Indefinite Summation with Unspecified Summands

Manuel Kauers [1] and Carsten Schneider [2]

Research Institute for Symbolic Computation
Johannes Kepler Universität
A-4040 Linz, Austria

Abstract

We provide a new algorithm for indefinite nested summation which is applicable to summands involving unspecified sequences $x(n)$. More than that, we show how to extend Karr's algorithm to a general summation framework by which additional types of summand expressions can be handled. Our treatment of unspecified sequences can be seen as a first illustrative application of this approach.

Key words: Combinatorial Identities, Symbolic Summation, Difference Fields

1 Introduction

In order to find a "closed form" for the sum $F(a,b) = \sum_{k=a}^{b} f(k)$, with $f(k)$ independent of a and b, we focus on the summand sequence $f(k)$. If there exists a solution $g(k)$ of the *telescoping equation*

$$g(k+1) - g(k) = f(k),$$

then we obtain the result $F(a,b) = g(b+1) - g(a)$. Solving the telescoping equation is therefore referred to as *indefinite summation,* and for various classes of sequences $f(k)$, there are algorithms available for doing this job. For instance, Gosper's algorithm (7; 13) and its q-generalization (14) can find

[1] Partially supported by the Austrian Science Foundation (FWF) grant F1305 and the German Academic Exchange Service (DAAD) grant D/03/40515
[2] Supported by the SFB-grant F1305 and by grant P16613-N12 of the Austrian FWF

classical (q-)hypergeometric identities like

$$\sum_{k=0}^{n} kk! = (n+1)! - 1 \qquad \text{or} \qquad \sum_{k=0}^{n} [k]_q [k]_q! = \frac{1}{q}([n+1]_q! - 1), \tag{1}$$

respectively, where $[k]_q := (1-q^k)/(1-q)$ and $[k]_q! = [1]_q[2]_q \cdots [k]_q$.

In this paper, we provide an algorithm that does indefinite summation where the summand $f(k)$ may depend on an *unspecified* sequence $x(k)$. This algorithm is able to compute identities like, e.g.,

$$\sum_{k=0}^{n} \Big(x(k+1) - 1\Big) \prod_{i=1}^{k} x(i) = \prod_{k=1}^{n+1} x(k) - 1, \tag{2}$$

that hold for all sequences $x(k)$. Observe that both classical identities mentioned above are included here for appropriate choices of $x(k)$. An equivalent form of (2), also in the scope of our algorithm, appears in Apery's proof of the irrationality of $\zeta(3)$ (27):

$$\sum_{k=1}^{n} \frac{1}{x(k)} \prod_{i=1}^{k} \frac{x(i)}{a+x(i)} = \frac{1}{a}\Big(1 - \prod_{k=1}^{n+1} \frac{x(k)}{a+x(k)}\Big)$$

($a \neq 0, x(k) \neq 0, x(k) \neq -a$ $(k \in \mathbb{N})$). A similar identity, given in (8, Exercise 5.93), is covered by (2) as well. Additional sums, like

$$\sum_{k=0}^{n} (-1)^k \Big(x(k) + x(a-k)\Big) \prod_{i=1}^{k} \tfrac{x(a-i+1)}{x(i)} = x(0) + x(a-n)(-1)^n \prod_{i=1}^{n} \tfrac{x(a-i+1)}{x(i)},$$

which specializes, e.g., with $x(i) = i$ and $a = n$ to

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

supplement this general type of identities.

While algorithms for *checking* identities of this type are already known (10; 11), our algorithm appears to be the first which can also *find* identities for this general class of summands.

Our approach extends the abilities of the summation package `Sigma` (20). This package is based on Karr's difference field theory (9) and allows not only to deal with (q-)hypergeometric terms, but also with rational expressions involving indefinite nested sums and products. For instance, `Sigma` is able to find identities like

$$\sum_{k=0}^{n} \sum_{i=0}^{k} \binom{a}{i} = \tfrac{1}{2}\Big((a-n)\binom{a}{n} + (2n-a+2) \sum_{k=0}^{n} \binom{a}{k}\Big), \quad \text{or} \tag{3}$$

$$\sum_{k=1}^{n} k^2 H_{n+k} = \tfrac{1}{3} n (n+\tfrac{1}{2})(n+1)(2H_{2n} - H_n) - \tfrac{1}{36}(10n^2 + 9n - 1), \tag{4}$$

appearing in (2, there for $n = a$) and (8, Exercise 6.69), respectively. (We write $H_n = \sum_{k=1}^{n} 1/k$ for the nth harmonic number.) Also these identities can be generalized by our new algorithm. The extended version of `Sigma` produces

$$\sum_{k=0}^{n}\sum_{i=0}^{k} x(i) = (n+1)\sum_{k=0}^{n} x(k) - \sum_{k=0}^{n} kx(k) \quad \text{and} \tag{5}$$

$$\sum_{k=0}^{n} k^2 \sum_{i=0}^{k} x(i) = \tfrac{1}{6}\Big(n(n+1)(2n+1)\sum_{k=0}^{n} x(k) - \sum_{k=0}^{n} kx(k) + 3\sum_{k=0}^{n} k^2 x(k) - 2\sum_{k=0}^{n} k^3 x(k)\Big). \tag{6}$$

Similarly, we obtain various identities in (12), like

$$\sum_{k=1}^{n}(-1)^k \sum_{j=1}^{k} x(j) = \frac{1}{2}\Big[(-1)^n \sum_{k=1}^{n} x(k) + \sum_{k=1}^{n}(-1)^k x(k)\Big] \tag{7}$$

which we can specialize to known identities, e.g, with $x(j) := \frac{1}{j}$ to

$$\sum_{k=1}^{n}(-1)^k H_k = \frac{1}{2}(-1)^n H_n + \frac{1}{2}\sum_{k=1}^{n}\frac{(-1)^k}{k},$$

or with $x(j) = \binom{m}{j-1}, n := m+1$ to (28, Thm. 4.2):

$$\sum_{k=0}^{m}(-1)^{k+1}\sum_{j=0}^{k}\binom{m}{j} = \frac{1}{2}(-1)^{m+1}2^m.$$

Simplifications (26; 19; 22) and generalizations (21; 23; 25) of Karr's summation algorithm (9) are the backbone of `Sigma`'s indefinite summation toolbox. These algorithms proceed by representing the sums under consideration as elements of suitable *difference fields,* called $\Pi\Sigma$-fields. Section 2 gives a short introduction into solving the telescoping equation in such fields. In this article, we show that these algorithms can as well be applied to certain difference fields that are not $\Pi\Sigma$-fields. In fact, in Section 3, we will give a precise list of all requirements that a difference field has to meet in order to be "compatible" to the $\Pi\Sigma$-algorithms. We obtain an algorithm for indefinite summation of nested sums and products over any expressions that can be represented as elements of a difference field that meets these requirements. As an example application, we provide such a difference field for representing unspecified sequences $x(k)$ in Section 5, adapting the idea of (10).

