Algorithms for Nonlinear Higher Order Difference Equations

Johannes-Kepler University Linz
Research Institute for Symbolic Computation

Algorithms for Nonlinear Higher Order Difference Equations

Manuel Kauers

Doctoral Thesis

advised by

Univ.-Prof. Dr. phil. Peter Paule

examined by

Univ.-Prof. Dr. phil. Peter Paule
a. Univ.-Prof. Dr. techn. Josef Schicho

Defended on October, 7th, 2005 in Linz, Austria.

The author was supported in parts by

- The Upper Austrian Government (2002/10–2003/12)
- The German Academic Exchange Service (DAAD) grant D/03/40515 (2004/01–2005/07)
- The Austrian Science Foundation (FWF) grant P16613-N12 (2005/08)
- The Austrian Science Foundation (FWF) grant SFB F 1305 (travel expenses and infra structure)

Eidesstattliche Erklärung

Ich erkläre an Eides statt, daß ich die vorliegende Dissertation selbstständig und ohne fremde Hilfe verfaßt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Linz, 31. August 2005

. . . .
Manuel Kauers

Zusammenfassung

In dieser Arbeit werden neue algorithmische Verfahren zur Behandlung spezieller Zahlen- und Funktionenfolgen vorgestellt. Die Folgen, die wir betrachten, werden durch Systeme von Differenzengleichungen (Rekurrenzen) beschrieben. Diese Systeme können gekoppelt, nichtlinear und/oder von höherer Ordnung sein. Die Klasse der so definierbaren Folgen (zulässige Folgen) enthält viele Folgen, die in verschiedenen mathematischen Anwendungen von Interesse sind. Während ein Teil dieser Folgen mit bereits bekannten Algorithmen behandelt werden kann, standen für viele weitere bisher keine geeigneten automatischen Verfahren zur Verfügung.

Im Mittelpunkt unseres Interesses stehen Algorithmen, mit denen bekannte Identitäten zwischen zulässigen Folgen automatisch bewiesen und neue Identitäten automatisch gefunden werden können. Zum Finden neuer Identitäten zählen wir dabei das Lösen von Differenzengleichungen in geschlossener Form, das Finden geschlossener Darstellungen von symbolischen Summen sowie das Finden von algebraischen Abhängigkeiten zwischen gegebenen Folgen. Außerdem geben wir ein Verfahren an, mit dem manche Ungleichungen für zulässige Folgen automatisch bewiesen werden können.

Zur algorithmischen Verarbeitung werden zulässige Folgen als Elemente von speziell konstruierten Differenzenringen dargestellt. In diesen Differenzenringen werden dann Berechnungen durchgeführt, deren Ergebnisse sich als Aussagen über die ursprünglichen zulässigen Folgen interpretieren lassen. Bekannte Techniken für kommutative multivariate Polynomringe, insbesondere die Theorie der Gröbnerbasen, werden dabei eingesetzt.

Teil der vorliegenden Arbeit ist eine Implementierung der vorgestellten Algorithmen in Form eines Software-Pakets für das Computeralgebra-System Mathematica. Mit Hilfe unserer Software ist es gelungen, viele Gleichungen und Ungleichungen aus der Literatur erstmals automatisch zu beweisen. Außerdem haben wir mit derselben Software einige Identitäten gefunden, die wahrscheinlich bisher unbekannt waren.

Stichwörter

Differenzengleichungen, Rekurrenzen, Identitäten, Symbolische Summation, Differenzenringe, Gröbnerbasen.

Abstract

In this thesis, new algorithmic methods for the treatment of special sequences are presented. The sequences that we consider are described by systems of difference equations (recurrences). These systems may be coupled, non-linear, and/or higher order. The class of sequences defined in this way (admissible sequences) contains a lot of sequences which are of interest in various mathematical applications. While some of these sequences can be handled also with known Algorithms, for many others no adequate methods were available up to now.

In the center of our interest, there are algorithms for automatically proving known identities of admissible sequences, and for automatically discovering new ones. By "finding new identities", we mean in particular solving of difference equations in closed form, finding closed forms for symbolic sums, and finding algebraic dependencies of given sequences. In addition, we present a procedure by which some inequalities of admissible sequences can be proven automatically.

For their algorithmic treatment, admissible sequences are represented as elements of certain special difference rings. In these difference rings, computations are then carried out, whose results can be interpreted as statements about the original admissible sequences. Known techniques for commutative multivariate polynomial rings, especially the theory of Gröbner bases, are applied to this end.

Part of the present thesis is an implementation of the presented algorithms in form of a software package for the computer algebra system Mathematica. With the aid of our software, we succeeded in proving a lot of identities and inequalities from the literature automatically for the first time. Additionally, with the same software, we have found some identities which were probably unknown up to now.

Keywords

Difference equations, recurrences, identities, symbolic summation, difference rings, Gröbner bases.

Contents

1 Introduction

The whole is more than the sum of its parts.
——— Aristotle

The guiding principle in symbolic summation may be summarized as follows: if a quantity is given in terms of an expression involving summation quantifiers $\sum$, we are interested in finding a representation of the same quantity without—or at least with fewer—summation signs. Such a representation is then referred to as a *closed form,* its systematic computation is called *symbolic summation.* Despite the fact that summation, like integration, depends on experience, intuition, and a "sharp eye" when it is done by hand, it turned out in the last decades that it can well be done by the computer to a great extent. Symbolic summation has meanwhile reached a state where problems of considerable difficulty can be solved automatically. It is now becoming a tool of increasing importance in various branches of mathematics, especially in combinatorics.

Algorithms for symbolic summation operate on *difference equations.* Difference equations are the discrete analogue of differential equations, with the shift $f(n+1)$ playing the role of the derivation $f'(x)$. Computing a closed form for the sum $\sum_{k=1}^{n} f(k)$ amounts to computing a solution $F(n)$ of the inhomogeneous first order linear difference equation $F(n+1) - F(n) = f(n+1)$, called the *telescoping equation.* A whole bunch of algorithms is known for doing summation, and, more generally, for solving difference equations. These algorithms differ from each other in the class of summands that they can handle, and in the kind of closed forms that they offer as results.

Classical symbolic summation focuses on sums whose summand $f(k)$ is a so-called *hypergeometric* term. A hypergeometric term is a solution of a homogeneous first order linear difference equation whose coefficients are polynomials in n. Examples include exponentials, the factorial, binomial coefficients, and products of these. The algorithms of Gosper (1978), Zeilberger (1991), and Petkovšek (1992) provide together a complete algorithmic framework by which it is possible to decide whether a sum $F(n) = \sum_{k=1}^{n} f(n,k)$ with $f(n,k)$ being a hypergeometric term with respect to both n and k admits a closed form as a linear combination of hypergeometric terms, and to compute such a closed form if one exists. Petkovšek *et al.* (1997) give a comprehensive account on these results. Implementations are now available for all major computer algebra systems (e.g. Paule and Schorn, 1995; Abramov *et al.*, 2004).

Not all sums encountered in practice have a hypergeometric term as summand, and there are several approaches to do summation on greater function classes. One approach, due to Zeilberger (1990), is to consider the class of (possibly multivariate) functions that can be defined via so-called *holonomic* systems of differential-difference equations. In this approach, the functions under consideration are represented by operators living in a certain non-commutative operator algebra that annihilate the given function. Algorithms based on Gröbner basis computation in this operator algebra are known for proving (Chyzak and Salvy, 1998) and finding (Chyzak, 1998, 2000) identities for holonomic functions. A univariate holonomic sequence satisfies a homogeneous linear difference equation with polynomial coefficients. In this special situation, holonomic sequences can be manipulated by classical methods using generating functions. Implementations of these algorithms are available (Salvy and Zimmermann, 1994; Mallinger, 1996).

Another generalization of the classical summation algorithms employs the notion of a *difference field.* In analogy to Risch's algorithm for symbolic integration (Risch, 1969, 1970; Bronstein, 1997), Karr (1981, 1985) has devised an algorithm that finds closed forms of expressions given in terms of nested sums and products. The sequences covered by Karr's algorithm are called $\Pi\Sigma$-sequences. Schneider (2001) has generalized Karr's algorithm in various directions. He has also got the only known implementation.

The case of higher order linear difference equations was considered, e.g., by Hendriks and Singer (1999). They introduce the notion of a *liouvillean sequence* in analogy to liouvillean functions known from the continuous case, and study the problem of solving linear difference equations in terms of these sequences. The mathematical foundation of this work is difference Galois theory (van der Put and Singer, 1997).

For the study of *nonlinear* difference equations from an algebraic viewpoint, Cohn (1965) developed the theory of *difference algebra,* which can be seen as an analogue of the theory of *differential algebra* (Ritt, 1950; Kaplansky, 1976). The method of *characteristic sets* provides computational tools for the treatment of nonlinear differential- and difference equation systems, which are based on differential and difference algebra, respectively. Characteristic set algorithms for the differential case depend heavily on the chain rule for differentiation. Conversely, due to the lack of a suitable chain rule for the difference operator, the development of characteristic set algorithms for the difference case has not been as successful.

* * *

Much more could be said about algorithms for difference equations, the sketch above is far from complete. But in the present thesis, the previous work on symbolic computation for difference equations plays hardly any role. Rather than extending and/or improving known algorithms, our main interest is in providing algorithms for sequences that have up to now been out of the scope of symbolic methods. Of course, it is pointless to think about algorithms applicable to *all* sequences. Any algorithm can only deal with objects which can be described in finite terms, and while there are uncountably many sequences already over $\{0,1\}$, only countably many objects can be described in finite terms. Therefore, every algorithm is applicable only to a certain class of sequences. We will define a very rich class of *admissible* sequences (Chapter 3), and develop algorithms for answering questions about sequences in this class.

The more restrictive a class of sequences is, the more powerful algorithms can be devised for it. Conversely, if a class of sequences is large, it can be difficult to find algorithms already for answering "simple" questions about the elements of that class. Our class of admissible sequences is quite large. It includes the class of $\Pi\Sigma$ sequences of Karr, which are defined by "nested" difference equations, as well as the univariate holonomic sequences, which are defined by higher order difference equations. In addition, also solutions of non-linear difference equations can be treated.

The algorithms described in the subsequent chapters are capable of solving problems involving admissible sequences that cannot be solved by any other algorithm presently known. The principal questions that we consider are the following.

Deciding zero equivalence. We present an algorithm which for a given admissible sequence $f(n)$ decides whether it is identically zero (Chapter 4). This algorithm can immediately be applied to proving identities for admissible sequences. It is also used as subalgorithm for a finder of algebraic dependencies (see below).

Proving inequalities. A procedure is presented by means of which inequalities for given admissible sequences can be proven automatically (Chapter 5). Though there is no sort of guarantee that the procedure will be successful on a particular identity at hand, we were able to verify a lot of inequalities appearing in the literature with our method. Our procedure is the only systematic symbolic treatment of special function inequalities that we know of.

Finding algebraic dependencies (Chapters 6 and 7). Identities for admissible sequences, which are of a special form, are algebraic dependencies. We present two methods for finding algebraic dependencies between some given admissible sequences. Using these methods, identities for admissible sequences can not only be proven, but even found algorithmically. In Chapter 7, we restrict the attention to a small subclass of admissible sequences, the class of C-finite sequences. We are able to give an algorithm that computes for a given choice of sequences from this class a description of *all* algebraic dependencies between them.

Summation of admissible sequences. We treat the problem of finding closed form representations of sums over admissible sequences (Chapter 8). The summation problem is the most important special case of finding algebraic dependencies in practice. Therefore it makes sense to investigate specialized algorithms for doing summation of admissible sequences. Both, indefinite and definite summation are considered.

The algorithms we present require only a moderate amount of mathematical background. Given some admissible sequences $f_1(n),\dots,f_m(n)$ over a field $\mathbb{K}$ as part of the input, our algorithms construct a difference ring R containing some elements $t_1,\dots,t_m$ which "correspond" to the sequences $f_i(n)$. Ideally, R will be isomorphic to a subring S of $\mathbb{K}^{\mathbb{N}}$ containing $f_1(n),\dots,f_m(n)$, and the correspondence means that t_i and $f_i(n)$ are mapped to each other via the isomorphism. In general, S will only be a homomorphic image of R, with $f_i(n)$ being the image of t_i. From a computational viewpoint, the ring R will be a polynomial ring, or a quotient ring of the form $\mathbb{K}[X]/\mathfrak{a}$ for some ideal $\mathfrak{a}$, and established algorithmic tools such as Gröbner basis techniques will be applied for doing computations in that ring. The results of these computations in R will then be interpreted as statements about the ring S, i.e., about the admissible sequences $f_1(n),\dots,f_m(n)$. Zimmermann (2005) is using a similar algebraic framework that served as a guide line for the construction we are using. One of the main advantages of our definition of admissibility is that we can always construct a suitable ring R where the questions listed above can be answered by means of Gröbner basis computations. In particular, we can avoid the use of characteristic set computations.

Though the class of admissible sequences includes several smaller classes that were defined for other algorithms, such "special purpose" algorithms are usually superior whenever they apply. Not only are these algorithms more efficient than ours, but also they are usually able to make sharper statements about the sequences to which they are applicable. For instance, Schneider's summation package Sigma (Schneider, 2001) is able not only to simplify a given sum, but it can also find out that a given sum *cannot* be simplified any further. Such negative results are out of the scope of our summation algorithms (Chapter 8).

Despite of this algorithmic shortcoming, we believe that the algorithms presented in this thesis are useful in daily work with special functions and combinatorial sequences. In order to support this claim, we have included more than one hundred examples, mostly taken from the literature. Some of these examples were chosen just to stress the wide applicability of our methods, while other—sometimes quite trivial—examples were selected for keeping more detailed illustrations as simple as possible.

All examples have been computed using a new software package named SumCracker, which we have implemented in the Mathematica system (Chapter 9). With this implementation, we want to underline that our algorithms are not only of theoretical interest, but that the proposed methods are actually feasible on current computer architectures. Parts of the implementation are rather prototypical and do perhaps still admit considerable improvement with respect to efficiency. Even so, we believe that the package is useful for practical work with special sequences that happen to be admissible.

Some Remarks on Notation

The use of mathematical notation is not always as precise as it seems to be. For example, the notation $\sum_{k=0}^{n} a_k$ can have at least three completely different meanings. Sometimes, the expression refers to the value that is obtained by adding $a_0, a_1, \dots, a_n$. Sometimes, however, the same notation refers to the sequence that maps every $n \in \mathbb{N}$ to the value $a_0 + a_1 + \cdots + a_n$. And sometimes, $\sum_{k=0}^{n} a_k$ is just understood as a "symbolic" sum, i.e., as element of some term algebra. We will follow general practice and usually assume that it is clear from the context what the particular meaning of an expression is. If space permits, we write $a_0 + a_1 + \cdots + a_n$ instead of $\sum_{k=0}^{n} a_k$ when n should be read as a particular, "known" integer rather than a symbolic variable, likewise for products.

Following not so general practice we allow ourselves to write $f(n)$ instead of f for a sequence $f\colon \mathbb{N} \to X$, though the notation $f(n)$ is also used for the value of f at the point $n \in \mathbb{N}$. The reason for introducing this additional ambiguity is that it is much more convenient—for both author and reader—to speak about, say, "the sequence $(-1)^n$" instead of "the sequence f defined via $f(n) := (-1)^n$ $(n \in \mathbb{N})$". The danger of confusion is comparatively small.

For further clarification about notational issues, see page 133f.

