $k(s)(n)[E]$) is already an annihilator for both f_1, f_2. In this case, the sequences f_1, f_2 are C-finite and we can proceed with Algorithm 3.

4.2 The nondegenerate Case

The nondegenerate case happens if $L := \operatorname{gcrd}(L_1, L_2^{(s)})$ (computed in $k(s)(n)[E]$) is not an annihilator of f_1, f_2. In this case, in view of Lemma 3, part 2, it is necessary for every solution $s \in \mathbb{Z}$ of the shift equivalence problem that $\operatorname{gcrd}(L_1/L, L_2^{(s)}/L)$ is nontrivial. By Prop. 1, this happens precisely for the integer roots of

$$\operatorname{res}(\operatorname{rquo}(L_1, L), \operatorname{rquo}(L_2^{(s)}, L)) \in k(s, n),$$

where $\operatorname{rquo}(A, B)$ denotes the right quotient of $A \in k(s)(n)[E]$ by $B \in k(s)(n)$. By Lemma 4, it follows that the resultant is not identically zero, for otherwise L would not be the greatest common right divisor of L_1 and $L_2^{(s)}$. Thus the resultant can only have finitely many roots in the integers, and the shift equivalence problem can be solved by trying each of them.

Alternatively, the values s for which $\operatorname{rquo}(L_1, L)$ and $\operatorname{rquo}(L_2^{(s)}, L)$ have a nontrivial greatest common right divisor could also be obtained by an efficient algorithm due to Glotov [11].

4.3 Summary

Putting things together, we obtain the following algorithm for solving the shift equivalence problem for P-finite sequences.

Algorithm 4
INPUT: $f_1, f_2 \colon \mathbb{N} \to k$, specified by annihilating operators $L_1, L_2 \in k(n)[E]$ and sufficiently many initial values.
OUTPUT: all $s \in \mathbb{Z}$ such that $f_1 = E^s f_2$

function pfiniteSE(f_1, f_2)
  $L := \operatorname{gcrd}(L_1, L_2^{(s)})$ // computed in $k(s)(n)[E]$
  if $L \cdot f_1 = 0$ **and** $L \cdot f_2 = 0$ **then**
    return cfiniteSE(f_1, f_2) // specifying L as ann. operator of both f_1 and f_2
  $R(s) := \operatorname{res}(\operatorname{rquo}(L_1, L), \operatorname{rquo}(L_2^{(s)}, L)) \in k(s)(n)$
  $C := \{s \in \mathbb{Z} : R(s) = 0\}; S := \emptyset$
  forall $s \in C$ **do**
    if $f_1 = E^s f_2$ **then** $S := S \cup \{s\}$
  return S

4.4 Examples

Example 3 *Let* $f_1, f_2 \colon \mathbb{N} \to \mathbb{Q}$ *be defined via*

$$\begin{aligned} L_1 \cdot f_1 &= 0, \qquad f_1(0) = 0, f_1(1) = -16, f_1(2) = -64, \\ L_2 \cdot f_2 &= 0, \qquad f_2(0) = \tfrac{1}{4}, f_2(1) = \tfrac{7}{16}, f_2(2) = \tfrac{3}{4}, \end{aligned}$$

where

$$\begin{aligned} L_1 &:= (n+1)E^3 - (5n+4)E^2 + 4(2n+1)E - 4n, \\ L_2 &:= nE^3 - (5n+1)E^2 + 4(2n+1)E - 4(n+1). \end{aligned}$$

Computing $L := \operatorname{gcrd}(L_1, L_2^{(s)})$ *in* $\mathbb{Q}(s)(n)[E]$*, we obtain*

$$L = E^2 - 4E + 4,$$

and since $L \cdot f_1 = L \cdot f_2 = 0$*, we may proceed as in Example 1, obtaining that* $f_1 = E^s f_2$ *if and only if* $s = 8$*.*

Example 4 *Let* $f_1, f_2 \colon \mathbb{N} \to \mathbb{Q}$ *be defined via*

$$\begin{aligned} L_1 \cdot f_1 &= 0, \qquad f_1(0) = 5, f_1(1) = \tfrac{125}{8}, f_1(2) = \tfrac{209}{4}, \\ L_2 \cdot f_2 &= 0, \qquad f_2(0) = 5, f_2(1) = \tfrac{5}{2}, f_2(2) = 5. \end{aligned}$$

where

$$\begin{aligned} L_1 &:= (n+6)(n+1)E^3 - (6n^2+33n+7)E^2 + (9n^2+30n-49)E - (2n-3)(n+4), \\ L_2 &:= (n+4)^2E^3 - 2(3n^2+18n+28)E^2 + 3(3n^2+9n+4)E - 2n(n+2). \end{aligned}$$