2 Telescoping Problems in Difference Fields

The overall strategy of our approach is as follows. In a first step we reformulate the telescoping problem by representing the summand $f(k)$ as element of a field $\mathbb{F}$ where the action of the shift operator $S_k\, f(k) := f(k+1)$ is reflected by a field automorphism $\sigma\colon \mathbb{F} \to \mathbb{F}$. This leads to the concept of *difference fields.* A difference field is a pair $(\mathbb{F}, \sigma)$ where $\mathbb{F}$ is a field and σ is an $\mathbb{F}$-automorphism. [3] The constant field of $(\mathbb{F}, \sigma)$ is defined as $\text{const}_\sigma\mathbb{F} = \{\, c \in \mathbb{F} \mid \sigma(c) = c\,\}$.

Then the second step consists of solving the telescoping equation in the difference field $(\mathbb{F}, \sigma)$: Given $f \in \mathbb{F}$, find, if possible, a $g \in \mathbb{F}$ such that

$$\sigma(g) - g = f. \tag{8}$$

Example 1 For identity

$$\sum_{k=0}^{n} H_k = (n+1)H_n - n$$

we proceed as follows. *1. Construction of the difference field* $(\mathbb{F}, \sigma)$*:* We start with the difference field $(\mathbb{Q}, \sigma)$ with $\sigma(c) = c$ for all $c \in \mathbb{Q}$, i.e., $\text{const}_\sigma\mathbb{Q} = \mathbb{Q}$. Next, we construct the transcendental field extension $\mathbb{Q}(k)$ and extend the automorphism $\sigma\colon \mathbb{Q} \to \mathbb{Q}$ to $\mathbb{Q}(k)$ by defining the shift relation $\sigma(k) = k + 1$. Finally, we extend this difference field $(\mathbb{Q}(k), \sigma)$ by taking the transcendental field extension $\mathbb{F} := \mathbb{Q}(k)(t)$ and extending $\sigma\colon \mathbb{Q}(k) \to \mathbb{Q}(k)$ to $\mathbb{Q}(k)(t)$ by the shift relation $\sigma(t) = t + \frac{1}{k+1}$. This means that our difference field $(\mathbb{F}, \sigma)$ consists of the rational function field $\mathbb{Q}(k)(t)$ and the field automorphism $\sigma\colon \mathbb{Q}(k)(t) \to \mathbb{Q}(k)(t)$ with $\sigma(k) = k + 1$ and $\sigma(t) = t + \frac{1}{k+1}$. Note that the shift $S_k\, H_k = H_k + \frac{1}{k+1}$ is reflected by the action of σ on t.
2. Solving the telescoping problem in $(\mathbb{F}, \sigma)$*:* `Sigma` finds the solution $g = k(t-1) \in \mathbb{F}$ for

$$\sigma(g) - g = t.$$

Hence $g(k) = k(H_k - 1)$ is an "antidifference" for $f(k) = H_k$. The desired identity follows immediately. □

Loosely speaking, our difference fields are towers of certain transcendental field extensions (called $\Pi\Sigma$-extensions) where each transcendental element represents a sum (Σ^*-extension) or a product (Π-extension). More precisely, a difference field extension [4] $(\mathbb{F}(t), \sigma)$ of $(\mathbb{F}, \sigma)$ is a Π-extension (resp. Σ^*-extension [5]) if $\mathbb{F}(t)$ is a rational function field, $\sigma(t) = \alpha\, t$ (resp. $\sigma(t) = t + \alpha$) for some

[3] All fields in this paper are understood as having characteristic 0.
[4] As illustrated in Example 1, a difference field $(\mathbb{E}, \sigma')$ is a difference field extension of a difference field $(\mathbb{F}, \sigma)$ if $\mathbb{F}$ is a subfield of $\mathbb{E}$ and $\sigma'(g) = \sigma(g)$ for all $g \in \mathbb{F}$; usually we do not distinguish between σ and σ'.
[5] For the sake of simplicity we will introduce Σ-extensions only in Section 3.1; the introduced Σ^*-extensions are a slight simplification.

$\alpha \in \mathbb{F}$ and $\text{const}_\sigma \mathbb{F}(t) = \text{const}_\sigma \mathbb{F}$. A $\Pi\Sigma$-extension is either a Π- or a Σ-extension. Moreover, we say that $(\mathbb{F}, \sigma)$ is a *nested* $\Pi\Sigma$*-extension* of $(\mathbb{G}, \sigma)$ if $\mathbb{F} = \mathbb{G}(t_1) \dots (t_e)$ is a rational function field and $(\mathbb{G}(t_1) \dots (t_i), \sigma)$ is a $\Pi\Sigma$-extension of $(\mathbb{G}(t_1) \dots (t_{i-1}), \sigma)$ for all $1 \le i \le e$. If $\mathbb{K} := \mathbb{G}$ is the constant field of $(\mathbb{F}, \sigma)$ we say that $(\mathbb{F}, \sigma)$ is a $\Pi\Sigma$*-field over* $\mathbb{K}$; for further details see (9; 18; 25).

Of course not all expressions can be represented by $\Pi\Sigma$-fields, but such expressions may still be representable by difference fields which are not $\Pi\Sigma$-fields. For instance, a *free difference field* (6, Chapter 2.6) is a suitable difference field representation for an unspecified sequence $x(n)$. A free difference field can be seen as a rational function field $\mathbb{K}\langle x \rangle := \mathbb{K}(\dots, x_{-1}, x_0, x_1, x_2, \dots)$ together with σ defined by $\sigma(c) = c$ ($c \in \mathbb{K}$) and $\sigma(x_i) := x_{i+1}$. The field element x_0 represents the expression $x(n)$, x_1 represents $x(n+1)$, etc. This difference field is obviously not a $\Pi\Sigma$-field, and the question arises to which extent $\Pi\Sigma$-techniques are applicable. Sections 3 to 5 give algorithms that make it possible to proceed as outlined in the following example.