Acknowledgements

I would like to express my gratitude to my thesis advisor *Peter Paule,* the "center of energy" (Schorn, 1995), who, despite being permanently busy, is open to both mathematical and non-mathematical matters of his students at any time. His way of leadership makes our *combgang* a magnificent group to work in. I could also learn much from discussions and collaborations with the other members of this group, notably with *Stefan Gerhold, Carsten Schneider,* and *Burki Zimmermann.*

Also the discussions with *Tobias Beck* (on Lemma 3.14), *Gábor Bodnár* (on geometric reasoning in Section 4.3 and elsewhere), and *Ralf Hemmecke* (on Algorithm 4.2) were very helpful.

Finally, I want to thank *Josef Schicho* for agreeing to serve as second examiner of this thesis.

2 Preliminaries

This chapter contains a collection of well-known definitions and facts, which are needed later on. Its main purpose is to fix the notation and provide labels for certain general theorems. The advanced reader may securely skip this chapter and proceed directly to Chapter 3.

2.1 Polynomials, Ideals, and Gröbner Bases

It is sufficient for most of this text if the reader is familiar with the most basic facts about commutative algebra as they are presented, for instance, in the introductory book of Cox *et al.* (1992). Only occasionally, we will need material going beyond this text.

Throughout this text, we assume that $\mathbb{K}$ is a field of characteristic zero. If $\mathbb{K}$ appears in the description of an algorithm, it is also assumed that $\mathbb{K}$ is *computable,* by which we mean that each element of $\mathbb{K}$ has a finite representation (not necessarily unique), the field operations can be carried out algorithmically, and it is decidable whether a given representation of an element of $\mathbb{K}$ represents the zero element. Typically, $\mathbb{K}$ will be a rational function field over some finite algebraic extension of $\mathbb{Q}$ in examples. All rings in this thesis are commutative and have unit element 1.

Definition 2.1 Let $X = (x_1, \dots, x_n)$ be an n-tuple of indeterminates.

(1) A *term* (or *power product*) in X is an expression of the form $X^e := x_1^{e_1} x_2^{e_2} \cdots x_n^{e_n}$ for some exponent vector $e = (e_1, \dots, e_n) \in \mathbb{N}_0^n$.

The number $\deg X^e := e_1 + \cdots + e_n \in \mathbb{N}_0$ is called the *total degree* of the term X^e.

The set of all terms in X is denoted by $[X]$.

(2) A *polynomial* in X over $\mathbb{K}$ is a linear combination of terms with coefficients in $\mathbb{K}$, i.e., p is a polynomial if

$$p = a_1 X^{e_1} + \cdots + a_r X^{e_r}$$

for some $a_1, \dots, a_r \in \mathbb{K}$ and $e_1, \dots, e_r \in \mathbb{N}_0^n$.

The number $\deg p := \max_{i=1}^{r} \deg X^{e_i}$ is called the *total degree* of the polynomial p.

The set of all polynomials in X over $\mathbb{K}$ is denoted by $\mathbb{K}[X]$. $\mathbb{K}$ is the *coefficient field* (or *ground field*) of $\mathbb{K}[X]$.

The set $[X]$ of all terms forms a monoid under multiplication. Its unit element is $1 := x_1^0 \cdots x_n^0$. The ring $\mathbb{K}[X]$ is an integral domain, and its fraction field $Q(\mathbb{K}[X])$ is denoted by $\mathbb{K}(X)$. The elements of $\mathbb{K}(X)$ are called *rational functions* over $\mathbb{K}$ in X.

Definition 2.2 Let R be a ring. A set $\mathfrak{a} \subseteq R$ is called an *ideal* in R if $a_1 + a_2 \in \mathfrak{a}$ for all $a_1, a_2 \in \mathfrak{a}$ and $pa \in \mathfrak{a}$ for all $a \in \mathfrak{a}, p \in R$. We write $\mathfrak{a} \trianglelefteq R$ to denote that $\mathfrak{a}$ is an ideal in R.

For $p_1, \dots, p_r \in R$, we denote by $\langle p_1, \dots, p_r \rangle$ the smallest ideal containing $p_1, \dots, p_r$. If $\mathfrak{a} = \langle p_1, \dots, p_r \rangle$, we say that $\mathfrak{a}$ is *generated* by $p_1, \dots, p_r$ and that $p_1, \dots, p_r$ is a *basis* of $\mathfrak{a}$.

If $X = (x_1,\dots,x_n)$ and $Y = (y_1,\dots,y_m)$ are such that $\{x_1,\dots,x_n\} \subseteq \{y_1,\dots,y_m\}$, then there is a canonical embedding $\mathbb{K}[X] \hookrightarrow \mathbb{K}[Y]$. To facilitate language, we will regard $\mathbb{K}[X]$ as a subring of $\mathbb{K}[Y]$, $\mathbb{K}[X] \subseteq \mathbb{K}[Y]$, in such cases. For nontrivial $S \subseteq \mathbb{K}[X]$, observe that the ideal generated by S in $\mathbb{K}[X]$ is different from the ideal generated in $\mathbb{K}[Y]$, the notation $\langle S \rangle$ is hence ambiguous. We hope that the careful reader will always be able to detect from the context which ideal generation is meant, and prefer to abstain from introducing more fancy notation.

If $\mathfrak{a} \trianglelefteq \mathbb{K}[X]$ is an ideal, then we denote by $V(\mathfrak{a}) \subseteq \mathbb{K}^n$ the set of all points $(x_1,\dots,x_n) \in \mathbb{K}^n$ such that $p(x_1,\dots,x_n) = 0$ for all $p \in \mathfrak{a}$. This set is called the *variety* of $\mathfrak{a}$. Conversely, a set $A \subseteq \mathbb{K}^n$ is called an *algebraic set* (or a *variety*) if $A = V(\mathfrak{a})$ for some ideal $\mathfrak{a} \trianglelefteq \mathbb{K}[X]$. For every set $A \subseteq \mathbb{K}^n$, we have the *vanishing ideal*

$$I(A) := \{\, p \in \mathbb{K}[X] : \forall\ (x_1,\dots,x_n) \in A : p(x_1,\dots,x_n) = 0 \,\}.$$

The algebraic set $V(I(A))$ is called the *Zariski closure* of A.

Definition 2.3 Let $\mathfrak{a},\mathfrak{b} \trianglelefteq \mathbb{K}[X]$.

(1) $\mathfrak{a}+\mathfrak{b} := \{\, a+b : a \in \mathfrak{a}, b \in \mathfrak{b} \,\}$ is called the *sum* of $\mathfrak{a}$ and $\mathfrak{b}$ and $\mathfrak{a}\cdot\mathfrak{b} := \langle\, a\cdot b : a \in \mathfrak{a}, b \in \mathfrak{b} \,\rangle$ is called the *product* of $\mathfrak{a}$ and $\mathfrak{b}$.

(2) $\mathfrak{a}$ is called a *radical ideal* if $p^m \in \mathfrak{a}$ implies $p \in \mathfrak{a}$ for all $p \in \mathbb{K}[X]$, $m \in \mathbb{N}$.

(3) $\mathrm{Rad}(\mathfrak{a}) := \{\, p \in \mathbb{K}[X] : \exists\ m \in \mathbb{N} : p^m \in \mathfrak{a} \,\}$ is called the *radical ideal* of $\mathfrak{a}$.

(4) $\mathfrak{a}$ is called *prime ideal* if, for all $p,q \in \mathbb{K}[X]$, $pq \in \mathfrak{a}$ implies $p \in \mathfrak{a}$ or $q \in \mathfrak{a}$.

(5) $\mathfrak{a}$ is called *primary ideal* if, for all $p,q \in \mathbb{K}[X]$, $pq \in \mathfrak{a}$ and $p \notin \mathfrak{a}$ implies $q \in \mathrm{Rad}\,\mathfrak{a}$.

Theorem 2.4 Let $\mathfrak{a},\mathfrak{b} \trianglelefteq \mathbb{K}[X]$.

(1) $\mathfrak{a}+\mathfrak{b}$ is an ideal of $\mathbb{K}[X]$. In particular, $\langle a_1,\dots,a_m\rangle + \langle b_1,\dots,b_\ell\rangle = \langle a_1,\dots,a_m,b_1,\dots,b_\ell\rangle$. For the product, we have

$$\langle a_1,\dots,a_m\rangle \cdot \langle b_1,\dots,b_\ell\rangle = \langle\, a_ib_j : i = 1,\dots,m,\ j = 1,\dots,\ell \,\rangle$$

(2) $\mathfrak{a}\cap\mathfrak{b}$ is an ideal of $\mathbb{K}[X]$, and $\mathfrak{a}\cdot\mathfrak{b} \subseteq \mathfrak{a}\cap\mathfrak{b}$.

(3) $\mathfrak{a}$ is prime if and only if $\mathbb{K}[X]/\mathfrak{a}$ is an integral domain.

(4) $\mathfrak{a}$ is primary if and only if $\mathrm{Rad}\,\mathfrak{a}$ is prime.

(5) There exist primary ideals $\mathfrak{p}_1,\dots,\mathfrak{p}_s \trianglelefteq \mathbb{K}[X]$ such that $\mathfrak{a} = \mathfrak{p}_1\cap\mathfrak{p}_2\cap\dots\cap\mathfrak{p}_s$.

A representation as in (5) is called a *primary decomposition* of $\mathfrak{a}$. The prime ideals $\mathrm{Rad}\,\mathfrak{p}_1$, ..., $\mathrm{Rad}\,\mathfrak{p}_s$ are called the *associated prime ideals* of the decomposition. If the primary decomposition is minimal in the sense that $\mathfrak{p}_i \not\supseteq \bigcap_{j\neq i}\mathfrak{p}_j$ and $\mathrm{Rad}\,\mathfrak{p}_i \neq \mathrm{Rad}\,\mathfrak{p}_j$ $(i \neq j)$, then the associated prime ideals are uniquely determined. These are called the associated prime ideals of $\mathfrak{a}$.

Given $\mathfrak{a} \trianglelefteq \mathbb{K}[X]$, there exists a longest descending chain of prime ideals $\mathfrak{a} \supsetneq \mathfrak{p}_1 \supsetneq \mathfrak{p}_2 \supsetneq \cdots$. Its length is called the *dimension* of $\mathfrak{a}$ and denoted by $\dim\mathfrak{a}$. If $\mathfrak{a}$ is prime, then $\dim\mathfrak{a}$ is the transcendence degree of $Q(\mathbb{K}[X]/\mathfrak{a})$ over $\mathbb{K}$ (Eisenbud, 1995). Observe that this field is well defined, by part (3) of the Theorem. If $\mathfrak{a}$ is any ideal, then $\dim\mathfrak{a} = \max_i \dim\mathfrak{p}_i$ where $\mathfrak{p}_i$ are the associated prime ideals of $\mathfrak{a}$.

Hilbert's basis theorem asserts that every ideal in $\mathbb{K}[X]$ has a finite basis. A *Gröbner basis* (Buchberger, 1965) of an ideal $\mathfrak{a} \trianglelefteq \mathbb{K}[X]$ is a basis of $\mathfrak{a}$ with special properties that make it possible to answer a lot of questions about $\mathfrak{a}$ algorithmically.

Definition 2.5

(1) A strict linear order $\prec$ on X is *admissible* (or *term order*) if

$$1 \prec t \text{ for all } t \in [X] \setminus \{1\}, \quad \text{and} \quad u \prec v \Longrightarrow ut \prec vt \text{ for all } t,u,v \in [X].$$

(2) Fix a term order $\prec$ and let $p \in \mathbb{K}[X]$ be of the form

$$p = a_1 X^{e_1} + a_2 X^{e_2} + \cdots + a_r X^{e_r}$$

with $X^{e_1} \prec X^{e_2} \prec \cdots \prec X^{e_r}$ and $a_i \in \mathbb{K} \setminus \{0\}$. Then $\mathrm{LT}(p) := X^{e_r}$, $\mathrm{LC}(p) := a_r$, $\mathrm{LM}(p) := a_r X^{e_r}$ are called the *leading term, leading coefficient,* and *leading monomial,* respectively.

If $\mathrm{LC}(p) = 1$ then p is called *monic*.

(3) For a set $S \subseteq \mathbb{K}[X]$, let $\mathrm{LT}(S) := \{\mathrm{LT}(p) : p \in S\}$. A finite set $G \subseteq \mathbb{K}[X] \setminus \{0\}$ is called a *Gröbner basis* w.r.t. $\prec$ if

$$\mathrm{LT}(\langle G \rangle) = \langle \mathrm{LT}(G) \rangle.$$

If we just say that G is a Gröbner basis and $\prec$ is not clear from the context, then we mean that G is a Gröbner basis w.r.t. some term ordering $\prec$.

(4) A polynomial $p \in \mathbb{K}[X]$ is called *reduced* (or *irreducible*) with respect to a set $G \subseteq \mathbb{K}[X]$ if no term of p is a multiple of the leading term $\mathrm{LT}(g)$ of some $g \in G$. Otherwise it is called *reducible*. (The notions "reducible" and "irreducible" are in particular applied in the special case where p is a term.)

Theorem 2.6 Let $\prec$ be a term order on $[X]$, and $\mathfrak{a} \trianglelefteq \mathbb{K}[X]$.

(1) There exists a Gröbner basis G with $\mathfrak{a} = \langle G \rangle$. Given any basis $p_1, \ldots, p_r$ of $\mathfrak{a}$, there is an algorithm that computes a Gröbner basis G of $\mathfrak{a}$ from $p_1, \ldots, p_r$.

(2) There is exactly one Gröbner basis G of $\mathfrak{a}$ whose elements are monic and each $g \in G$ is reduced w.r.t. $G \setminus \{g\}$. This basis is called *the* Gröbner basis of $\mathfrak{a}$.

(3) If G is a Gröbner basis of $\mathfrak{a}$, then the set of all $p \in \mathbb{K}[X]$ which are reduced w.r.t. G forms a complete set of representatives of $\mathbb{K}[X]/\mathfrak{a}$. Given $p \in \mathbb{K}[X]$ the representative of $p + \mathfrak{a}$ which is reduced w.r.t. G is called the *normal form* (or *reductum*) of p w.r.t. G, and is denoted by $\bar{p}$.

(4) If G is a Gröbner basis of $\mathfrak{a}$, then the classes of $[X] \setminus \mathrm{LT}(\langle G \rangle)$ is a $\mathbb{K}$-vector space basis of $\mathbb{K}[X]/\mathfrak{a}$.

(5) The normal form of $p \in \mathbb{K}[X]$ w.r.t. a Gröbner basis G is 0 if and only if $p \in \langle G \rangle$.

(6) If $\mathfrak{a}$ contains an element of $\mathbb{K} \setminus \{0\}$, then the Gröbner basis of $\mathfrak{a}$ is $\{1\}$.

Proofs of the above statements can be found in every textbook on Gröbner bases. We list some standard applications of Gröbner bases, and refer again to the literature for proofs and further details.

Theorem 2.7 Let $\mathfrak{a} = \langle p_1, \ldots, p_m \rangle \in \mathbb{K}[X] = \mathbb{K}[x_1, \ldots, x_n]$ be given.

(1) For a given $p \in \mathbb{K}[X]$, the normal form $\bar{p}$ of p w.r.t. a Gröbner basis of $\mathfrak{a}$ can be computed. In particular, it can be decided whether $p \in \mathfrak{a}$ (Ideal membership).

(2) (Radical membership) For a given $p \in \mathbb{K}[X]$, it can be decided whether $p \in \operatorname{Rad}\mathfrak{a}$.