We have $\operatorname{gcrd}(L_1, L_2^{(s)}) = 1$ *when computing in* $\mathbb{Q}(s)(n)[E]$*, and* 1 *obviously does not annihilate* f_1 *or* f_2*, so we are in the nondegenerate case. The resultant reads*

$$\begin{aligned} \operatorname{res}(L_1, L_2^{(s)}) = -3(s-2)^2(&27n^7 + 18sn^6 + 549n^6 - 108s^2n^5 - 72sn^5 + 3276n^5 \\ &- 162s^3n^4 - 2304s^2n^4 - 3714sn^4 - 1722n^4 - 63s^4n^3 - 2196s^3n^3 \\ &- 15753s^2n^3 - 29847sn^3 - 50634n^3 - 513s^4n^2 - 8976s^3n^2 - 32808s^2n^2 \\ &- 34370sn^2 - 26246n^2 - 213s^4n + 699s^3n + 53200s^2n + 227440sn \\ &+ 353172n + 3222s^4 + 60336s^3 + 237486s^2 + 205572s - 95040). \end{aligned}$$

The last factor is irreducible, so the resultant has the only integer root $s = 2$. Comparing initial values confirms that $f_1 = E^s f_2$ if and only if $s = 2$.

Acknowledgement. I would like to thank Carsten Schneider for helpful discussions.

References

[1] Sergei A. Abramov and Manuel Bronstein. Hypergeometric dispersion and the orbit problem. In *Proceedings of ISSAC'00*, pages 8–13, 2000.

[2] Sergej A. Abramov. On the summation of rational functions. *Zh. vychisl. mat. Fiz*, pages 1071–1075, 1971.

[3] Manuel Bronstein. On solutions of linear ordinary difference equations in their coefficient field. *Journal of Symbolic Computation*, 29:841–877, 2000.

[4] Manuel Bronstein and Marko Petkovšek. An introduction to pseudo-linear algebra. *Theoretical Computer Science*, 157(1):3–33, 1996.

[5] Marc Chardin. Differential resultants and subresultants. In *Proceedings of FCT'91*, volume 529 of *Lecture Notes in Computer Science*, pages 1–10, 1991.

[6] Frédéric Chyzak and Bruno Salvy. Non-commutative elimination in Ore algebras proves multivariate identities. *Journal of Symbolic Computation*, 26:187–227, 1998.

[7] Richard M. Cohn. *Difference Algebra*. Interscience Publishers, John Wiley & Sons, 1965.

[8] Guoqiang Ge. *Algorithms related to multiplicative representations of algebraic numbers*. PhD thesis, U.C. Berkeley, 1993.

[9] Jürgen Gerhard. Modular algorithms for polynomial basis conversion and greatest factorial factorization. In *Proceedings of the 7th Rhine Workshop of Computer Algebra*, pages 125–141, 1999.

[10] Jürgen Gerhard. *Modular Algorithms in Symbolic Summation and Symbolic Integration*, volume 3218 of *LNCS*. Springer, 2004.

[11] Peter E. Glotov. An algorithm of searching the greatest common divisor for ore polynomial with polynomial coefficients depending on a parameter. *Programming and Computer Software*, 24(6):275–283, 1998.

[12] William Gosper. Decision procedure for indefinite hypergeometric summation. *Proceedings of the National Academy of Sciences of the United States of America*, 75:40–42, 1978.

[13] Curtis Greene and Herbert S. Wilf. Closed form summation of C-finite sequences. *Transactions of the American Mathematical Society*, 2006. to appear.

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

[15] Michael Karr. Theory of summation in finite terms. *Journal of Symbolic Computation*, 1(3):303–315, 1985.

[16] Ziming Li. *A Subresultant Theory for Linear Differential, Linear Difference, and Ore Polynomials, with Applications.* PhD thesis, RISC-Linz, 1996.

[17] Ziming Li. A subresultant theory for Ore polynomials with applications. In *Proceedings of ISSAC'98*, pages 132–139, 1998.

[18] Christian Mallinger. Algorithmic manipulations and transformations of univariate holonomic functions and sequences. Master's thesis, J. Kepler University, Linz, August 1996.

[19] Yiu-Kwong Man and Francis J. Wright. Fast polynomial dispersion computation and its application to indefinite summation. In *Proceedings of ISSAC'94*, pages 175–180, 1994.

[20] O. Ore. Theory of non-commutative polynomials. *Annals of Mathematics*, 34:480–508, 1933.

[21] Peter Paule. Greatest factorial factorization and symbolic summation. *Journal of Symbolic Computation*, 20:235–268, 1995.

[22] Peter Paule and Volker Strehl. Symbolic summation – some recent developments. In *Computer Algebra in Science and Engeneering – Algorithms, Systems, and Applications*, pages 138–162, 1995.

[23] Bruno Salvy and Paul Zimmermann. Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable. *ACM Transactions on Mathematical Software*, 20(2):163–177, 1994.

[24] Carsten Schneider. A collection of denominator bounds to solve parameterized linear difference equations in $\Pi\Sigma$-extensions. In *Proceedings of SYNASC'04*, pages 269–282, 2004.

[25] Richard P. Stanley. *Enumerative Combinatorics, Volume 2.* Cambridge Studies in Advanced Mathematics 62. Cambridge University Press, 1999.

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