Example 2 In order to find a closed form evaluation of (2) we consider the difference field $(\mathbb{Q}\langle x \rangle(t), \sigma)$ where $(\mathbb{Q}\langle x \rangle, \sigma)$ is free and $(\mathbb{Q}\langle x \rangle(t), \sigma)$ is the $\Pi\Sigma$-extension defined by $\sigma(t) = x_1 t$. In this field we solve (8) with $f = (x_1 - 1)t$, and obtain $g = t$. This gives the solution $g(k) = \prod_{i=1}^{k} x(i)$ for the telescoping equation with $f(k) = (x(k+1) - 1) \prod_{i=1}^{k} x(i)$ which allows to derive (2). □

There is another subtlety with respect to symbolic summation over nested sums. Observe that so far we constructed the difference field $(\mathbb{F}, \sigma)$ for the telescoping problem (8) by adjoining only those sums and products that are involved in the summand $f(k)$ of the telescoping equation. But this is in many cases not sufficient, compare (3), (5), (6), and (7). In order to overcome this situation, we can use refined summation techniques from (21; 23). Namely, we search for suitable $\Pi\Sigma$-extensions in which a closed form evaluation exists.

Example 3 The summand $f(k) = k^2 \sum_{i=0}^{k} x(i)$ in identity (6) can be represented as $f = k^2 t$ in the $\Pi\Sigma$-extension $(\mathbb{Q}\langle x \rangle(k)(t), \sigma)$ of $(\mathbb{Q}\langle x \rangle, \sigma)$ with $\sigma(k) = k+1$ and $\sigma(t) = t + x_1$. The telescoping equation has no solution in this difference field, but with our refined summation tools, see problem *RTΠΣ* in Section 3, we find *automatically* the $\Pi\Sigma$-extension $(\mathbb{Q}\langle x \rangle(k)(t)(s_1)(s_2)(s_3), \sigma)$ of $(\mathbb{Q}\langle x \rangle(k)(t), \sigma)$ with $\sigma(s_i) = s_i + k^i x_1$ in which a telescoping solution of (8) exists. Namely, we obtain $g = \frac{1}{6}\Big((k-1)k(2k-1)t - s_1 + 3s_2 - 2s_3\Big)$ which finally gives (6). □

In a similar fashion the identities (3), (5), and (7) can be derived.

3 Indefinite Summation in ΠΣ-Extensions

As illustrated in Section 2 we are interested in two problems: the representation of the summand $f(k)$ in a suitable difference field and solving the telescoping problem (8) in this domain. We approach this goal by looking for algorithms that solve the following problem.

TΠΣ: **T**elescoping in **ΠΣ**-extensions

- **Given** a nested ΠΣ-extension $(\mathbb{F}, \sigma)$ of $(\mathbb{G}, \sigma)$ and $f \in \mathbb{F}$.
- **Find,** if possible, a $g \in \mathbb{F}$ with $\sigma(g) - g = f$.

The ground field $(\mathbb{G}, \sigma)$ is chosen such as to cover the expressions that occur in a particular summation problem at hand. In the simplest case, $\mathbb{G}$ is just a field of constants, but the problem formulation makes sense for any difference field.

In this section we work out that a particular application of Karr's algorithm (9) allows one to solve problem *TΠΣ* if there are algorithms for various subproblems in the ground field $(\mathbb{G}, \sigma)$. We use the fact that Karr's algorithm reduces by recursion all arising problems to subproblems in the ground field $(\mathbb{G}, \sigma)$. The solutions of these subproblems are then combined to a solution of the original problem *TΠΣ*.

In the following subsections we will analyze Karr's algorithm, in particular, a simplified version given by (26; 19; 22), in order to determine all the subproblems that have to be solved. Difference fields for which all those problems can be solved will be called *σ-computable;* a formal definition is given below.

As direct consequence we shall obtain also refined summation tools in such a σ-computable difference field. Namely, we can search for suitable ΠΣ-extensions in which a solution for the telescoping problem exists; see Example 3.

RTΠΣ: **R**efined **T**elescoping in **ΠΣ**-extensions

- **Given** a nested ΠΣ-extension $(\mathbb{F}, \sigma)$ of $(\mathbb{G}, \sigma)$ with $\mathbb{F} := \mathbb{G}(t_1) \dots (t_e)$, $f \in \mathbb{F}$ and $r \in \{0, \dots, e\}$.
- **Decide** if there exists a nested ΠΣ-extension $(\mathbb{F}(x_1) \dots (x_n), \sigma)$ of $(\mathbb{F}, \sigma)$ with $\sigma(x_i) = \alpha_i x_i + \beta_i$ and $\alpha_i, \beta_i \in \mathbb{G}(t_1) \dots (t_r)$ such that there is a $g \in \mathbb{F}(x_1) \dots (x_n)$ with (8). If yes, **compute** such an extension and such a g. [6]

Then, assuming that $(\mathbb{F}, \sigma)$ is σ-computable, refined telescoping can be handled as follows. Join the "simplest" sums and products in $\mathbb{G}(t_1) \dots (t_e)$ first. Then looking for the smallest possible r such that we obtain a solution in *RTΠΣ* gives the "simplest" solution g for (8).

3.1 A Constructive Theory of ΠΣ-extensions

In all our examples from Section 2 the summand $f(k)$ is represented in a tower of Π- or Σ*-extensions over a σ-computable ground field $(\mathbb{G}, \sigma)$, e.g., $\mathbb{G} = \mathbb{K}$

[6] By difference field theory (21) it suffices to look for nested Σ*-extensions, i.e., $\sigma(x_i) - x_i \in \mathbb{G}(t_1) \dots (t_r)$.

or $\mathbb{G} = \mathbb{K}\langle x \rangle$. By using results from (9) it turns out that this construction can be carried out completely algorithmically.