By Hilbert's Nullstellensatz, this is equivalent to: given $p_1, \ldots, p_r$ and $p \in \mathbb{K}[X]$ it is decidable whether

$$\forall\, x_1, \ldots, x_n \in \bar{\mathbb{K}} : \big(p_1(x_1, \ldots, x_n) = \cdots = p_r(x_1, \ldots, x_n) = 0 \Longrightarrow p(x_1, \ldots, x_n) = 0\big).$$

Here $\bar{\mathbb{K}}$ denotes the algebraic closure of $\mathbb{K}$.

We have $p \in \operatorname{Rad}\mathfrak{a}$ if and only if $\mathfrak{a} + \langle yp - 1 \rangle \trianglelefteq \mathbb{K}[X, y]$ is the trivial ideal $\langle 1 \rangle$.

(3) (Elimination) An ideal basis of the *elimination ideals*

$$\mathfrak{a}_i := \mathfrak{a} \cap \mathbb{K}[x_1, \ldots, x_i] \qquad (i = 1, \ldots, n)$$

can be computed.

If R is a ring, then R^d has a canonical R-module structure. Its generators are denoted by $\varepsilon_1, \ldots, \varepsilon_d$, i.e., $\varepsilon_1 = (1, 0, \ldots, 0)$, etc. For $p_1, \ldots, p_r \in R^d$, we write $[p_1, \ldots, p_r]$ for the smallest submodule of R^d containing $p_1, \ldots, p_r$. If $M = [p_1, \ldots, p_r]$, we say that M is *generated* by $p_1, \ldots, p_r$ and that $p_1, \ldots, p_r$ is a *basis* of M.

The notion of Gröbner basis can be generalized from polynomial ideals to finitely generated submodules of $\mathbb{K}[X]^d$. The book of Kreuzer and Robbiano (2000) gives a careful description of the theory of Gröbner basis, which includes this case in full generality.

A *term* in $\mathbb{K}[X]^d$ is an expression of the form $x_1^{e_1} x_2^{e_2} \cdots x_n^{e_n} \varepsilon_i$ for $e_1, \ldots, e_n \in \mathbb{N}_0$ and $i \in \{1, \ldots, d\}$. With a suitable adaption of the notion of admissible ordering, parts (2)–(4) of Def. 2.5 and Theorem 2.6.(1)–(5) and Theorem 2.7.(1) and (3) apply analogously to submodules $M \subseteq \mathbb{K}[X]^d$ in place of ideals $\mathfrak{a} \trianglelefteq \mathbb{K}[X]$. We refer to Kreuzer and Robbiano's book for details.

Let $\mathfrak{a} = \langle p_1, \ldots, p_r \rangle \trianglelefteq \mathbb{K}[X]$ and $p \in \mathbb{K}[X]$. Then $p \in \mathfrak{a}$ if and only if

$$p = q_1 p_1 + q_2 p_2 + \cdots + q_r p_r$$

for some $q_1, \ldots, q_r \in \mathbb{K}[X]$. These polynomials are called *cofactors* of p w.r.t. $p_1, \ldots, p_r$. The cofactors need not be uniquely determined. If $q_1', \ldots, q_r' \in \mathbb{K}[X]$ form a different representation of p, then

$$(q_1 - q_1')p_1 + (q_2 - q_2')p_2 + \cdots + (q_r - q_r')p_r = 0.$$

This motivates the definition of a *syzygy.*

Definition 2.8 Let R be a ring and $p_1, \ldots, p_r \in R$. A tuple $(q_1, \ldots, q_r) \in R^r$ is called a *syzygy* for $p_1, \ldots, p_r$ iff

$$q_1 p_1 + q_2 p_2 + \cdots + q_r p_r = 0.$$

We write

$$\operatorname{Syz}(p_1, \ldots, p_r) := \{\, (q_1, \ldots, q_r) \in R^r : q_1 p_1 + \cdots + q_r p_r = 0 \,\} \subseteq R^r$$

for the set of all syzygies for $p_1, \ldots, p_r$.

It is easy to see that the set of syzygies forms a submodule of R^r. In the case $R = \mathbb{K}[X]$, it is a standard application of Gröbner bases to compute a basis of the syzygy module (Becker *et al.*, 1993, Section 6.1). This algorithm can easily be extended to the case $R = \mathbb{K}[X]/\mathfrak{a}$. As this is usually not included in textbooks, we carry out the details.

Theorem 2.9 (Syzygy Computation) Let $\mathfrak{a} = \langle p_1, \dots, p_m \rangle \trianglelefteq \mathbb{K}[X]$ and $q_1, \dots, q_r \in \mathbb{K}[X]/\mathfrak{a}$. For $q \in \mathbb{K}[X]/\mathfrak{a}$, we use the notation $\bar{q} \in \mathbb{K}[X]$ to denote a representative of the equivalence class q. Then

$$\begin{aligned} & (s_1, \dots, s_r) \in \operatorname{Syz}(q_1, \dots, q_r) \subseteq (\mathbb{K}[X]/\mathfrak{a})^r \\ \iff \quad & \exists\, c_1, \dots, c_m \in \mathbb{K}[X] : (\bar{s}_1, \dots, \bar{s}_r, c_1, \dots, c_m) \in \operatorname{Syz}(\bar{q}_1, \dots, \bar{q}_r, p_1, \dots, p_m) \subseteq \mathbb{K}[X]^{r+m}. \end{aligned}$$

Hence a basis of $\operatorname{Syz}(q_1, \dots, q_r)$ can be obtained from a basis of $\operatorname{Syz}(\bar{q}_1, \dots, \bar{q}_r, p_1, \dots, p_m)$ by discarding the last m coordinates.

Proof "$\Rightarrow$" If $(s_1, \dots, s_r) \in \operatorname{Syz}(q_1, \dots, q_r)$ then

$$\begin{aligned} & s_1 q_1 + \cdots + s_r q_r = 0 \\ \Longrightarrow \quad & \bar{s}_1 \bar{q}_1 + \cdots + \bar{s}_r \bar{q}_r \in \mathfrak{a} \\ \Longrightarrow \quad & \bar{s}_1 \bar{q}_1 + \cdots + \bar{s}_r \bar{q}_r = (-c_1) p_1 + \cdots + (-c_m) p_m \end{aligned}$$

for some $c_1, \dots, c_m \in \mathbb{K}[X]$, and hence

$$(\bar{s}_1, \dots, \bar{s}_r, c_1, \dots, c_m) \in \operatorname{Syz}(\bar{q}_1, \dots, \bar{q}_r, p_1, \dots, p_m).$$

"$\Leftarrow$" If $(\bar{s}_1, \dots, \bar{s}_r, c_1, \dots, c_m) \in \operatorname{Syz}(\bar{q}_1, \dots, \bar{q}_r, p_1, \dots, p_m)$, then

$$\begin{aligned} & \bar{s}_1 \bar{q}_1 + \cdots + \bar{s}_r \bar{q}_r = (-c_1) p_1 + \cdots + (-c_m) p_m \\ \Longrightarrow \quad & \bar{s}_1 \bar{q}_1 + \cdots + \bar{s}_r \bar{q}_r \in \mathfrak{a} \\ \Longrightarrow \quad & s_1 q_1 + \cdots + s_r q_r = 0 \\ \Longrightarrow \quad & (s_1, \dots, s_r) \in \operatorname{Syz}(q_1, \dots, q_r). \quad \blacksquare \end{aligned}$$

Let M be a submodule of $\mathbb{K}[X]^d$. For an ideal $\mathfrak{a} \trianglelefteq \mathbb{K}[X]$, define the submodule $A := \mathfrak{a}\varepsilon_1 + \cdots + \mathfrak{a}\varepsilon_d$ of $\mathbb{K}[X]^d$. Using the isomorphisms

$$(\mathbb{K}[X]/\mathfrak{a})^d \cong \mathbb{K}[X]^d/A \quad \text{and} \quad (\mathbb{K}[X]^d/A)/(M + A/A) \cong \mathbb{K}[X]^d/(M + A)$$

and Gröbner basis for submodules of $\mathbb{K}[X]^d$, we can obtain unique normal forms of $(\mathbb{K}[X]/\mathfrak{a})^d$ modulo a given submodules of it.

2.2 Sequences, Difference Equations, and Recurrences

A *sequence* in $\mathbb{K}$ is a function $f \colon \mathbb{N} \to \mathbb{K}$. A *multivariate sequence* (m-variate sequence) in $\mathbb{K}$ is a function $f \colon \mathbb{N}^m \to \mathbb{K}$. We allow ourselves to use the symbolism $f(n)$ for both the value of f at the point n and for the whole sequence f itself. The context may clarify whether the whole sequence or a particular value of it is meant. In examples, sequences may have starting points different from 1.

The set of all sequences over $\mathbb{K}$ is denoted by $\mathbb{K}^{\mathbb{N}}$. Together with pointwise addition and multiplication, $\mathbb{K}^{\mathbb{N}}$ forms a ring.

A *difference equation* is an equation of the form

$$F(f(n), f(n+1), \dots, f(n+r), n) = 0 \qquad (n \geq 1), \tag{2.1}$$

for some function $F\colon \mathbb{K}^{r+1} \to \mathbb{K}$. Difference equations form a discrete analogue of *differential equations* (replace n by a continuous variable x and $f(n+i)$ by the ith derivative $f^{(i)}(x)$). For many classical results about differential equations there are analogous results for difference equations (Milne-Thomson, 1933). The number r in (2.1) is called the *order* of the difference equation, and a sequence $f(n)$ that makes (2.1) valid is called a *solution* of the difference equation.

A special type of difference equations are recurrences. A *recurrence* for a sequence $f(n)$ is an equation of the form

$$f(n+r) = R(f(n), f(n+1), \dots, f(n+r-1), n) \qquad (n \geq 1)$$

for some function $R\colon \mathbb{K}^{r+1} \to \mathbb{K}$. The number $r \in \mathbb{N}$ is called the *order* of the recurrence.

Recurrences may be used for defining of sequences. For, if $f(n)$ satisfies a recurrence of order r, then $f(n)$ is uniquely determined up to the choice of r initial values $f(0), f(1), \dots, f(r-1)$. In order to define a sequence, it is not necessary that the function R on the right hand side of a recurrence be defined for every point $x \in \mathbb{K}^{r+1}$. For example,

$$R\colon \mathbb{K}^3 \to \mathbb{K}, \qquad R(x,y,z) := \frac{x+y}{z+1}$$

is not defined for $z = -1$, nevertheless

$$f(n+2) = R(f(n), f(n+1), n) \quad (n \geq 1), \qquad f(1) = \alpha, f(2) = \beta$$

well defines a sequence $f(n)$, because the singularity $z = -1$ is never encountered. We generalize the notion of a recurrence accordingly: An equation

$$f(n+r) = R(f(n), f(n+1), \dots, f(n+r-1), n) \quad (n \geq 1)$$

is called a recurrence if $R\colon D \to \mathbb{K}$ for some $D \subseteq \mathbb{K}^{r+1}$. A sequence $f(n)$ in $\mathbb{K}$ is a solution of this recurrence if

$$(f(n), f(n+1), \dots, f(n+r-1), n) \in D \text{ for all } n \in \mathbb{N}.$$

We define some special types of recurrences that are of particular importance.

Definition 2.10

(1) A *linear recurrence* is a recurrence of the form

$$f(n+r) = a_0(n)f(n) + a_1(n)f(n+1) + \cdots + a_{r-1}(n)f(n+r-1) + a(n) \quad (n \geq 1). \tag{2.2}$$

The recurrence is called *homogeneous* if $a(n) = 0$ and *inhomogeneous* otherwise. A recurrence is called *nonlinear* if it is not linear.

(2) A homogeneous linear recurrence (2.2) is called *P-finite* (or *holonomic*) if $a_1(n),\dots,a_{r-1}(n)$ are rational functions in n. A P-finite recurrence of order 1 is called *hypergeometric*.

(3) A homogeneous linear recurrence (2.2) is called *C-finite* if $a_1(n),\dots,a_{r-1}(n)$ are constant, i.e., independent of n.

(4) A sequence is called P-finite (holonomic, hypergeometric, C-finite, ...) if it satisfies a P-finite (holonomic, hypergeometric, C-finite, ...) recurrence.

Example 2.11

(1) The following well-known sequences are all C-finite.

$$\begin{aligned}
F_{n+2} &= F_{n+1}+F_n, & F_0 &= 0, F_1 = 1 & &\text{(Fibonacci sequence)}\\
L_{n+2} &= L_{n+1}+L_n, & L_0 &= 2, L_1 = 1 & &\text{(Lucas sequence)}\\
P_{n+2} &= 2P_{n+1}+P_n, & P_0 &= 0, P_1 = 1 & &\text{(Pell sequence)}\\
Q_{n+2} &= 2Q_{n+1}+Q_n, & Q_0 &= 2, Q_1 = 2 & &\text{(Pell-Lucas sequence)}
\end{aligned}$$

(2) Most classical families of *orthogonal polynomials* satisfy P-finite recurrences. For instance,

$$P_{n+2}(x) = \frac{2n+3}{n+2}xP_{n+1}(x) - \frac{n+1}{n+2}P_n(x) \quad (n \geq 0), \qquad P_0(x) = 1, P_1(x) = x$$

defines the Legendre polynomials (Abramowitz and Stegun, 1972). This recurrence is P-finite, but not C-finite.

More generally, for $\alpha,\beta > -1$, the sequence of *Jacobi polynomial*s is defined via

$$\begin{aligned}
&P_{n+2}^{(\alpha,\beta)}(x) = AP_{n+1}^{(\alpha,\beta)}(x) + BP_n^{(\alpha,\beta)}(x) \quad (n \geq 0),\\
&P_0^{(\alpha,\beta)}(x) = 1, \quad P_1^{(\alpha,\beta)}(x) = \tfrac{1}{2}(2+\alpha+\beta)x + \tfrac{1}{2}(\alpha-\beta),
\end{aligned}$$

where

$$\begin{aligned}
A &= \frac{(2n+3+\alpha+\beta)((\alpha^2-\beta^2)+(2n+\alpha+\beta)(2n+4+\alpha+\beta)x)}{2(n+2)(n+2+\alpha+\beta)(2n+2+\alpha+\beta)}\\
B &= \frac{(n+1+\alpha)(n+\alpha+\beta)(2n+4+\alpha+\beta)}{(n+2)(n+2+\alpha+\beta)(2n+2+\alpha+\beta)}
\end{aligned}$$

This is again a P-finite recurrence (using, e.g., $\mathbb{K} = \mathbb{Q}(x,\alpha,\beta)$ as ground field).

The Jacobi polynomials reduce to the Legendre polynomials for $(\alpha,\beta) = (0,0)$.

Other parameterized families of orthogonal polynomials are the Gegenbauer polynomials $C_n^m(x)$, defined via

$$C_{n+2}^m(x) = 2x\frac{m+n+1}{n+2}C_{n+1}^m(x) - \frac{2m+n}{n+2}C_n^m(x) \quad (n \geq 0), \qquad C_0^m(x) = 1, C_1^m(x) = 2mx,$$

and the Laguerre polynomials $L_n^\alpha(x)$, defined via

$$L_{n+2}^\alpha(x) = \frac{2n+3+\alpha-x}{n+2}L_{n+1}^\alpha(x) - \frac{n+\alpha+1}{n+2}L_n^\alpha(x) \ (n \geq 0), \quad L_0^\alpha(x) = 1, L_1^\alpha(x) = 1+\alpha-x.$$

(3) Let $a(n)$ and $b(n)$ be two sequences in $\mathbb{K} \setminus \{0\}$ and consider the sequence

$$c(n) := \mathop{\mathrm{K}}_{k=0}^{n}(b(k)/a(k)) := a(0) + \cfrac{b(1)}{a(1) + \cfrac{b(2)}{a(2) + \cfrac{b(3)}{\cdots + \cfrac{b(n)}{a(n)}}}}$$

of partial continued fractions. The *continuants* $p(n)$ and $q(n)$ are defined via the linear recurrences

$$\begin{aligned} p(n+2) &= a(n+2)p(n+1) + b(n+2)p(n) \quad (n \geq -1) \qquad p(-1) = 1,\, p(0) = a(0), \\ q(n+2) &= a(n+2)q(n+1) + b(n+2)q(n) \quad (n \geq -1) \qquad q(-1) = 0,\, q(0) = 1, \end{aligned}$$

and a fundamental theorem in the theory of continued fractions states that $c(n) = p(n)/q(n)$ for all $n \in \mathbb{N}$ (Perron, 1929).

For instance, we have the classical result

$$\mathrm{e} = 1 + \mathop{\mathrm{K}}_{n=0}^{\infty}(1/a(n))$$

with $a(n) = 1, 1, 2, 1, 1, 4, 1, 1, 6, 1, 1, 8, \ldots$. Using the gfun package (Salvy and Zimmermann, 1994) or the package of Mallinger (1996), it is easy to find a P-finite recurrence for $a(n)$:

$$\begin{aligned} a(n+3) = \frac{(n+2)(4n^2+28n+21)}{(n+1)(4n^2+20n-3)}a(n) - \frac{9(2n+3)}{(n+1)(4n^2+20n-3)}a(n+1) \\ - \frac{9(2n+1)}{(n+1)(4n^2+20n-3)}a(n+2) \qquad (n \geq 0). \end{aligned}$$

However, $a(n)$ is not a rational function and so the recurrences defining the continuants associated to this continued fraction are not P-finite.

(4) The doubly exponential sequence $f(n) = 2^{2^n}$ satisfies the simple nonlinear recurrence

$$f(n+1) = f(n)^2 \quad (n \geq 0), \qquad f(0) = 2.$$

(5) Golomb (1963) considers sequences $f(n)$ satisfying the equation

$$f(n+1) = r + \prod_{k=1}^{n} f(k) \qquad (n \geq 0),$$

with $r \in \mathbb{N}$ fixed. It is easily verified that these sequences satisfy the nonlinear recurrence

$$f(n+1) = (f(n)-1)(f(n)-(r-1)) + 1 \qquad (n \geq 1).$$

Aho and Sloane (1973) give solutions of these and similar nonlinear recurrences in terms of doubly exponential sequences.

(6) The class of *Somos sequences* (Somos, 1989; Gale, 1991) provides another source of non-linear recurrences. A sequence C_n is called Somos-r ($r \in \mathbb{N}$ fixed) if it satisfies a recurrence of the form

$$C_{n+2r} = \frac{1}{C_n}\big(a_1 C_{n+2r-1} C_{n+1} + a_2 C_{n+2r-2} C_{n+2} + \cdots a_r C_{n+2r-r} C_{n+r}\big) \qquad (n \geq 0)$$

where $a_1, \ldots, a_r$ are fixed. These sequences arise in the manipulation of trigonometric functions.

It is of interest to know whether a given Somos-r-sequences is also Somos-r' for some given $r' \neq r$ (van der Poorten, 2004).

(7) The sequence of *Bernoulli numbers* B_n may be defined by

$$B_{n+1} = -\sum_{k=0}^{n} \binom{n+1}{k} B_k \quad (n \geq 0), \qquad B_0 = 1.$$

This equation is not understood as a recurrence at all in our considerations.

A set S of equations of the form

$$\begin{aligned} S = \big\{\, & F_1\big(f_1(n), \ldots, f_1(n+r_{1,1}), f_2(n), \ldots, f_2(n+r_{1,2}), \ldots\ldots f_m(n), \ldots, f_m(n+r_{1,m})\big) = 0, \\ & F_2\big(f_1(n), \ldots, f_1(n+r_{2,1}), f_2(n), \ldots, f_2(n+r_{2,2}), \ldots\ldots f_m(n), \ldots, f_m(n+r_{2,m})\big) = 0, \\ & \quad \vdots \\ & F_\ell\big(f_1(n), \ldots, f_1(n+r_{\ell,1}), f_2(n), \ldots, f_2(n+r_{\ell,2}), \ldots\ldots f_m(n), \ldots, f_m(n+r_{\ell,m})\big) = 0 \,\big\} \end{aligned}$$

for functions $F_1, \ldots, F_m \colon \mathbb{K}^{m(r+1)} \to \mathbb{K}$ is called a *system of difference equations.* The number $r := \max_{i=1}^{\ell} \max_{j=1}^{m} r_{i,j}$ is called the *order* of the system S, and a tuple $f_1(n), \ldots, f_m(n)$ of sequences in $\mathbb{K}$ is said to be a *solution* if it makes every equation in S valid. A *system of recurrences* is a system of difference equations each of which is a recurrence.

2.3 Difference Rings, Difference Homomorphisms, and Difference Ideals

The theory of *difference algebra* (Cohn, 1965) aims at providing an algebraic formalization of difference equations and sequences. It can be seen as a "discrete analog" of the more widely known field of *differential algebra* (Ritt, 1950; Kaplansky, 1976), which provides an algebraic viewpoint to differential equations. We collect the basic definitions and results of difference algebra.

Definition 2.12 Let R be a ring and $s\colon R \to R$ be a homomorphism. The pair (R, s) is called a *difference ring.* R is called the *underlying ring* and s is called the *shift* of the difference ring (R, s). A difference ring whose underlying ring is a field is called a *difference field.*

We will simply write R instead of (R, s) when s is clear from the context or arbitrary.

The definition of s is extended to subsets $M \subseteq R$ via $s(M) := \{\, s(m) : m \in M \,\}$.

Example 2.13

(1) Any ring R together with the identity $s\colon R \to R$, $r \mapsto r$ ($r \in R$) is a difference ring.

(2) The polynomial ring $\mathbb{K}[t]$ becomes a difference ring by specifying $s(x) := x$ ($x \in \mathbb{K}$) and $s(t) := t+1$. The rational function field $\mathbb{K}(t)$, with s defined analogously, is a difference field.

(3) The ring $\mathbb{K}^{\mathbb{N}}$ of all sequences has a natural difference ring structure. If $\mathbf{E}\colon \mathbb{K}^{\mathbb{N}} \to \mathbb{K}^{\mathbb{N}}$ is defined via

$$(f(1), f(2), f(3), \dots) \xmapsto{\mathbf{E}} (f(2), f(3), f(4), \dots) \qquad (f(n) \in \mathbb{K}^{\mathbb{N}})$$

then $(\mathbb{K}^{\mathbb{N}}, \mathbf{E})$ is a difference ring.

Unless otherwise stated, $\mathbb{K}^{\mathbb{N}}$ is always assumed to be equipped with this shift.

(4) Let $m \in \mathbb{N}$ be fixed and consider the ring

$$\begin{aligned} R := \mathbb{K}[&t_{1,0},\ t_{1,1},\ t_{1,2}, \dots\dots \\ &t_{2,0},\ t_{2,1},\ t_{2,2}, \dots\dots \\ &\vdots \\ &t_{m,0},\ t_{m,1},\ t_{m,2}, \dots\dots] \end{aligned}$$

where the $t_{i,j}$ are indeterminates, i.e., they form an algebraically independent set over $\mathbb{K}$. R may be understood as a ring of polynomials with infinitely many variables.

Let $s\colon R \to R$ be defined via $s(x) = x$ ($x \in \mathbb{K}$) and

$$s(t_{i,j}) := t_{i,j+1} \qquad (i = 1, \dots, m,\ j \geq 0).$$

The difference ring (R, s) is denoted by $\mathbb{K}\{t_1, \dots, t_m\}$. $\mathbb{K}$ is called its *ground field*. The $t_1, \dots, t_m$ are called the *difference variables,* and $\mathbb{K}\{t_1, \dots, t_m\}$ is called the *free difference ring* in m (difference) variables. Its quotient field is denoted by $\mathbb{K}\langle t_1, \dots, t_m\rangle$ and called the *free difference field* in m (difference) variables.

The elements of $\mathbb{K}\{t_1, \dots, t_m\}$ are called *difference polynomials.* Usually, we refer to $t_{i,j}$ using the notation $s^j t_i$ (writing t_i instead of $s^0 t_i$ for $t_{i,0}$).

Definition 2.14 Let $R = (R, s)$ be a difference ring.

(1) An element $x \in R$ is called a *constant,* if $s(x) = x$. The set of all constants in R is denoted by $\operatorname{const} R$.

(2) An ideal $\mathfrak{a} \trianglelefteq R$ is called a *difference ideal* of R if

$$\forall\, x \in R : x \in \mathfrak{a} \iff s(x) \in \mathfrak{a}.$$

For a set $S \subseteq R$, we write $\langle S\rangle$ for the smallest ideal of R that contains S, and $\langle\langle S\rangle\rangle$ for the smallest difference ideal of R containing S.

Of course we write $\langle p_1, \dots, p_r\rangle$ and $\langle\langle p_1, \dots, p_r\rangle\rangle$, respectively, instead of $\langle\{p_1, \dots, p_r\}\rangle$ and $\langle\langle\{p_1, \dots, p_r\}\rangle\rangle$.

(3) Let (R', s') be another difference ring. A ring homomorphism $\varphi\colon R \to R'$ is called a *difference homomorphism* if
$$\forall\, x \in R : \varphi(s(x)) = s'(\varphi(x)).$$

(4) R is *isomorphic* to a difference ring R', written $R \cong R'$, if there exists a bijective difference homomorphism $\varphi\colon R \to R'$.

Cohn (1965) distinguishes between difference ideals and reflexive difference ideals. In his definition, $\mathfrak{a}$ is a difference ideal if $p \in \mathfrak{a}$ implies $s(p) \in \mathfrak{a}$, and a reflexive difference ideal if, in addition, $s(p) \in \mathfrak{a}$ implies $p \in \mathfrak{a}$. In our considerations, all difference ideals will be reflexive, and hence we can securely drop this attribute. The definition above takes this into account.

Theorem 2.15 Let (R_1, s_1) and (R_2, s_2) be difference rings and $\varphi\colon R_1 \to R_2$ be a difference homomorphism.

(1) $\operatorname{const} R_1$ is a subring of R_1.

(2) $\varphi(\operatorname{const} R_1) \subseteq \operatorname{const} R_2$.

(3) $\ker\varphi \trianglelefteq R_1$ is a difference ideal.

(4) If $\mathfrak{a} \trianglelefteq R_1$ is a difference ideal, then
$$s\colon R_1/\mathfrak{a} \to R_1/\mathfrak{a}, \qquad s(p+\mathfrak{a}) := s_1(p) + \mathfrak{a}$$
turns $R_1/\mathfrak{a}$ into a difference ring. The canonical projection $\pi\colon R_1 \to R_1/\mathfrak{a}$ is a surjective difference homomorphism.

(5) (Homomorphism Theorem) For every difference ideal $\mathfrak{a} \trianglelefteq R_1$ with $\mathfrak{a} \subseteq \ker\varphi$, there exists a unique difference homomorphism $\bar{\varphi}\colon R_1/\mathfrak{a} \to R_2$ that makes the diagram
$$\begin{tikzcd} R_1 \arrow[rr, "\varphi"] \arrow[dr, "\pi"'] & & R_2 \\ & R_1/\mathfrak{a} \arrow[ur, "\bar{\varphi}"'] & \end{tikzcd}$$
commute. We have $\operatorname{im}\varphi = \operatorname{im}\bar{\varphi}$, $\ker\bar{\varphi} = \pi(\ker\varphi)$, and $\ker\varphi = \pi^{-1}(\ker\bar{\varphi})$. The homomorphism $\bar{\varphi}$ is injective iff $\mathfrak{a} = \ker\varphi$.

Proof (1), (3), and (4) are in Cohn's book, Chapter 2, Sections 8 and 15. The other statements are immediate:

(2) If $c \in \operatorname{const} R_1$, then $c = s_1(c)$, so $\varphi(c) = \varphi(s_1(c)) = s_2(\varphi(c))$, and hence $\varphi(c) \in \operatorname{const} R_2$.

(5) By the homomorphism theorem for rings, we only need to show that the ring homomorphism
$$\bar{\varphi}\colon R_1/\mathfrak{a} \to R_2, \qquad \bar{\varphi}(p+\mathfrak{a}) := \varphi(p)$$
is a difference homomorphism. The calculation
$$\bar{\varphi}(s(p+\mathfrak{a})) = \bar{\varphi}(s_1(p)+\mathfrak{a}) = \varphi(s_1(p)) = s_2(p)$$
completes the proof. ∎

3 Admissible Sequences

3.1 Definition

We consider in this thesis sequences that are defined by systems of recurrences which are of a certain shape. Such systems, and the sequences they are defining, are called *admissible.*

Roughly speaking, a sequence $f(n)$ is admissible if it satisfies a recurrence that expresses $f(n+r)$ as a rational function in $f(n),\ldots,f(n+r-1)$. This rational function may, however, involve other admissible sequences. The precise definition is as follows.

Definition 3.1

(1) A system $S=\{rec_1,\ldots,rec_m\}$ of difference equations for sequences $f_1(n),\ldots,f_m(n)$ is *admissible* if each rec_i is of the form

$$\begin{aligned} f_i(n+r_i) = \mathrm{rat}_i\big(&f_1(n), \quad f_1(n+1), \quad \ldots, f_1(n+r_i-1), \quad f_1(n+r_i),\\ &f_2(n), \quad f_2(n+1), \quad \ldots, f_2(n+r_i-1), \quad f_2(n+r_i),\\ &\vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots\\ &f_{i-1}(n), f_{i-1}(n+1), \ldots, f_{i-1}(n+r_i-1), f_{i-1}(n+r_i),\\ &f_i(n), \quad f_i(n+1), \quad \ldots, f_i(n+r_i-1),\\ &\vdots \qquad\qquad\qquad\qquad \vdots\\ &f_m(n), \quad f_m(n+1), \quad \ldots, f_m(n+r_i-1)\big) \qquad\qquad (n\geq 1)\end{aligned}$$

where $r_i\geq 0$ is fixed and rat_i is some fixed multivariate rational function over $\mathbb{K}$.

The number m is called the *depth* of the system, and $r:=\max_i r_i$ is called the *order* of the system.

(2) A sequence $f(n)$ in $\mathbb{K}$ is called *admissible* if there exists an admissible system of recurrences $S=\{rec_1,\ldots,rec_m\}$ and solutions $f_1(n),\ldots,f_m(n)$ of S with $f(n)=f_i(n)$ for some i.

The admissible systems provide a data structure for representing sequences. In the sequel, we will usually assume that a tuple of sequences $f_1(n),\ldots,f_m(n)$ in $\mathbb{K}$ is given by an admissible system plus the initial values $f_i(n)\in\mathbb{K}$ for $n=1,\ldots,r_i$, $i=1,\ldots,m$. The algorithms presented in this thesis assume as input a definition of sequences $f_1(n),\ldots,f_m(n)$ in this sense, and compute from this representation various information about the involved sequences.

The way of defining a sequences by means of an admissible defining recurrence system is far from unique. For example, the zero sequence $f(n)=0$ has nontrivial representations such as $f(n+2)=f(n+1)^9-f(n)$, $f(1)=f(2)=0$.

Theorem 3.2 Let $f(n)\in\mathbb{K}^{\mathbb{N}}$ be given. Then the following statements are equivalent.

(1) $f(n)$ is admissible.