In order to accomplish this task, we use the following facts for Π- and Σ^*-extensions defined in Section 2; see (9) and (25) for further explanations. Given any difference field extension $(\mathbb{F}(t), \sigma)$ of $(\mathbb{F}, \sigma)$, the following holds: (1) This is a Π-extension iff $\sigma(t) = \alpha\, t$, $t \neq 0$, $\alpha \in \mathbb{F}^*$ and there are no $n > 0$ and $g \in \mathbb{F}$ with $\sigma(g) = \alpha^n g$. (2) This is a Σ^*-extension iff $\sigma(t) = t + \alpha$, $t \notin \mathbb{F}$, $\alpha \in \mathbb{F}^*$, and there is no $g \in \mathbb{F}$ with $\sigma(g) - g = \alpha$.

In particular, this result states that indefinite summation/telescoping and the construction of a Σ^*-extension are very closely related. Namely, one can either adjoin a sum in form of a Σ^*-extension with $\sigma(t) = t + \beta$ to $(\mathbb{F}, \sigma)$, or one can express this sum by a $g \in \mathbb{F}$ with $\sigma(g) = g + \beta$. The product case can be handled essentially in the same way; see (25).

We call

$$\mathrm{H}_{(\mathbb{F},\sigma)} := \{\, \sigma(g)/g \mid g \in \mathbb{F}^* \,\}$$

the *homogeneous group* of a difference field $(\mathbb{F}, \sigma)$. We can construct a nested ΠΣ-extension over a given ground field $(\mathbb{G}, \sigma)$ step by step if we know how to solve the following problem.

CΠΣ: **C**onstruction of **ΠΣ**-extensions

- **Given** a nested ΠΣ-extension $(\mathbb{F}, \sigma)$ of $(\mathbb{G}, \sigma)$ and $\alpha, \beta \in \mathbb{F}$.
- **Decide** if there is an $n > 0$ with $\alpha^n \in \mathrm{H}_{(\mathbb{F},\sigma)}$ (The *homogeneous group problem* for Π-extensions).
- **Find,** if possible, a $g \in \mathbb{F}$ with $\sigma(g) - \alpha\, g = \beta$ (for Σ^*- and Σ-extensions).

Together with results from Sections 3.2 and 3.3 we can handle this problem if the ground field $(\mathbb{G}, \sigma)$ is σ-computable; see Definition 2.

For completeness reasons we introduce Σ-extensions, a slightly more general form of Σ^*-extensions; see (9; 18; 25). An extension $(\mathbb{F}(t), \sigma)$ of $(\mathbb{F}, \sigma)$ is a Σ-*extension* if $\mathbb{F}(t)$ is a rational function field, $\sigma(t) = \alpha\, t + \beta$ with $\alpha, \beta \in \mathbb{F}^*$ and $\mathrm{const}_\sigma \mathbb{F}(t) = \mathrm{const}_\sigma \mathbb{F}$ where the following two properties hold: (1) there does not exist a $g \in \mathbb{F}$ with $\sigma(g) - \alpha g = \beta$, and (2) if $\alpha^n \in \mathrm{H}_{(\mathbb{F},\sigma)}$ for some $n \in \mathbb{Z}^*$ then $\alpha \in \mathrm{H}_{(\mathbb{F},\sigma)}$. Note that we can decide algorithmically whether we can adjoin a Σ-extension by solving certain instances of problem *CΠΣ*.

3.2 The Homogeneous Group Problem and Subproblems

Before we get to the definition of σ-computability, we introduce the notion of σ^*-computability as an intermediate step. This notion will be defined in such a way that we can solve the homogeneous group problem of *CΠΣ* in any nested ΠΣ-extension $(\mathbb{G}(t_1) \dots (t_e), \sigma)$ of $(\mathbb{G}, \sigma)$ if $(\mathbb{G}, \sigma)$ is σ^*-computable.

We call a difference field $(\mathbb{F}, \sigma)$ *torsion free* if

$$\forall r \in \mathbb{Z}^* \, \forall f \in \mathrm{H}_{(\mathbb{F},\sigma)} : f^r = 1 \Rightarrow f = 1. \tag{9}$$

Moreover, we define for a difference field $(\mathbb{F}, \sigma)$ and $r \geq 0$ the *σ-factorial*

$$f_{(r,\sigma)} := f \cdot \sigma(f) \cdots \sigma^{r-1}(f).$$

In particular, $f_{(0,\sigma)} := 1$. If σ is clear from the context, we shall abbreviate $f_{(r)} := f_{(r,\sigma)}$.

Definition 1 A difference field $(\mathbb{F}, \sigma)$ is called *σ^*-computable* if the following holds.

(1) There is an algorithm that can factor multivariate polynomials over $\mathbb{F}$.

(2) $(\mathbb{F}, \sigma^k)$ is torsion free for all $k \in \mathbb{Z}^*$.

(3) There is an algorithm that can solve problem *ΠReg*:

ΠReg: **Π-Reg**ularity

- **Given** $(\mathbb{F}, \sigma)$ and $f, g \in \mathbb{F}^*$.
- **Find,** if possible, an $n \geq 0$ with $f_{(n)} = g$.

(4) There is an algorithm that can solve problem *ΣReg*:

ΣReg: **Σ-Reg**ularity

- **Given** $(\mathbb{F}, \sigma)$, $k \in \mathbb{Z}^*$ and $f, g \in \mathbb{F}^*$.
- **Find,** if possible, an $n \geq 0$ with $f_{(0,\sigma^k)} + \cdots + f_{(n,\sigma^k)} = g$.

(5) There is an algorithm that can solve problem *OHG*:

OHG: **O**rbits of the **H**omogeneous **G**roup

- **Given** $(\mathbb{F}, \sigma)$ and $\alpha_1, \ldots, \alpha_r \in \mathbb{F}^*$.
- **Find** a basis of the submodule $\{ (n_1, \ldots, n_r) \in \mathbb{Z}^r \mid \alpha_1^{n_1} \ldots \alpha_r^{n_r} \in \mathrm{H}_{(\mathbb{F},\sigma)} \}$ of $\mathbb{Z}^r$ over $\mathbb{Z}$.

Theorem 1 Let $(\mathbb{F}(t), \sigma)$ be a ΠΣ-extension of $(\mathbb{F}, \sigma)$. If $(\mathbb{F}, \sigma)$ is σ^*-computable then $(\mathbb{F}(t), \sigma)$ is σ^*-computable.