(2) There exists a system of recurrences $S = \{rec_1, \ldots, rec_m\}$ where each rec_i is of the form

$$\begin{aligned} f_i(n+r_i) = \mathrm{rat}_i\big(& f_1(n), \quad f_1(n+1), \quad \ldots\ \ldots\ \ldots\ \ldots, \quad f_1(n+r_{i,1}),\\ & f_2(n), \quad f_2(n+1), \quad \ldots\ \ldots\ \ldots\ \ldots, \quad f_2(n+r_{i,2}),\\ & \vdots \qquad\qquad\qquad\qquad\qquad\qquad\quad \vdots\\ & f_{i-1}(n), f_{i-1}(n+1), \ldots\ \ldots\ \ldots\ \ldots, \quad f_{i-1}(n+r_{i,i-1}),\\ & f_i(n), \quad f_i(n+1), \quad \ldots, f_i(n+r_i-1),\\ & \vdots \qquad\qquad\qquad\qquad \vdots\\ & f_m(n), \quad f_m(n+1), \quad \ldots, f_m(n+r_i-1)\big) \qquad\qquad (n \geq 1) \end{aligned}$$

for some fixed $r_i, r_{i,1}, \ldots, r_{i,i-1} \in \mathbb{N}$, and there are solutions $f_1(n), \ldots, f_m(n)$ of S such that $f(n) = f_i(n)$ for some i.

(3) There exists a system of recurrences $S = \{rec_1, \ldots, rec_m\}$ where each rec_i is of the form

$$\begin{aligned} f_i(n+r_i) = \mathrm{rat}_i\big(& f_1(n), \quad f_1(n+1), \quad \ldots, f_1(n+r_i-1), \quad f_1(n+r_i),\\ & f_2(n), \quad f_2(n+1), \quad \ldots, f_2(n+r_i-1), \quad f_2(n+r_i),\\ & \vdots \qquad\qquad\qquad\qquad\qquad\qquad\quad \vdots\\ & f_{i-1}(n), f_{i-1}(n+1), \ldots, f_{i-1}(n+r_i-1), f_{i-1}(n+r_i),\\ & f_i(n), \quad f_i(n+1), \quad \ldots, f_i(n+r_i-1),\\ & \vdots \qquad\qquad\qquad\qquad \vdots\\ & f_m(n), \quad f_m(n+1), \quad \ldots, f_m(n+r_i-1)\big) \qquad\qquad (n \geq 1) \end{aligned}$$

where each rat_i ($i = 1, \ldots, m$) is either a polynomial or a rational function whose numerator belongs to $\mathbb{K}$, and solutions $f_1(n), \ldots, f_m(n)$ of S such that $f(n) = f_i(n)$ for some i.

(4) There exists a system of recurrences $S = \{rec_1, \ldots, rec_m\}$ where each rec_i is of the form

$$f_i(n+1) = \mathrm{rat}_i(f_1(n), \ldots, f_m(n)) \qquad (n \geq 1)$$

and solutions $f_1(n), \ldots, f_m(n)$ of S such that $f(n) = f_i(n)$ for some i.

The depth m of the respective systems need not be the same in all cases.

Proof

(2) $\Rightarrow$ (1) Consider the indices $i \in \{1, \ldots, m\}$ for which $\max_{j=1}^{i-1} r_{i,j} > r_i$. If no such index exists, then there is nothing to prove. Otherwise, take the smallest such i.

Let $r := \max_{j=1}^{i-1} r_j$. If $r \geq r_i$, then shift the recurrence rec_i by $r_i - r$, i.e., replace every occurrence of $f_j(n+\ell)$ by $f_j(n+\ell+r_i-r)$ (for all j,ℓ). The result, rec_i', is a recurrence of order r. Next, apply the recurrences $rec_1, \ldots, rec_{i-1}$ in order to remove all occurrences of $f_j(n+\ell)$ with $\ell > r$. Note that this is possible by definition of r. The resulting recurrence is of the desired form. Replace rec_i by rec_i' in S, and repeat the procedure until there are no further indices i with $\max_{j=1}^{i-1} r_{i,j} > r_i$ left.

$(1) \Rightarrow (3)$ Let $S = \{rec_1, \ldots, rec_m\}$ be an admissible system for $f(n)$ and write rec_i in the form $f_i(n+r_i) = p_i(n)/q_i(n)$ where $p_i(n)$ and $q_i(n)$ depend polynomially on $f_j(n+\ell)$.

Introduce new function symbols $g_1, \ldots, g_m$. Then the system

$$\begin{aligned}
\{\, & g_1(n+r_1) = 1/q_1(n),\ f_1(n+r_1) = p_1(n)g_1(n+r_1),\\
& g_2(n+r_2) = 1/q_2(n),\ f_2(n+r_2) = p_2(n)g_2(n+r_2),\\
& \vdots\\
& g_m(n+r_m) = 1/q_m(n),\ f_m(n+r_m) = p_m(n)g_m(n+r_m)\,\}
\end{aligned}$$

has the desired property.

$(1) \Rightarrow (4)$ The method usually applied for linear differential or difference equations carries over to the present case. Let S be an admissible system having a tuple of solutions one of which is $f(n)$. In S, replace each occurrence of $f_i(n+j)$ $(i = 1, \ldots, m,\ j = 0, \ldots, r_i - 1)$ by $f_{i,j}(n)$ and $f_i(n+r_i)$ $(i = 1, \ldots, m)$ by $f_{i,r_i-1}(n+1)$. Add the recurrences $f_{i,j}(n+1) = f_{i,j+1}(n)$ $(i = 1, \ldots, m,\ j = 0, \ldots, r_i - 1)$ to S. Now S is transformed into a system S' of recurrences all of which have order 1. Finally, reduce each recurrence rec_i in S' with respect to all recurrences in $S' \setminus \{rec_i\}$ in order to remove occurrences of $f_{i,j}(n+1)$ on some right hand side. Then the resulting system has the desired properties.

The remaining implications are trivial. ■

The proof of the Theorem is constructive, and so we may assume in the description of algorithms that an admissible system be in one of the shapes listed in the Theorem. Of particular importance are admissible systems of the form of part (3) of the Theorem, so we assign a special name to them.

Definition 3.3 An admissible system $S = \{rec_1, \ldots, rec_m\}$ for $f_1(n), \ldots, f_m(n)$ is called *normal* if each rec_i is of the form

$$\begin{aligned}
f_i(n+r_i) = \mathrm{rat}_i\bigl(& f_1(n),\ f_1(n+1),\ \ldots, f_1(n+r_i-1),\ f_1(n+r_i),\\
& f_2(n),\ f_2(n+1),\ \ldots, f_2(n+r_i-1),\ f_2(n+r_i),\\
& \vdots \qquad\qquad\qquad\qquad\qquad\qquad \vdots\\
& f_{i-1}(n), f_{i-1}(n+1), \ldots, f_{i-1}(n+r_i-1), f_{i-1}(n+r_i),\\
& f_i(n),\ f_i(n+1),\ \ldots, f_i(n+r_i-1),\\
& \vdots \qquad\qquad\qquad \vdots\\
& f_m(n),\ f_m(n+1),\ \ldots, f_m(n+r_i-1)\bigr) \qquad (n \geq 1)
\end{aligned}$$

where each rat_i $(i = 1, \ldots, m)$ is either a polynomial or a rational function whose numerator belongs to $\mathbb{K}$.

3.2 Examples and Closure Properties

The class of admissible sequences is extensive. As illustrated in the present section, it includes a large number of sequences arising in applications.

Example 3.4

(1) Constant sequences and the sequence $f(n) := n$ are admissible. The latter satisfies the equation $f(n+1) = f(n)+1$.

(2) Every P-finite sequence is admissible.

(3) Each of the sequences in Example 2.11 satisfying a nonlinear recurrence is admissible.

(4) The sequence $f(n) := 2^{F_n}$ is admissible, a suitable admissible system is

$$\{f(n+2) = f(n)f(n+1)\}.$$

(5) The sequence $f(n) := F_{2^n}$ is admissible: Using Binet's identity

$$F_n = \frac{1}{\sqrt{5}}\Big(\Big(\frac{1+\sqrt{5}}{2}\Big)^n - \Big(\frac{1-\sqrt{5}}{2}\Big)^n\Big) \qquad (n \geq 0)$$

we can set up the admissible system

$$\{\, g(n+1) = g(n)^2,\ h(n+1) = h(n)^2,\ f(n) = (g(n)-h(n))/sqrt5\,\}.$$

The sequences $g(n) = \big(\frac{1+\sqrt{5}}{2}\big)^{2^n}$, $h(n) = \big(\frac{1-\sqrt{5}}{2}\big)^{2^n}$, and $f(n) = F_{2^n}$ form a solution of this system, and hence $f(n) = F_{2^n}$ is admissible.

(6) The sequence $f(n) := F_{F_n}$ is also admissible. Using the addition theorems for the Fibonacci sequence,

$$\begin{aligned} F_{i+j} &= F_{i+1}F_j + F_iF_{j+1} - F_iF_j \qquad && (i,j \geq 0),\\ F_{i+j+1} &= F_iF_j + F_{i+1}F_{j+1} && (i,j \geq 0), \end{aligned}$$

we obtain

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

Hence,

$$\begin{aligned} \big\{\, & f_0(n+2) = f_1(n+1)f_0(n) + f_0(n+1)f_1(n) - f_0(n+1)f_0(n),\\ & f_1(n+2) = f_0(n+1)f_0(n) + f_1(n+1)f_1(n)\,\big\} \end{aligned}$$

is an admissible system for $f_0(n) = F_{F_n}$ and $f_1(n) = F_{F_n+1}$.

Similar constructions are possible for sequences $f(g(n))$ where $f(n)$ and $g(n)$ are C-finite and the coefficients in the recurrence of g as well as its initial values are integers.

(7) The sequences $(-1)^{\lfloor \log n \rfloor}$ and $2^{n!}$ are not admissible (cf. Section 4.3).

The class of admissible sequences enjoys the following closure properties.

Theorem 3.5 Let $f(n)$ and $g(n)$ be admissible.

(1) $f(n)+g(n)$, $f(n)g(n)$, and — if $g(n)\neq 0$ for all n — $f(n)/g(n)$ are admissible.

(2) $S(n):=\sum_{k=1}^{n} f(k)$ and $P(n):=\prod_{k=1}^{n} f(k)$ are admissible.

(3) If $f(n)\neq 0$ and $g(n)\neq 0$ for all $n\geq 1$, then $K(n):=\mathrm{K}_{k=1}^{n}\, g(k)/f(k)$ is admissible.

Proof Let S_f and S_g be admissible systems for f and g, respectively, and assume without loss of generality that S_f and S_g have no function symbols in common.

Then $S_f\cup S_g\cup\{F(n)=f(n)+g(n)\}$ is an admissible system for $f(n)+g(n)$, and admissible systems for $f(n)g(n)$ and $f(n)/g(n)$ are obtained analogously.

For the indefinite sum and product, we have the admissible systems

$$S_f\cup\{S(n+1)=S(n)+f(n+1)\} \text{ and } S_f\cup\{P(n+1)=P(n)f(n+1)\},$$

respectively.

For the continued fraction,

$$\begin{aligned} S_f\cup S_g\cup\bigl\{&p(n+2)=f(n+2)p(n+1)+g(n+2)p(n),\\ &q(n+2)=f(n+2)q(n+1)+g(n+2)q(n),\ K(n)=p(n)/q(n)\bigr\}\end{aligned}$$

is an admissible system (cf. page 12). ■

It is worth noting that Theorem 3.5 is constructive: If $f_1(n),\ldots,f_m(n)$ are admissible sequences for which a defining system is given, and if $F(n)$ is defined from $f_1(n),\ldots,f_m(n)$ by an expression involving field operations as well as indefinite sums, products and continued fractions, arbitrarily mixed and nested, then a defining system for $F(n)$ can be easily computed. The computation involves hardly more than simple rewriting. The software package described in Chapter 9 has routines for carrying out this rewriting procedure automatically.

Example 3.6

(1) The sequence

$$f(n):=\sum_{k=1}^{n}\frac{\Bigl(\sum_{i=1}^{k}\frac{i!-1}{i^2+1}(-1)^i P_j(x)^2\Bigr)\bigl((F_k^2-F_{3^k})^2+1\bigr)}{F_k^2+\mathrm{K}_{i=1}^{k}(P_i(x^2)/P_{i+1}(x))}$$

is obviously admissible.

(2) Any sequence $f(n)$ which is ultimately zero, i.e., $f(n)=0$ for $n\geq n_0$ is admissible. To see this, consider the auxiliary sequence

$$z(n):=\prod_{k=0}^{n}(k-n_0).$$

We have $z(n)\neq 0$ for $n=1,2,\ldots,n_0-1$ and $z(n)=0$ for $n\geq n_0$. Let $p(n)\in\mathbb{K}[n]$ be such that $p(n)=f(n)/z(n)$ for $n=1,2,\ldots,n_0-1$. Such a polynomial is easily found by interpolation. Then $f(n) = p(n)z(n)$, and by reference to the closure properties, $p(n), z(n)$, and hence also $f(n)$ are admissible.

More generally, we have shown that if $f(n)$ is admissible then every representative of the germ of $f(n)$ at infinity is admissible, too.

By a different argument, it can be shown that also the class of P-finite sequences has this property (Stanley, 1999).

(3) The sequence $f(n)$ defined by

$$f(n) = \sum_{k=0}^{n} \Big(\sum_{i=0}^{k} \binom{n}{i} \Big)^3$$

(Calkin, 1994) is admissible, because it satisfies the recurrence

$$f(n+2) = \frac{1}{n+1}\big((7n+12)f(n+1) + 4(2n+1)f(n) + (9n-10)2^{3n+1}\big),$$

according to the Sigma package (Schneider, 2001). Note that the admissibility of $f(n)$ does not follow from Theorem 3.5, because $f(n)$ does not satisfy the telescoping equation $f(n+1) = f(n) + \mathrm{summand}(n+1)$. A sum satisfying this equation is called an *indefinite sum.* All other sums are called *definite.*

Roughly speaking, a sum is indefinite if the variables appearing in the bounds of the sum do not occur in the summand.

Theorem 3.7 Let $f(n)$ be admissible and $a \in \mathbb{N}$. Then

(1) The sequence $f(n+a)$ is admissible.
(2) The sequence $f(\lfloor n/a \rfloor)$ is admissible.
(3) The sequence $f(an)$ is admissible.

Consequently, $f(\lfloor pn+q \rfloor + a)$ is admissible for all nonnegative $p, q \in \mathbb{Q}$, $a \in \mathbb{N}_0$.

Proof Let $S = \{rec_1, \ldots, rec_m\}$ be an admissible system of recurrences in $f_1(n), \ldots, f_m(n)$, and let $r_1, r_2, \ldots, r_m$ be the respective orders of $rec_1, \ldots, rec_m$.

Suppose a tuple of solutions $g_1(n), \ldots, g_m(n)$ of S is given.

(1) The sequences $g_1(n+a), \ldots, g_m(n+a)$ also form a set of solution for S. These solutions are obtained by stipulating the initial conditions $f_i(j) = g_i(a+j)$ ($i = 1, \ldots, m$, $j = 1, \ldots, r_i$).
(2) For all $n, a, r \in \mathbb{N}$ we have $\lfloor n/a \rfloor + r = \lfloor (n+ar)/a \rfloor$. It follows that a recurrence system for the solution set $g_1(\lfloor n/a \rfloor), \ldots, g_m(\lfloor n/a \rfloor)$ is obtained from S by replacing each occurrence of $f_i(n+j)$ by $f_i(n+aj)$. Note that the resulting system is of the form required in Definition 3.1.
(3) By Theorem 3.2.(4), we may assume without loss of generality that each $rec_i \in S$ is of the form

$$f_i(n+1) = \mathrm{rat}_i(f_1(n), \ldots, f_{i-1}(n)) \qquad (n \geq 1).$$

If this holds for n, then it continues to hold for $an+j$ in place of n:

$$\begin{aligned}
f_i(an+1) &= \mathrm{rat}_i(f_1(an),\dots,f_{i-1}(an)),\\
f_i(an+2) &= \mathrm{rat}_i(f_1(an+1),\dots,f_{i-1}(an+1)),\\
&\ \ \vdots\\
f_i(an+a) &= \mathrm{rat}_i(f_1(an+(a-1)),\dots,f_{i-1}(an+(a-1))).
\end{aligned}$$

Writing $g_{i,j}(n)$ for $f_i(an+j)$, we find that

$$\begin{aligned}
\{\,&g_{1,0}(n+1) = \mathrm{rat}_1(g_{1,a-1}(n)),\\
&g_{1,1}(n) = \mathrm{rat}_1(g_{1,0}(n)),\dots,\ \ g_{1,a-1}(n) = \mathrm{rat}_1(g_{1,a-2}(n)),\\
&g_{2,0}(n+1) = \mathrm{rat}_2(g_{1,a-1}(n), g_{2,a-1}(n)),\\
&g_{2,1}(n) = \mathrm{rat}_2(g_{1,0}(n), g_{2,0}(n)),\dots,\ \ g_{2,a-1}(n) = \mathrm{rat}_m(g_{1,a-2}(n), g_{2,a-2}(n)),\\
&\ \ \vdots\\
&g_{m,0}(n+1) = \mathrm{rat}_m(g_{1,a-1}(n),\dots,g_{m,a-1}(n)),\\
&g_{m,1}(n) = \mathrm{rat}_m(g_{1,0}(n),\dots,g_{m,0}(n)),\dots,\ \ g_{m,a-1}(n) = \mathrm{rat}_m(g_{1,a-2}(n),\dots,g_{m,a-2}(n))\,\}
\end{aligned}$$

is an admissible system with the desired property. ∎

Example 3.8 Let $f(n)$ and $g(n)$ be admissible and let $h(n)$ be defined by

$$h(2n) := f(n), \quad h(2n-1) := g(n) \qquad (n \geq 1).$$

The sequence $h(n)$ defined in this way is called the *interlacing* of $f(n)$ and $g(n)$. Its first values are $f(1), g(1), f(2), g(2), f(3), g(3), \dots$.

The identity

$$h(n) = \tfrac{1}{2}(1-(-1)^n)f(\lfloor n/2\rfloor + 1) + \tfrac{1}{2}(1+(-1)^n)g(\lfloor n/2\rfloor)$$

asserts that $h(n)$ is admissible, too.

More generally, if $f_1(n),\dots,f_m(n)$ are admissible then so is their interlacing sequence $h(n)$, which starts

$$f_1(1), f_2(1), \dots, f_m(1), f_1(2), f_2(2), \dots\dots.$$

The construction is the same as in the case of holonomic sequences (Mallinger, 1996).

3.3 Evaluation of Admissible Sequences

Admissible systems may be used to evaluate admissible sequences in a dynamic programming style, as follows. (See page 135 for a description of the pseudo code constructions that we are using.)

Algorithm 3.9 (Evaluation of Admissible Sequences)
Input: Admissible sequences $f_1(n),\dots,f_m(n)$ over $\mathbb{K}$, defined by an admissible system $S = \{rec_1,\dots,rec_m\}$ as in Def. 3.1 and initial values; a number $n_0 \in \mathbb{N}$
Output: The values $(f_1(n_0),\dots,f_m(n_0)) \in \mathbb{K}^m$
Assumption: rec_i has the form $f_i(n+r_i) = \mathrm{rat}_i$ where rat_i is a rational function depending on $f_j(n+\ell)$ $(\ell = 0,\dots,r_j-1,\ j = 1,\dots,m)$

$T := [[f_i(j) : j = 1,\dots,r_i-1] : i = 1,\dots,m]$
for $\ell = 1$ **to** n_0 **do**
  $T := [\mathrm{append}([T_{i,j+1} : j = 1,\dots,r_i-2], \mathrm{rat}_i(T)) : i = 1,\dots,m]$
return $[T_{i,1} : i = 1,\dots,m]$

Theorem 3.10 Algorithm 3.9 is correct and consumes $\mathrm{O}(n_0 m \max_i r_i)$ field operations and a constant amount of memory. ■

Example 3.11 Consider the sequence $s(n)$ defined via

$$s(n) = \sum_{i_7=1}^{n}\Big(\frac{1}{i_7^2}\sum_{i_6=1}^{i_7}\Big(\frac{1}{i_6^2}\sum_{i_5=1}^{i_6}\Big(\frac{1}{i_5^2}\sum_{i_4=1}^{i_5}\Big(\frac{1}{i_4^2}\sum_{i_3=1}^{i_4}\Big(\frac{1}{i_3^2}\sum_{i_2=1}^{i_3}\Big(\frac{1}{i_2^2}\sum_{i_1=1}^{i_2}\frac{1}{i_1^2}\Big)\Big)\Big)\Big)\Big)\Big)$$

The software of Chapter 9 obtains the exact value of

$$s(75) \approx 1.973568805$$

by using Algorithm 3.9 in about 0.65 seconds on our machine (1.5 GHz), while Mathematica's default evaluation requires about 4 hours and 45 minutes.

It is of course possible to evaluate $s(n)$ for much larger values of n in a reasonable time. The value

$$s(1000) \approx 1.997880822$$

suggests the conjecture that $\lim_{n\to\infty} s(n) \stackrel{?}{=} 2$. Conjectures like this have to be made with care (Pemantle and Schneider, 2004), and in fact, the conjecture is wrong.

It is, however, not entirely wrong. If we let $s_i(n)$ denote the sequence defined just like $s(n)$, but with i summation signs (so $s(n) = s_7(n)$), and put $s_i := \lim_{n\to\infty} s_i(n)$, then it can be shown using generating functions and asymptotic analysis (Odlyzko, 1995), that $\lim_{i\to\infty} s_i = 2$.

3.4 The Associated Difference Ideal

Let $f_1(n),\dots,f_m(n)$ be admissible sequences in $\mathbb{K}$, defined by some admissible system S and initial values. The algorithms presented in the subsequent chapters will work on difference rings containing elements $t_1,\dots,t_m$ that "represent" the sequences $f_1(n),\dots,f_m(n)$ in a certain sense.

Consider the difference homomorphism

$$\varphi\colon (\mathbb{K}\{t_1,\dots,t_m\}, s) \to (\mathbb{K}^{\mathbb{N}}, \mathbf{E}), \qquad t_i \mapsto f_i(n).$$

According to Theorem 2.15.(5), we have

$$\mathbb{K}\{t_1,\ldots,t_m\}/\ker\varphi \cong \operatorname{im}\varphi \subseteq \mathbb{K}^{\mathbb{N}}.$$

As the sequences $f_i(n)$ are assumed to be admissible, the kernel $\ker\varphi$ cannot be empty. For each recurrence

$$f_i(n+r_i) = \frac{p_i\left(f_1(n),\ldots,f_m(n+r_i-1)\right)}{q_i\left(f_1(n),\ldots,f_m(n+r_i-1)\right)}$$

it must at least contain the difference polynomial

$$q_i(t_1,\ldots,s^{r_i-1}t_m)s^{r_i}t_j - p_i(t_1,\ldots,s^{r_i-1}t_m).$$

The difference ideal generated by these polynomials is called the *associated difference ideal* of S.

Definition 3.12 Let S be an admissible system for $f_1(n),\ldots,f_m(n)$. For each recurrence

$$f_i(n+r_i) = \frac{p_i\left(f_1(n),\ldots,f_m(n+r_i-1)\right)}{q_i\left(f_1(n),\ldots,f_m(n+r_i-1)\right)} \qquad (i=1,\ldots,m)$$

in S, define

$$P_i := q_i(t_1,\ldots,s^{r_i-1}t_m)s^{r_i}t_j - p_i(t_1,\ldots,s^{r_i-1}t_m) \in \mathbb{K}\{t_1,\ldots,t_m\} \qquad (i=1,\ldots,m).$$

Then the difference ideal

$$\langle\langle P_1,\ldots,P_m\rangle\rangle \trianglelefteq \mathbb{K}\{t_1,\ldots,t_m\}$$

is called the *associated difference ideal* of S.

If $\mathfrak{a}$ is the associated difference ideal of S, then, in the notation above, $\mathfrak{a} \subseteq \ker\varphi$. Equality does not hold in general. For example, define $f(n) := (-1)^n$ via the recurrence $f(n+1) = -f(n)$ and the initial value $f(1) = -1$. If $\varphi\colon \mathbb{K}\{t\} \to \mathbb{K}^{\mathbb{N}}$ maps t to $(-1)^n$ and $\mathfrak{a} \trianglelefteq \mathbb{K}\{t\}$ is the associated difference ideal, then

$$\mathfrak{a} = \langle\langle st+t\rangle\rangle \subsetneq \langle\langle t^2-1, st+t\rangle\rangle = \ker\varphi.$$

Nonzero difference polynomials in $\ker\varphi \setminus \mathfrak{a}$, if they exist, are nontrivial *algebraic dependencies.*They will be further discussed in Chapter 6.

If the q_i in definition above are all constants, then

$$\mathbb{K}\{t_1,\ldots,t_m\}/\langle\langle P_1,\ldots,P_m\rangle\rangle \cong \mathbb{K}[t_1,\ldots,s^{r_1-1}t_1,t_2,\ldots,s^{r_2-1}t_2,\ldots\ldots t_m,\ldots,s^{r_m-1}t_m],$$

i.e., the underlying ring of $\mathbb{K}\{t_1,\ldots,t_m\}/\langle\langle P_1,\ldots,P_m\rangle\rangle$ is an ordinary polynomial ring, and the difference ideal only defines the action of s on this ring. Taking quotient fields, we obtain

$$Q\left(\mathbb{K}\{t_1,\ldots,t_m\}/\langle\langle P_1,\ldots,P_m\rangle\rangle\right) \cong \mathbb{K}(t_1,\ldots,s^{r_1-1}t_1,t_2,\ldots,s^{r_2-1}t_2,\ldots\ldots t_m,\ldots,s^{r_m-1}t_m).$$

This latter isomorphism continues to hold for arbitrary denominators q_i if the system S is normal, as we will see below.

Definition 3.13 Let S be an admissible system for $f_1(n),\dots,f_m(n)$, and let the difference polynomials $P_1,\dots,P_m \in \mathbb{K}\{t_1,\dots,t_m\}$ be defined as in Def. 3.12, $r_1,\dots,r_m \in \mathbb{N}$ their respective orders. Then for each $r \in \mathbb{N}$ the ideal

$$\begin{aligned}&\langle P_1, sP_1,\dots,s^{r-r_1}P_1, P_2, sP_2,\dots,s^{r-r_2}P_2,\dots\dots,P_m,sP_m,\dots,s^{r-r_m}P_m\rangle\\ &\qquad \trianglelefteq \mathbb{K}[t_1,st_1,\dots,s^r t_1,t_2,st_2,\dots,s^r t_2,\dots\dots,t_m,st_m,\dots,s^r t_m] =: \mathbb{K}\{t_1,\dots,t_m\}_r\end{aligned}$$

is called the *associated polynomial ideal* of order r of S.

We want to show that the associated polynomial ideals of some normal admissible system are precisely the elimination ideals of the associated difference ideals with respect to $\mathbb{K}\{t_1,\dots,t_m\}_r$. For this we need the following lemma. Roughly speaking, it says that extending a ring R by an element $q \in R$ does not change R, and extending R by $1/q$ does not change the quotient field.

Lemma 3.14 Let $\mathfrak{p} \trianglelefteq \mathbb{K}[X] := \mathbb{K}[x_1,\dots,x_n]$ be a prime ideal.

(1) For all $q \in \mathbb{K}[X]$, the ideal $\mathfrak{p}' := \langle \mathfrak{p} \cup \{p\}\rangle \trianglelefteq \mathbb{K}[X,y]$ with $p = y-q$ is prime and $\mathbb{K}[X]/\mathfrak{p} \cong \mathbb{K}[X,y]/\mathfrak{p}'$.

(2) For all $q \in \mathbb{K}[X]\setminus\mathfrak{p}$, the ideal $\mathfrak{p}' := \langle \mathfrak{p} \cup \{p\}\rangle \trianglelefteq \mathbb{K}[X,y]$ with $p = qy-1$ is prime and $Q(\mathbb{K}[X]/\mathfrak{p}) \cong Q(\mathbb{K}[X,y]/\mathfrak{p}')$.

Proof Write $R := \mathbb{K}[X]/\mathfrak{p}$ and $R' := \mathbb{K}[X,y]/\mathfrak{p}'$. By $\mathfrak{p}' = \langle\mathfrak{p}\rangle + \langle p\rangle$, we have $R' \cong R[y]/\langle p\rangle$.

(1) Consider the homomorphism $\phi\colon R[y] \to R$ with $\phi(y) = q$. We obviously have $\ker\phi = \langle y-q\rangle$, and hence $R' \cong R[y]/\langle p\rangle \cong R$, as claimed.

(2) As $\mathfrak{p}$ is prime, R is an integral domain and we may consider $Q(R)$. By $q \notin \mathfrak{p}$, there must be an element $1/q \in Q(R)$. Consider the homomorphism $\phi\colon R[y] \to Q(R)$ with $\phi(y) = 1/q$. By $\langle qy-1\rangle \subseteq \ker\phi$ we get an induced homomorphism $\bar\phi\colon R' \to Q(R)$. The proof is complete if we can show that $\bar\phi$ is an embedding, for then $R \hookrightarrow R' \hookrightarrow Q(R)$ implies that $Q(R) \hookrightarrow Q(R') \hookrightarrow Q(R)$ and hence $Q(R') \cong Q(R)$.

Indeed, $\bar\phi$ is injective: we have to show $\ker\phi \subseteq \langle qy-1\rangle$. Let $a = \sum_{k=0}^n a_k y^k \in \ker\phi$, and assume that $a \notin \langle qy-1\rangle$. We may assume without loss of generality that $q \nmid a_n$ in R. We have

$$0 = \phi(a) = \sum_{k=0}^n a_k \phi(y)^k = \sum_{k=0}^n \frac{a_k}{q^k} = \frac{1}{q^n}\sum_{k=0}^n a_k q^{n-k}.$$

As $1/q^n \neq 0$, it follows that

$$0 = \sum_{k=0}^n a_k q^{n-k} = a_n + q\underbrace{\sum_{k=0}^{n-1} a_k q^{(n-1)-k}}_{\in R},$$

and hence $q \mid a_n$ in contradiction to the assumption on a. ∎

Theorem 3.15 Let S be a normal admissible system for $f_1(n),\dots,f_m(n)$.

(1) The associated polynomial ideals $\mathfrak{a} \trianglelefteq \mathbb{K}\{t_1,\dots,t_m\}_r$ of any order $r \in \mathbb{N}$ are prime.

(2) The associated difference ideal $\mathfrak{a} \trianglelefteq \mathbb{K}\{t_1,\dots,t_m\}$ is prime.