Proof. Suppose that $(\mathbb{F}, \sigma)$ is σ^*-computable. Then first observe that by (9, Lemma 4) the extension $(\mathbb{F}(t), \sigma^k)$ of $(\mathbb{F}, \sigma)$ is a ΠΣ-extension for all $k \in \mathbb{Z}^*$. We have:

(1) There is an algorithm that can factor multivariate polynomials over $\mathbb{F}(t)$: this follows since there is an algorithm that can factor multivariate polynomials over $\mathbb{F}$, hence over $\mathbb{F}[t]$, and consequently over $\mathbb{F}(t)$.
(2) $(\mathbb{F}(t), \sigma)$ is torsion free: this follows by (9, Lemma 5) and the assumption that $(\mathbb{F}, \sigma)$ is torsion free.
(3) There is an algorithm that solves problem *ΠReg* in $(\mathbb{F}(t), \sigma)$: this follows

by (9, Theorem 5) and the assumption that one can solve problem *ΠReg* in $(\mathbb{F}, \sigma)$.
(4) There is an algorithm that solves problem *ΣReg* in $(\mathbb{F}(t), \sigma)$: this follows by (9, Theorem 6) and the assumptions that one can factor polynomials $\mathbb{F}[t]$, $(\mathbb{F}, \sigma^k)$ is torsion free for all $k \in \mathbb{Z}$, and one can solve problem *ΣReg*.
(5a) Using the already proven statements (3) and (4), it follows by (9, Theorem 4) that there is an algorithm for problem

SE: **S**hift **E**quivalence in a ΠΣ-extension

- **Given** a ΠΣ-extension $(\mathbb{F}(t), \sigma)$ of $(\mathbb{F}, \sigma)$ and $f, g \in \mathbb{F}(t)^*$.
- **Find,** if possible, an $n \in \mathbb{Z}$ with $\sigma^n(f)/g \in \mathbb{F}$.

(5) There is an algorithm that solves problem *OHG* in $(\mathbb{F}(t), \sigma)$: This follows by Theorem 7 and Theorem 8 of (9) by using the fact that one can factor polynomials in $\mathbb{F}[t]$, there is an algorithm for problem *OHG* in $(\mathbb{F}, \sigma)$, and statement (5a) holds. □

With the proof step (5a) for Theorem 1 there is the following fact needed in Section 3.3.

Corollary 1 Let $(\mathbb{F}(t), \sigma)$ be a ΠΣ-extension of $(\mathbb{F}, \sigma)$. If $(\mathbb{F}, \sigma)$ is σ^*-computable then one can solve problem *SE*.

Summarizing, one can lift the property σ^*-computable from the ground field $(\mathbb{G}, \sigma)$ to any nested ΠΣ-extension $(\mathbb{F}, \sigma)$ with $\mathbb{F} = \mathbb{G}(t_1) \dots (t_e)$. Hence one can solve problem *OHG* in any nested ΠΣ-extension $(\mathbb{F}, \sigma)$ of a σ^*-computable $(\mathbb{G}, \sigma)$, which is a generalization of the homogeneous group problem in *CΠΣ*.

3.3 Parameterized First Order Linear Difference Equations

In order to handle problem *TΠΣ* and *CΠΣ* we consider the following more general problem.

PFLDE: **P**arameterized **F**irst Order **L**inear **D**ifference **E**quations

- **Given** $(\mathbb{F}, \sigma)$ with $\mathbb{K} := \text{const}_\sigma \mathbb{F}$, $a_1, a_2 \in \mathbb{F}^*$ and $(f_1, \dots, f_n) \in \mathbb{F}^n$.
- **Find all** $g \in \mathbb{F}$ and $(c_1, \dots, c_n) \in \mathbb{K}^n$ with $a_1\, \sigma(g) + a_2\, g = c_1\, f_1 + \dots + c_n\, f_n$.

Definition 2 A difference field $(\mathbb{F}, \sigma)$ is σ-computable if it is σ^*-computable and there is an algorithm that solves problem *PFLDE*.

Theorem 2 Let $(\mathbb{F}(t), \sigma)$ be a ΠΣ-extension of $(\mathbb{F}, \sigma)$. If $(\mathbb{F}, \sigma)$ is σ-computable then $(\mathbb{F}(t), \sigma)$ is σ-computable.

We outline the algorithm given by (26; 19; 22), which only makes use of properties of the ground field $(\mathbb{F}, \sigma)$ that are included in the definition of σ-computability. By Corollary 1 we have an algorithm to solve problem *SE*. We can apply the following chain of reductions.

Reduction I (denominator bounding). Find a polynomial $d \in \mathbb{F}[t]^*$ such that for all $c_i \in \mathbb{K}$ and $g \in \mathbb{F}(t)$ with

$$a_1\,\sigma(g) + a_2\,g = c_1\,f_1 + \dots + c_n\,f_n \tag{10}$$

we have $d\,g \in \mathbb{F}[t]$. Then it follows that

$$\frac{a_1}{\sigma(d)}\,\sigma(g') + \frac{a_2}{d} g' = c_1\,f_1 + \dots + c_n\,f_n$$

for $g' \in \mathbb{F}[t]$ if and only if (10) with $g = g'/d$. Such a polynomial d is called a *denominator bound.* Using results from (9; 3), Schneider (19) has given an algorithm for computing a denominator bound $d \in \mathbb{F}[t]^*$. This algorithm requires solving problems of type *SE* in the $\Pi\Sigma$-extension $(\mathbb{F}(t), \sigma)$ of $(\mathbb{F}, \sigma)$ and of type *OHG* in $(\mathbb{F}, \sigma)$. We can do this by assumption. After computing a denominator bound, it suffices to look only for $c_i \in \mathbb{K}$ and polynomial solutions $g \in \mathbb{F}[t]$ with (10).

Reduction II (degree bounding). Next, we look for a degree bound $b \in \mathbb{N}_0$ for the polynomial solutions. By (9), see (22) for further details, there is an algorithm that computes such a degree bound if there are algorithms that solve problems *PFLDE* and *OHG* in $(\mathbb{F}, \sigma)$. By assumption we can solve these problems.

Reduction III (polynomial degree reduction). Given this degree bound one looks for $c_i \in \mathbb{K}$ and $g_i \in \mathbb{F}$ such that (10) holds for $g = \sum_{i=0}^{b} g_i t^i$. Loosely speaking, this can be achieved as follows. First derive the possible leading coefficients g_b by solving a specific instance of problem *PFLDE* in $(\mathbb{F}, \sigma)$, then plug in the corresponding solutions into (10) and look for the remaining solutions $g = \sum_{i=0}^{b-1} g_i t^i$ by recursion. Summarizing, one can derive the solutions for (10) by solving several problems of the type *PFLDE* in $(\mathbb{F}, \sigma)$.

Corollary 2 There is an algorithm that solves problems *PFLDE* and *CΠΣ* for a nested $\Pi\Sigma$-extension $(\mathbb{F}, \sigma)$ of $(\mathbb{G}, \sigma)$ when $(\mathbb{G}, \sigma)$ is σ-computable.

Problem *RTΠΣ* (and generalized versions given in (23)) can be solved in a nested $\Pi\Sigma$-extension $(\mathbb{G}(t_1)\dots(t_e), \sigma)$ of $(\mathbb{G}, \sigma)$, if the degree and denominator bounding algorithms given in (19; 22) can be applied in all $\Pi\Sigma$-extensions $(\mathbb{G}(t_1)\dots(t_i), \sigma)$ of $(\mathbb{G}(t_1)\dots(t_{i-1}), \sigma)$, see (21).

As shown in the reduction above, this is possible when $(\mathbb{G}, \sigma)$ is σ-computable.

Corollary 3 There is an algorithm that solves Problem *RTΠΣ* for a nested $\Pi\Sigma$-extension $(\mathbb{F}, \sigma)$ of $(\mathbb{G}, \sigma)$ where $(\mathbb{G}, \sigma)$ is σ-computable.

4 Special Case: ΠΣ-Fields

We have defined a $\Pi\Sigma$-field as a tower of $\Pi\Sigma$-extensions over a field $\mathbb{K}$ of constants. There are some requirements (9, Theorem 9) which the underlying constant field has to fulfill such that all the sub-problems presented earlier can be solved algorithmically. Our notion of σ-computability generalizes these requirements.

Theorem 3 Let $(\mathbb{K}, \sigma)$ be a constant field, i.e., $\sigma(k) = k$ $(k \in \mathbb{K})$. Assume that $\mathbb{K}$ has the following properties: (1) for any $k \in \mathbb{K}$ it can be decided if $k \in \mathbb{Z}$; (2) multivariate polynomials over $\mathbb{K}$ can be factored; (3) for any vector $(c_1, \ldots, c_k) \in \mathbb{K}^k$, a basis of the module

$$\{ (n_1, \ldots, n_k) \in \mathbb{Z}^k \mid c_1^{n_1} \cdots c_k^{n_k} = 1 \} \subseteq \mathbb{Z}^k$$

can be computed.

Then $(\mathbb{K}, \sigma)$ is σ-computable.

Proof. It is immediate that any constant field is torsion-free, because $\mathrm{H}_{(\mathbb{K},\sigma^i)} = \{1\}$ for all $i \in \mathbb{N}$. Π-regularity can be decided using (9, Lemma 2), due to property (3). Σ-regularity can be decided using (9, Lemma 3) and properties (1) and (2). *OHG* is property (3). *PFLDE* only requires solving a linear system as $\mathbb{K}$ is a constant field. □

For example, any rational function field over an algebraic number field is σ-computable (25).

5 Application of Free Difference Fields

In this section, we show that a free difference field $(\mathbb{K}\langle x\rangle, \sigma)$ is σ-computable if the underlying constant field $\mathbb{K}$ is. Together with the results of Section 3, this provides a complete algorithmic framework that produces identities with unspecified sequences $x(k)$, as indicated in the introduction. Recall the definition $\mathbb{K}\langle x\rangle = \mathbb{K}(\ldots, x_{-1}, x_0, x_1, \ldots)$ with $\sigma(x_k) = x_{k+1}$.

Even though $\mathbb{K}\langle x\rangle$ has infinitely many indeterminates, each particular element does only involve finitely many of them. Therefore, for any $f \in \mathbb{K}\langle x\rangle \setminus \mathbb{K}$ we may define $\max\operatorname{ord}(f)$ as the maximum $r \in \mathbb{Z}$ such that x_r occurs in f. The minimum order $\min\operatorname{ord}(f)$ is defined analogously. For convenience, we may put $\max\operatorname{ord}(f) := -\infty$ and $\min\operatorname{ord}(f) := \infty$ when $f \in \mathbb{K}$. Reasoning about the order of elements leads to a rather straightforward algorithmic treatment of $(\mathbb{K}\langle x\rangle, \sigma)$. We will make free use of obvious relations such as $\max\operatorname{ord}(\sigma f) = 1 + \max\operatorname{ord}(f)$ $(f \in \mathbb{K}\langle x\rangle)$ or $\max\operatorname{ord}(f \cdot g) = \max\{\max\operatorname{ord}(f), \max\operatorname{ord}(g)\}$ $(f, g \in \mathbb{K}\langle x\rangle^*)$.

Theorem 4 If $(\mathbb{K}, \sigma)$ is a σ-computable constant field, then $(\mathbb{K}\langle x\rangle, \sigma)$ is σ-computable as well.

The rest of this section is devoted to the proof of this theorem.

First, $(\mathbb{K}\langle x\rangle, \sigma)$ is torsion free: Let $f \in \mathrm{H}_{(\mathbb{K}\langle x\rangle,\sigma^i)}$ with $f^k = 1$ for some i and k. By definition, there is a $g \in \mathbb{K}\langle x\rangle$ with $f = \sigma^i(g)/g$, so $1 = f^k = \sigma^i(g)^k/g^k = \sigma^i(g^k)/g^k$, so $g^k = \sigma^i(g^k)$. This shows that $g^k \in \mathbb{K}$, because otherwise, g^k and $\sigma^i(g^k)$ would have different order and could hence not be equal. But $(\mathbb{K}, \sigma)$ is torsion-free, so $f = 1$.

We now provide algorithms for solving the required problems.