Proof

(1) Let $P_1,\dots,P_m \in \mathbb{K}\{t_1,\dots,t_m\}$ be as in Def. 3.12 and $r_1,\dots,r_m$ be their respective orders. Define the ideals

$$\mathfrak{a}_{i,j} \trianglelefteq \mathbb{K}[t_1,\dots,t_m,\dots\dots,s^{j-1}t_1,\dots,s^{j-1}t_m,s^jt_1,\dots,s^jt_i]$$

recursively via

$$\mathfrak{a}_{i,j} := \begin{cases} \langle \mathfrak{a}_{i-1,j}\rangle & \text{if } j < r_i \\ \langle \mathfrak{a}_{i-1,j} \cup \{s^{j-r_i}P_i\}\rangle & \text{otherwise} \end{cases}$$

where we identify $\mathfrak{a}_{0,j}$ with $\mathfrak{a}_{m,j-1}$ for simplicity of notation. As base case, define $\mathfrak{a}_{0,0} := \{0\} \trianglelefteq \mathbb{K}$.

Clearly $\mathfrak{a}_{0,0}$ is prime, and by Lemma 3.14 the primeness of each $\mathfrak{a}_{i,j}$ carries over to $\mathfrak{a}_{i+1,j}$, because by the normality of S the difference polynomials s^jP_i are precisely of the form required in Lemma 3.14.

The proof is completed by noting that $\mathfrak{a} = \mathfrak{a}_{m,r}$.

(2) Let $\mathfrak{a}_1,\mathfrak{a}_2,\mathfrak{a}_3,\dots$ be the associated polynomial ideals of order $1,2,3,\dots$, and denote by $\langle\mathfrak{a}_i\rangle$ the ideal generated by $\mathfrak{a}_i$ in $\mathbb{K}\{t_1,\dots,t_m\}$. Then we have the chain $\langle\mathfrak{a}_1\rangle \subseteq \langle\mathfrak{a}_2\rangle \subseteq \langle\mathfrak{a}_3\rangle \subseteq \cdots$ and the identity

$$\mathfrak{a} = \{\, p \in \mathbb{K}\{t_1,\dots,t_m\} : \exists\, i : p \in \langle\mathfrak{a}_i\rangle \,\}. \tag{3.1}$$

Proof of (3.1). The inclusion "$\supseteq$" is evident. For the inclusion "$\subseteq$", take an element $a \in \mathfrak{a}$. If $P_1,\dots,P_m$ are the generators of $\mathfrak{a}$ from Def. 3.12, then a can be written in the form

$$a = \sum_{i=1}^{m}\sum_{j=0}^{\infty} a_{i,j}\, s^jP_i$$

for some $a_{i,j} \in \mathbb{K}\{t_1,\dots,t_m\}$ almost all of which are zero. If r denotes the maximum index j such that $a_{i,j} \neq 0$ for some i, then $a \in \langle a_r\rangle$. This proves (3.1).

In order to see that $\mathfrak{a}$ is prime, let $p,q \in \mathbb{K}\{t_1,\dots,t_m\}$ be such that $pq \in \mathfrak{a}$. Then, by (3.1), $pq \in \langle\mathfrak{a}_i\rangle$ for some i. The ideals $\mathfrak{a}_i$, and hence also the $\langle\mathfrak{a}_i\rangle$, are prime by part (1) of the Theorem. It follows that $p \in \langle\mathfrak{a}_i\rangle$ or $q \in \langle\mathfrak{a}_i\rangle$, which implies $p \in \mathfrak{a}$ or $q \in \mathfrak{a}$ by using (3.1). ■

Theorem 3.15 justifies that $\mathbb{K}\{t_1,\dots,t_m\}/\mathfrak{a}$ is an integral domain when $\mathfrak{a}$ is the associated difference ideal of some normal admissible system S. Hence the quotient field $Q(\mathbb{K}\{t_1,\dots,t_m\}/\mathfrak{a})$ is well defined in that case.

Another consequence of Theorem 3.15 is that the associated polynomial ideals are precisely the elimination ideals of the associated difference ideals w.r.t. the polynomial rings $\mathbb{K}\{t_1,\dots,t_m\}_r$.

Theorem 3.16 Let S be a normal admissible system for $f_1(n),\dots,f_m(n)$, let $\mathfrak{a} \trianglelefteq \mathbb{K}\{t_1,\dots,t_m\}$ be the associated difference ideal and denote by $\mathfrak{a}_r \trianglelefteq \mathbb{K}\{t_1,\dots,t_m\}_r$ the associated polynomial ideal of order r. Then

(1) $\mathfrak{a}_{r+1} \cap \mathbb{K}\{t_1,\dots,t_m\}_r = \mathfrak{a}_r$ for every $r \geq 1$
(2) $\mathfrak{a}_{\tilde{r}} \cap \mathbb{K}\{t_1,\dots,t_m\}_r = \mathfrak{a}_r$ for every $\tilde{r} \geq r \geq 1$
(3) $\mathfrak{a} \cap \mathbb{K}\{t_1,\dots,t_m\}_r = \mathfrak{a}_r$ for every $r \geq 1$

Proof Let $P_1,\dots,P_m \in \mathbb{K}\{t_1,\dots,t_m\}$ be as in Def. 3.12, and let $r_1,\dots,r_m$ be their respective orders.

(1) "$\supseteq$" obvious. "$\subseteq$" Recall that, for every integral domain R and every $p \in R[x] \setminus R$, we have $\langle p \rangle \cap R = \{0\}$. This implies that for every prime ideal $\mathfrak{p} \trianglelefteq \mathbb{K}[X]$ and every $p_0, p_1 \in \mathbb{K}[X]$ with $p_1 \notin \mathfrak{p}$, we have $\langle \mathfrak{p} \cup \{p_1 y - p_0\} \rangle \cap \mathbb{K}[X] = \mathfrak{p}$, where the ideal on the left is in $\mathbb{K}[X,y]$.

Define the ideals

$$\mathfrak{a}_{r,i} \trianglelefteq \mathbb{K}[t_1,\dots,t_m,\dots\dots,s^r t_1,\dots,s^r t_m,s^{r+1}t_1,\dots,s^{r+1}t_i] \qquad (i=0,\dots,m)$$

recursively via $\mathfrak{a}_{r,0} := \mathfrak{a}_r$ and

$$\mathfrak{a}_{r,i} := \begin{cases} \langle \mathfrak{a}_{r,i-1} \rangle & \text{if } r+1 < r_i \\ \langle \mathfrak{a}_{r,i-1} \cup \{s^{r+1-r_i} P_i\} \rangle & \text{otherwise} \end{cases}$$

By Theorem 3.15, $\mathfrak{a}_r = \mathfrak{a}_{r,0}$ is prime, and by applying Lemma 3.14 and the remark above repeatedly, we obtain that $\mathfrak{a}_{r,i}$ is prime and

$$\mathfrak{a}_{r,i+1} \cap \mathbb{K}[t_1,\dots,t_m,\dots\dots,s^r t_1,\dots,s^r t_m,s^{r+1}t_1,\dots,s^{r+1}t_i] = \mathfrak{a}_{r,i} \qquad (i=0,\dots,m-1).$$

The proof is completed by noting that $\mathfrak{a}_{r+1} = \mathfrak{a}_{r,m}$.

(2) By repeated application of part (1), we find

$$\begin{aligned}
& \mathfrak{a}_{\tilde r} \cap \mathbb{K}\{t_1,\dots,t_m\}_{\tilde r-1} = \mathfrak{a}_{\tilde r-1} \\
\Longrightarrow\quad & \mathfrak{a}_{\tilde r} \cap \mathbb{K}\{t_1,\dots,t_m\}_{\tilde r-2} = \mathfrak{a}_{\tilde r-1} \cap \mathbb{K}\{t_1,\dots,t_m\}_{\tilde r-2} = \mathfrak{a}_{\tilde r-2} \\
\Longrightarrow\quad & \mathfrak{a}_{\tilde r} \cap \mathbb{K}\{t_1,\dots,t_m\}_{\tilde r-3} = \mathfrak{a}_{\tilde r-1} \cap \mathbb{K}\{t_1,\dots,t_m\}_{\tilde r-3} = \mathfrak{a}_{\tilde r-2} \cap \mathbb{K}\{t_1,\dots,t_m\}_{\tilde r-3} = \mathfrak{a}_{\tilde r-3} \\
\Longrightarrow\quad & \cdots \\
\Longrightarrow\quad & \mathfrak{a}_{\tilde r} \cap \mathbb{K}\{t_1,\dots,t_m\}_r = \mathfrak{a}_r.
\end{aligned}$$

(3) "$\supseteq$" Obvious. "$\subseteq$" Every $a \in \mathfrak{a} \cap \mathbb{K}\{t_1,\dots,t_m\}_r$ can be written in the form

$$a = \sum_{i=1}^{m} \sum_{j=0}^{\infty} a_{i,j} s^j P_i$$

for some $a_{i,j} \in \mathbb{K}\{t_1,\dots,t_m\}$, almost all zero. There exists an $r \in \mathbb{N}$ such that $a_{i,j} s^j P_i \in \mathbb{K}\{t_1,\dots,t_m\}_{\tilde r}$, and hence $a \in \mathfrak{a}_{\tilde r} \cap \mathbb{K}\{t_1,\dots,t_m\}_r$. By part (2), it follows that $a \in \mathfrak{a}_r$. ∎

4 Zero Equivalence of Admissible Sequences

In this chapter we describe an algorithm which decides whether a sequence, which is given by an admissible system plus initial values, equals the zero sequence. By the closure of the class of admissible sequences under field operations, this algorithm immediately gives rise to an algorithm for proving combinatorial identities $f(n) = g(n)$ with admissible sequences on both sides of the identity. Further applications of this algorithm will be encountered in subsequent chapters.

The algorithm proceeds by setting up a proof by complete induction on n. A similar algorithm was proposed by Shackell (1993, 2004) for the differential case. That algorithm shows by a complete induction that all Taylor coefficients of a given function $f(x)$ are zero. If the attention is restricted to analytic functions, this is a sufficient condition for $f(x)$ to be zero entirely.

4.1 Deciding Zero Equivalence Algorithmically

Suppose we are given an admissible sequence $f(n)$ by an admissible system S of difference equations and initial values. In order to decide whether $f(n) = 0$ for all n, we proceed in two steps. The first step consists of computing a number $N \geq 0$ with the property that $f(n) = 0$ for all $n \in \mathbb{N}$ if and only if $f(n) = 0$ for $n \leq N$. Once such a number N is known, deciding whether $f(n) = 0$ for all $n \in \mathbb{N}$ reduced to evaluating $f(n)$ for $n = 1,2,\dots,N$ and deciding zero equivalence of these constants.

Algorithm 4.1 settles the first step. For $N = 1,2,\dots$, it determines whether the implication

$$\forall\, n \geq 0 : f(n) = f(n+1) = \cdots = f(n+N) = 0 \Longrightarrow f(n+N+1) = 0$$

follows from the recurrences in the admissible system S. It will be shown below (Theorem 4.5) that this will be the case if N is large enough.

Algorithm 4.1 (Induction Step)
Input: An admissible system S for sequences $f_1(n),\dots,f_m(n)$, an index $i \in \{1,\dots,m\}$.
Output: A number $N \in \mathbb{N}$ such that $f_i(n) = \cdots = f_i(n+N) = 0$ implies $f_i(n+N+1) = 0$ for all $n \in \mathbb{N}$

1 Replace S by an equivalent normal admissible system (cf. Theorem 3.2.(3))
2 Let $\mathfrak{a} \trianglelefteq \mathbb{K}\{t_1,\dots,t_m\}$ be the associated difference ideal of S
3 Set N to the order of S
4 **while** $s^{N+1}t_i \notin \operatorname{Rad}\left(\langle t_i, st_i, \dots, s^N t_i\rangle + \mathfrak{a} \cap \mathbb{K}\{t_1,\dots,t_m\}_{N+1}\right)$ **do**
5 $\quad N := N+1$
6 **return** N

Observe that all the steps in Algorithm 4.1 are computable. Step 1 can be carried out as shown in the proof of Theorem 3.2. Step 2 is nothing more than a rewriting of the recurrences in S

in terms of difference polynomials. Step 4 contains several substeps. First, we have to compute the elimination ideal $\mathfrak{a} \cap \mathbb{K}\{t_1,\ldots,t_m\}_{N+1}$. According to Theorem 3.16, this ideal is just the associated polynomial ideal of order N, and the generators of $\mathfrak{a}$ give rise to generators of $\mathfrak{a} \cap \mathbb{K}\{t_1,\ldots,t_m\}_{N+1}$. As $\mathbb{K}\{t_1,\ldots,t_m\}_{N+1}$ is now just a polynomial ring, deciding radical membership is a standard application of Gröbner bases (Theorem 2.7.(2)).

The zero equivalence algorithm may now be stated as follows.

Algorithm 4.2 (Zero Equivalence)
Input: An admissible system S for sequences $f_1(n),\ldots,f_m(n)$, initial values, a distinguished index $i \in \{1,\ldots,m\}$.
Output: True if $f_i(n) = 0$ for all $n \geq 1$, False otherwise.

```
1  Apply Algorithm 4.1 to S and i, obtaining the number N
2  for n = 1 to N + 1 do
3     Compute the value f_i(n) by Algorithm 3.9
4     if f_i(n) ≠ 0 then
5        return False
6  return True
```

Correctness and termination of Algorithm 4.1 immediately imply correctness and termination of Algorithm 4.2. If Algorithm 4.2 returns False, then it has actually found a number n with $f_i(n) \neq 0$, so $f_i(n)$ is not the zero sequence. On the other hand, if it returns True, then there is a proof by complete induction to n that $f_i(n) = 0$ for all natural n: the termination of the loop in lines 2–5 asserts that $f_i(1) = f_i(2) = \cdots = f_i(N+1) = 0$. This serves as induction base, and the definition of N supplies the induction step.

We have divided zero equivalence test into two algorithms because we will need Algorithm 4.1 later, in Chapter 6, as a subroutine. For the sole purpose of testing zero equivalence, it is more convenient to apply a "merged" version of the two algorithms where $f_i(N)$ is evaluated in the while loop of Algorithm 4.1. If it is nonzero, we can stop—we do not need to know the precise value of N anymore.

We will next prove correctness and termination of Algorithm 4.1.

Lemma 4.3 Let $f_1(n),\ldots,f_m(n)$ be admissible sequences, defined by some admissible system S of order r. For some $N \geq r$, let $\mathfrak{a} \trianglelefteq \mathbb{K}\{t_1,\ldots,t_m\}_{N+1}$ denote the $(N+1)$st associated polynomial ideal of S. Let $n \in \mathbb{N}$. Then

$$\begin{aligned} &\mathfrak{a} + \langle t_1 - f_1(n),\ \ldots, s^N t_1 - f_1(n+N), \ldots\ldots, \\ &\qquad t_m - f_m(n), \ldots, s^N t_m - f_m(n+N)\rangle \cap \mathbb{K}[s^{N+1}t_1,\ldots,s^{N+1}t_m] \\ &= \langle s^{N+1}t_1 - f_1(n+N+1),\ldots,s^{N+1}t_m - f_m(n+N+1)\rangle. \end{aligned}$$

Proof By $N \geq r$, the ideal $\mathfrak{a}$ contains a polynomial p which is free of $s^{N+1}t_2,\ldots,s^{N+1}t_m$ and depends only linearly on $s^{N+1}t_1$. As $s^i t_j - f_j(n+i) \in \mathfrak{a}$, the polynomial p' which is obtained from p by replacing $s^i t_j$ by $f_j(n+i)$ belongs to $\mathfrak{a}$ as well. But this polynomial is, up to multiplication by a constant, equal to $s^{N+1}t_1 - f_1(n+N+1)$.