5.1 *ΠΣ-Regularity*

Consider the problem *ΠΣReg* in $(\mathbb{K}\langle x\rangle, \sigma)$. Let $f, g \in \mathbb{K}\langle x\rangle$. If both f, g belong to $\mathbb{K}$, σ-computability of $\mathbb{K}$ applies. If only one of them belongs to $\mathbb{K}$ and the other one does not, then there cannot be an $n \in \mathbb{N}$ with $g = f_{(n)}$. Now suppose that neither of f, g belongs to $\mathbb{K}$. For all $n \geq 0$, we have $\min \operatorname{ord}(f_{(n)}) = \min \operatorname{ord}(f)$ and $\max \operatorname{ord}(f_{(n)}) = \max \operatorname{ord}(f) + n - 1$ by the definition of $f_{(n)}$. Comparing the orders of f and g one obtains at most one candidate n which may satisfy the required equation $g = f_{(n)}$. For this candidate, compute $f_{(n)}$ and compare it to g.

Σ-regularity can be decided by a similar reasoning.

5.2 *The Orbits of the Homogeneous Group*

Consider problem *OHG* in $(\mathbb{K}\langle x\rangle, \sigma)$. First observe that given $f, g \in \mathbb{K}\langle x\rangle$, we can decide whether f and g are *shift equivalent* (problem *SE*): consideration of the orders of f and g gives at most one candidate n for which $\sigma^n(f)/g \in \mathbb{K}$ could possibly be the case. For this candidate n, $\sigma^n(f)/g \in \mathbb{K}$ can be decided by inspection. Decidability of shift equivalence together with the ability to factor multivariate polynomials over $\mathbb{K}$ allows the computation of so-called σ-factorizations (9, Definition 23) of vectors $(f_1, \ldots, f_n) \in \mathbb{K}\langle x\rangle^n$.

Using σ-factorizations in $\mathbb{K}\langle x\rangle$ and the σ-computability of $\mathbb{K}$, *OHG* can be solved in analogy to (9, Theorem 8).

5.3 *Solving Linear Difference Equations*

We complete the discussion by presenting an algorithm for solving *PFLDE* in $(\mathbb{K}\langle x\rangle, \sigma)$. The algorithm transforms the problem to a system of linear equations over $\mathbb{K}$ by a chain of several reductions, similar as in Section 3.3. In view of possible further generalizations, motivated by (26), we consider the following more general problem.

Given $a_1, \ldots, a_m \in \mathbb{K}\langle x\rangle$ and $f_1, \ldots, f_n \in \mathbb{K}\langle x\rangle$, find all $g \in \mathbb{K}\langle x\rangle$ and $c_1, \ldots, c_n \in \mathbb{K}$ such that

$$a_1\sigma^{m-1}(g) + a_2\sigma^{m-2}(g) + \cdots + a_{m-1}\sigma(g) + a_m g = c_1 f_1 + c_2 f_2 + \cdots + c_n f_n. \tag{11}$$

PFLDE is included here for $m = 2$.

Reduction I (denominator bounding) First we reduce (11) from $\mathbb{K}\langle x\rangle$ to the *polynomial difference ring* $\mathbb{K}\{x\} := \mathbb{K}[\dots, x_{-1}, x_0, x_1, \dots]$. As $\sigma^k(f)/f \notin \mathbb{K}$ for all $f \in \mathbb{K}\langle x\rangle \setminus \mathbb{K}$ and $k > 0$, we can apply Abramov's denominator bounding algorithm (1; 3; 19) without complication. As in Section 3.3, denominator bounding reduces the rational function problem to a polynomial problem.

Reduction II (order bounding) It remains to consider equation (11) for the case where the a_i and f_i are in $\mathbb{K}\{x\}$ and a solution $g \in \mathbb{K}\{x\}$ is required. If g is such a solution and ρ is the maximum of the $\max\operatorname{ord}(a_i)$ and $\max\operatorname{ord}(f_i)$, then g must be free of x_k for all $k > \rho - m + 1$. This can be seen as follows. Suppose $\max\operatorname{ord}(g) = k > \rho - m + 1$. Then $\sigma^{m-1}(g)$ contains a term of order $k + m - 1 > \rho$. As no such term can occur in $\sigma^i(g)$ $(i < m - 1)$ or in a_i, the left hand side of (11) has order $k+m-1$. But the right hand side has at most order ρ. So g cannot be a solution, in contradiction to the assumption.

A lower bound for $\min\operatorname{ord}(g)$ can be found by a similar argument. For clarity of notation, we may assume for the next steps without loss of generality that 0 bounds $\min\operatorname{ord}(g)$ from below and r is an upper bound for $\max\operatorname{ord}(g)$.

Reduction III (degree bounding) Now, after bounding the denominator and the orders, and after clearing denominators, we may consider (11) as an equation in the polynomial ring $\mathbb{K}[x_0, \dots, x_r]$. We seek a polynomial solution g in this ring. For efficiency reasons, we will show how to compute separate degree bounds for each indeterminate x_i, rather than using an overshooting bound for the total degree.

We write $a_i = \sum_k a_{i,k} x_0^k$, $f_i = \sum_k f_{i,k} x_0^k$ and make the ansatz $g = \sum_k g_k x_0^k$. The $a_{i,k}$, $f_{i,k}$ and g_k are supposed to be free of x_0. A degree bound for g in x_0 is readily found using that $\sigma^k(g)$ is free of x_0 for all $k > 0$. Based on this starting point, we will incrementally find degree bounds for the $g_{i,k}$ with respect to x_1, then degree bounds for the coefficients w.r.t. x_2, etc., as follows.

For obtaining a degree bound of g_k w.r.t. x_1, we plug in the ansatz for g into the difference equation (11). This gives

$$\sum_k a_{m,k} x_0^k \sum_k g_k x_0^k + \sum_{j=1}^{m-1} \sum_k a_{j,k} x_0^k \sum_k \underbrace{\sigma^{m-j}(g_k)}_{\text{free of } x_1} x_{m-j} = c_1 f_1 + \cdots + c_n f_n.$$

We now iterate through all terms x_0^k in the product $\sum_k a_{m,k} x_0^k \sum_k g_k x_0^k$ and compare coefficients. This leads to equations of the form $p(x_1) g_k = q(x_1)$ for some polynomials p, q, from which lower and upper bounds for the degree of x_1 in g_k can be read of. We need not worry about the symbolic $\sigma^{m-j}(g_k)$ contained in q, because they are free of x_1.

Having degree bounds for all the g_k w.r.t. x_1 at hand, we write

$$a_i = \sum_{k,l} a_{i,k,l} x_0^k x_1^l, \qquad f_i = \sum_{k,l} f_{i,k,l} x_0^k x_1^l$$

and we refine the ansatz $g = \sum_k g_k x_0^k$ by $g_k = \sum_l g_{k,l} x_1^l$. Comparing coefficients of $x_0^k x_1^l$ leads to degree bounds w.r.t. x_2, and we proceed in the same way until the bound r for the maximum order is reached.

The result of this procedure is a finite set of candidate terms that may possibly occur in a solution. Compared to a naively computed total degree bound, the procedure described here offers a severe reduction of the computational overhead. The set of candidate terms is often optimal in practice, or overshooting by a few terms only. Naive bounds for the total degree, on the other hand, overshoot by a factor of up to 100 on certain examples.

Reduction IV (coefficient comparison) Based on the set of terms computed in the previous step, we make an ansatz with undetermined coefficients for the solution, plug it into the difference equation and compare coefficients. This leads to a linear system over $\mathbb{K}$.

6 Conclusion

We were able to extend the algorithmic theory of ΠΣ-fields to new types of difference fields. Indefinite nested summation over sequences represented by any difference field is now possible, provided that the difference field satisfies the stated sufficient conditions (σ-computability, Def. 2). With this approach, we obtained a new summation algorithm for dealing with unspecified sequences.

Concerning unspecified sequences, summation techniques beyond indefinite summation still have to be investigated. Since problem *PFLDE* allows to model creative telescoping (15) in the difference field setting (see (20) for the ΠΣ-case) our ideas apply also to definite summation; for some results in this direction see (12). As also higher order difference equations can be solved in free difference fields, our summation framework can possibly be extended to indefinite and definite summation with summands involving ∂-finite expressions as well (5; 24).

Instead of summation of unspecified summands, we may also consider the problem of integration of unspecified functions. Campbell (4) presents a simple integration procedure for this purpose. Extending the Risch algorithm for symbolic integration (16; 17) as done in the present paper for Karr's summation algorithm (using a free differential field for representing unspecified functions) should lead to a more powerful integration routine than Campbell's. For example, identities like

$$\int_0^x \int_0^t f(\tau)\,d\tau\,dt = x\int_0^x f(t)\,dt - \int_0^x t f(t)\,dt$$

(compare (6) above) should be possible to find automatically.

References

[1] S.A. Abramov. Rational solutions of linear difference and q-difference equations with polynomial coefficients. In T. Levelt, editor, *Proc. ISSAC'95*, pages 285–289. ACM Press, New York, 1995.

[2] G.E. Andrews and P. Paule. MacMahon's partition analysis IV: Hypergeometric multisums. *Sém. Lothar. Combin.*, B42i:1–24, 1999.

[3] M. Bronstein. On solutions of linear ordinary difference equations in their coefficient field. *J. Symbolic Comput.*, 29(6):841–877, June 2000.

[4] Graham H. Campbell. Symbolic integration of expressions involving unspecified functions. *Sigsam Bulletin*, 22(1):25–27, 1987.

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

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

[7] R.W. Gosper. Decision procedures for indefinite hypergeometric summation. *Proc. Nat. Acad. Sci. U.S.A.*, 75:40–42, 1978.

[8] R.L. Graham, D.E. Knuth, and O. Patashnik. *Concrete Mathematics*. Addison-Wesley Publishing Company, Amsterdam, 2nd edition, 1994.

[9] M. Karr. Summation in finite terms. *J. ACM*, 28:305–350, 1981.

[10] M. Kauers. Computer proofs for polynomial identities in arbitrary many variables. In *Proceedings of ISSAC'04*, pages 199–204, July 2004.

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

[12] Manuel Kauers and Carsten Schneider. Application of unspecified sequences in symbolic summation. In *Proceedings of ISSAC'06*, 2006.

[13] P. Paule. Greatest factorial factorization and symbolic summation. *J. Symbolic Comput.*, 20(3):235–268, 1995.

[14] P. Paule and A. Riese. A Mathematica q-analogue of Zeilberger's algorithm based on an algebraically motivated aproach to q-hypergeometric telescoping. In M. Ismail and M. Rahman, editors, *Special Functions, q-Series and Related Topics*, volume 14, pages 179–210. Fields Institute Toronto, AMS, 1997.

[15] M. Petkovšek, H.S. Wilf, and D. Zeilberger. $A = B$. A.K. Peters, Wellesley, MA, 1996.

[16] R. Risch. The problem of integration in finite terms. *Trans. Amer. Math. Soc.*, 139:167–189, 1969.

[17] R. Risch. The solution to the problem of integration in finite terms. *Bull. Amer. Math. Soc.*, 76:605–608, 1970.

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

[19] 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.

[20] Carsten Schneider. The summation package Sigma: Underlying principles

and a rhombus tiling application. *Discrete Mathematics and Theoretical Computer Science*, 6(2):365–386, 2004.

[21] Carsten Schneider. Symbolic summation with single-nested sum extensions. In *Proceedings of ISSAC '04*, pages 282–289, July 2004.

[22] Carsten Schneider. Degree bounds to find polynomial solutions of parameterized linear difference equations in ΠΣ-fields. *Applicable Algebra in Engeneering, Communication and Computing*, 16(1):1–32, 2005.

[23] Carsten Schneider. Finding telescopers with minimal depth for indefinite nested sum and product expressions. In *Proceedings of ISSAC '05*, pages 285–292, July 2005.

[24] Carsten Schneider. A new Sigma approach to multi-summation. *Advances in Applied Mathematics*, 34(4):740–767, 2005.

[25] Carsten Schneider. Product representations in ΠΣ-fields. *Annals of Combinatorics*, 9(1):75–99, 2005.

[26] 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.

[27] A. van der Poorten. A proof that Euler missed... Apéry's proof of the irrationality of $\zeta(3)$. *Math. Intelligencer*, 1:195–203, 1979.

[28] Z. Zhang. A kind of binomial identity. *Discrete Math.*, 196(1-3):291–298, 1999.