The same argument, applied consecutively, shows that also $s^{N+1}t_2 - f_2(n+N+1), \dots, s^{N+1}t_m - f_m(n+N+1) \in \mathfrak{a}$. This establishes the inclusion "$\supseteq$." As the ideal on the right is maximal, we can only have "$\supsetneq$" if it is equal to $\langle 1\rangle$, but this is impossible because the sequences $f_i(n)$ are assumed to be solutions of S. ∎

Theorem 4.4 Algorithm 4.1 is correct. That is, the number N delivered by the algorithm has the property claimed in the specification.

Proof Suppose the algorithm returns the number N. Let $\mathfrak{a} \cap \mathbb{K}\{t_1,\dots,t_m\}_{N+1} = \langle p_1,\dots,p_\ell\rangle$, and define $\mathfrak{a}' := \langle p_1,\dots,p_\ell,t_i,st_i,\dots,s^N t_i\rangle$.
Suppose that $n \in \mathbb{N}$ is such that $f_i(n) = \cdots = f_i(n+N) = 0$. Then, using Lemma 4.3,

$$\begin{aligned}
&\mathfrak{a}' \cap \mathbb{K}[s^{N+1}t_1,\dots,s^{N+1}t_m]\\
&\subseteq \langle p_1,\dots,p_\ell\rangle + \langle s^i t_j - f_j(n+i) : i=0,\dots,N; j=1,\dots,m\rangle \cap \mathbb{K}[s^{N+1}t_1,\dots,s^{N+1}t_m]\\
&= \langle s^{N+1}t_1 - f_1(n+N+1),\dots,s^{N+1}t_i - f_i(n+N+1),\dots,s^{N+1}t_m - f_m(n+N+1)\rangle.
\end{aligned}$$

By the termination condition, we have $s^{N+1}t_i \in \operatorname{Rad}\mathfrak{a}'$, and so the ideal in the last line must contain $(s^{N+1}t_i)^e$ for some $e \geq 0$. This forces $f_i(n+N+1) = 0$. ∎

Theorem 4.5 Algorithm 4.1 terminates. That is, for every input it will produce an output after a finite number of steps.

Proof For a polynomial ideal $\mathfrak{b} \trianglelefteq \mathbb{K}[X]$, whose set of associated prime ideals contains n_d ideals of dimension d ($d = 0,1,2,\dots$), define the vector

$$v(\mathfrak{b}) := (\dots,n_3,n_2,n_1,n_0).$$

For two ideals $\mathfrak{b} \trianglelefteq \mathbb{K}[X]$, $\mathfrak{b}' \trianglelefteq \mathbb{K}[X']$, we say $v(\mathfrak{b}) \prec v(\mathfrak{b}')$ if $v(\mathfrak{b})$ is lexicographically smaller than $v(\mathfrak{b}')$, i.e., if

$$v(\mathfrak{b}) = (\dots,n_3,n_2,n_1,n_0), \qquad v(\mathfrak{b}') = (\dots,n_3',n_2',n_1',n_0')$$

then $v(\mathfrak{b}) \prec v(\mathfrak{b}')$ iff $n_d < n_d'$ for the maximal d with $n_d \neq n_d'$. Observe that almost all entries n_j and n_j' and $v(\mathfrak{b}')$ are zero, so that such a d exists unless the vectors are identical.
Consider the sequence of ideals $\mathfrak{a}_N := \langle t_i,st_i,\dots,s^N t_i\rangle + \mathfrak{a} \cap \mathbb{K}\{t_1,\dots,t_m\}_{N+1}$ for $N = r, r+1, r+2,\dots$, where r is the order of S. Assume that $s^{N+1}t_i \notin \operatorname{Rad}\mathfrak{a}_N$ (otherwise there is nothing to prove). If $\mathfrak{p}_1,\dots,\mathfrak{p}_s$ are the associated prime ideals of $\mathfrak{a}_N$, then

$$s^{N+1}t_i \notin \operatorname{Rad}\mathfrak{a}_N = \bigcap_{j=1}^{s} \mathfrak{p}_j$$

implies that $s^{N+1}t_i \notin \mathfrak{p}_j$ for at least one j. For these indices j, we have $\dim(\mathfrak{p}_j + \langle s^{N+1}t_i\rangle) < \dim(\mathfrak{p}_j)$ (Shafarevich, 1972, Thm. I.6.1), so the inequality $\dim(\mathfrak{p}_j + \langle s^{N+1}t_i\rangle) \leq \dim(\mathfrak{p}_j)$, which holds for all j, is sharp for at least one j. It follows that

$$v(\mathfrak{a}_N + \langle s^{N+1}t_i\rangle) \prec v(\mathfrak{a}_N).$$

As S is normal, m-fold application of Lemma 3.14 asserts furthermore that $v(\mathfrak{a}_{N+1}) = v(\mathfrak{a} + \langle s^{N+1} t_i \rangle)$, so $v(\mathfrak{a}_{N+1}) \prec v(\mathfrak{a}_N)$. By Dickson's Lemma, an infinite sequence

$$v(\mathfrak{a}_N) \succ v(\mathfrak{a}_{N+1}) \succ v(\mathfrak{a}_{N+2}) \succ \cdots$$

cannot exist, and hence $s^{N+1} t_i \notin \operatorname{Rad} \mathfrak{a}_N$ cannot be true for all N. ■

Corollary 4.6 Algorithm 4.2 is correct and terminates. ■

4.2 Examples

Example 4.7 Let us illustrate the zero equivalence algorithm in full detail at the simple example sequence $f(n)$, defined by

$$f(n) = \frac{F_n}{F_{n+1}} + \sum_{k=1}^{n} \frac{(-1)^k}{F_k F_{k+1}} \qquad (n \geq 0),$$

where F_n denotes the n-th Fibonacci number (cf. p. 11). Using the closure properties of Section 3.2, we can easily form an admissible system for $f(n)$.

$$\begin{aligned} S = \{\, & f_1(n+2) = f_1(n+1) + f_1(n), \\ & f_2(n+1) = -f_2(n), \\ & f_3(n+1) = 1/f_1(n) f_1(n+1), \\ & f_4(n+1) = f_4(n) + f_2(n+1) f_3(n+1), \\ & f_5(n) = f_1(n)^2 f_3(n) + f_4(n) \,\}. \end{aligned}$$

We have $f(n) = f_5(n)$. The system S is already normal. Translating S to difference polynomials gives the associated difference ideal

$$\mathfrak{a} = \langle\langle s^2 t_1 - s t_1 - t_1, s t_2 + t_2, s t_3 t_1 s t_1 - 1, s t_4 - t_4 - s t_2 s t_3, t_5 - t_1^2 t_3 + t_4 \rangle\rangle \trianglelefteq \mathbb{K}\{t_1, \ldots, t_5\}.$$

First, we compute the number N of Algorithm 4.1. We have

$$\begin{aligned} & s^2 t_5 \notin \mathfrak{a} + \langle t_5, s t_5 \rangle \cap \mathbb{K}\{t_1, \ldots, t_5\}_2 \,, \\ & s^3 t_5 \notin \mathfrak{a} + \langle t_5, s t_5, s^2 t_5 \rangle \cap \mathbb{K}\{t_1, \ldots, t_5\}_3 \,, \\ & s^4 t_5 \in \mathfrak{a} + \langle t_5, s t_5, s^2 t_5, s^3 t_5 \rangle \cap \mathbb{K}\{t_1, \ldots, t_5\}_4 \,. \end{aligned}$$

Hence, $N = 3$. We conclude that

$$\forall\, n \in \mathbb{N} \colon f(n) = f(n+1) = f(n+2) = f(n+3) = 0 \implies f(n+4) = 0.$$

By evaluation we find that $f(1) = f(2) = f(3) = f(4) = 0$, and this gives $f(n) = 0$ for all $n \in \mathbb{N}$.

Example 4.8

(1) The number of iterations of the while loop in Algorithm 4.1 can be arbitrarily large. Choose a fixed number $M \in \mathbb{N}$ and consider the admissible sequence

$$f(n) := (n-1)(n-2)\cdots(n-M).$$

Then $f(n) = 0$ for $n = 1,2,\ldots,M$, but not for all $n \in \mathbb{N}$. Hence the number N computed by Algorithm 4.1 must be at least as large as M.

(2) The requirement that the admissible system S be made normal in Step 1 of Algorithm 4.1 is essential. Consider the simple hypergeometric identity

$$\sum_{k=0}^{n} \frac{1}{4^k}\binom{2k}{k} = \frac{2(n+1)}{4^{n+1}}\binom{2n+2}{n+1}.$$

The admissible system S, defined by

$$S = \Big\{ f_1(n+1) = f_1(n)+1, f_2(n+1) = 2\frac{2f_1(n)+1}{f_1(n)+1} f_2(n), f_3(n+1) = 4f_3(n),$$
$$f_4(n+1) = f_4(n) + \frac{f_2(n+1)}{f_3(n+1)}, f_5(n) = -f_4(n) + \frac{(f_1(n)+1)(2f_1(n)+1)f_2(n)}{(f_1(n)+1)f_3(n)} - 1 \Big\}$$

gives rise to the associated difference ideal

$$\mathfrak{a} = \langle\langle -st_1+t_1+1, -st_2(t_1+1)+2t_2(2t_1+1), -st_3+4t_3,$$
$$-st_4st_3+t_4st_3+st_2, -t_5(t_1+1)t_3-t_4(t_1+1)t_3+(t_1+1)t_2(2t_1+1)-(t_1+1)t_3\rangle\rangle.$$

If $\mathfrak{a}_2, \mathfrak{a}_3, \mathfrak{a}_4, \ldots$ are the associated polynomial ideals of order $2,3,4,\ldots$, then

$$\begin{aligned} st_5 &\notin \mathfrak{a}_2 + \langle t_5\rangle \\ s^2t_5 &\notin \mathfrak{a}_3 + \langle t_5, st_5\rangle \\ s^3t_5 &\notin \mathfrak{a}_4 + \langle t_5, st_5, s^2t_5\rangle, \ldots \end{aligned}$$

The algorithm would not terminate.

The reason for this phenomenon is that an ideal $\mathfrak{a} + \langle py-q\rangle \trianglelefteq \mathbb{K}[X,y]$ (with $\mathfrak{a} \trianglelefteq \mathbb{K}[X]$, $p,q \in \mathbb{K}[X]\setminus\mathfrak{a}$) may have more primary components than $\mathfrak{a}$ itself. In fact, every component of $\langle p,q\rangle$ gives rise to an additional component, and thus the component counting argument in the termination proof is violated.

The requirement that the admissible system be normal contains the requirement that p or q be constant, hence $\langle p,q\rangle = \langle 1\rangle$ and new components do not appear.

(3) Applying Algorithm 4.2 to the sequence $f(n)$ defined by

$$f(n+1) = \frac{1}{n-5} f(n) \quad (n \geq 1), \qquad f(1) = 0$$

yields True, disregarding that $f(n)$ is not well defined for $n \geq 6$.

Problems with rational function coefficients having poles in the range of interest are common phenomena in symbolic summation (Abramov and Petkovšek, 2005). We assume throughout that the admissible system, by which the involved sequences are defined, is chosen such that no denominators vanish for any $n \in \mathbb{N}$. (Compare our remarks on recurrences in Section 2.2.)

Example 4.9 The zero equivalence algorithm may be used for proving combinatorial identities, such as the following ones. To our knowledge, none of them was proven algorithmically before.

(1) $\sum_{k=0}^{n} \frac{1}{F_{2^k}} = 3 - \frac{F_{2^n-1}}{F_{2^n}}$ (Graham *et al.*, 1994, Exercise 6.61)

(2) $\prod_{k=1}^{n} (L_{2\cdot 3^k} - 1) = \frac{1}{2} F_{3^{n+1}}$ (Filipponi, 1996)

(3) The q-generalization

$$d(n)e(n+1) - d(n+1)e(n) = (-1)^n q^{\binom{n}{2}} \qquad (n \geq 0)$$

of Cassini's identity appearing in Andrews *et al.* (2000), where

$$\begin{aligned} d(n+2) &= d(n+1) + q^n d(n) \quad (n \geq 0), \qquad d(0) = 1, d(1) = 0 \\ e(n+2) &= e(n+1) + q^n e(n) \quad (n \geq 0), \qquad e(0) = 0, e(1) = 1 \end{aligned}$$

(4) For

$$h(n) := \frac{(n+1)\,((-1)^n(\pi+2) + \pi - 2)\,\Gamma(n/2)}{4\sqrt{\pi}\Gamma((n+1)/2)}$$

we have

$$\sum_{k=0}^{n} \frac{1}{k!} = \mathop{\mathrm{K}}_{k=1}^{n} \Big(h(k) - \prod_{i=1}^{k} (-\tfrac{7}{4} i^2 + 9i - \tfrac{45}{4}) \Big) \qquad (n \geq 1)$$

(Kauers, 2003).

(5) Consider the exponential integral $E_n(x)$ and the incomplete Gamma function $\Gamma(n,x)$ defined via

$$E_n(x) := \int_1^\infty t^{-n} \exp(-xt)\, dt \qquad \Gamma(n,x) := \int_x^\infty t^{n-1} \exp(-t)\, dt.$$

According to Abramowitz and Stegun (1972), these function satisfy the recurrences

$$E_{n+1}(x) = \frac{1}{n}(\exp(-x) - xE_n(x)), \qquad \Gamma(n,x) = (n-1)\Gamma(n-1,x) + x^{n-1}\exp(-x),$$

thus they are admissible as sequences in n. The identity

$$E_n(x) = x^{n-1}\Gamma(1-n,x)$$

can be proven by the algorithm.

(6) Consider the Somos sequence C_n defined by

$$C_{n+2} = \frac{1}{C_{n-2}}(C_{n-1}C_{n+1} + C_n^2) \quad (n \geq 2), \qquad C_{-2} = C_{-1} = C_0 = C_1 = 1.$$

The identities

$$\begin{aligned} C_{n+3}C_{n-2} &= -C_{n-1}C_{n+2} + 5C_nC_{n+1} \\ C_{n+3}C_{n-3} &= C_{n-1}C_{n+1} + 5C_n^2 \\ C_{n+4}C_{n-4} &= 25C_{n+1}C_{n-1} - 4C_n^2 \end{aligned}$$

by van der Poorten (2004) can be verified by the algorithm.

4.3 Two Non-Admissibility Criteria

The fact that Algorithm 4.1 terminates for all admissible sequences shades some light on the extent of this class. We have already seen in Section 3.2 that the class of admissible sequences is quite large. In the present section, we state two simple sufficient criteria for a sequence *not* to be admissible.

As in the proof of the termination theorem (Theorem 4.5), it can be shown that the sequence

$$\dim \mathfrak{a}_1, \dim \mathfrak{a}_2, \dim \mathfrak{a}_3, \dots$$

where $\mathfrak{a}_i$ are associated polynomial ideals of some normal admissible system S is ultimately constant, where $\dim \mathfrak{a}_i$ is of course understood with respect to $\mathbb{K}\{t_1,\dots,t_m\}_i$. Now, consider the elimination ideals $\mathfrak{a}'_r := \mathfrak{a}_r \cap \mathbb{K}[t_i, st_i, \dots, s^r t_i]$. The dimension of $\mathfrak{a}_r$ is bounded by, say, d, whereas the dimension of $\mathbb{K}[t_i,\dots,s^r t_i]$ is $r+1$. As soon as $r > d$, the ideal $\mathfrak{a}'_r$ cannot be empty because, geometrically speaking, the projection of a variety into a higher dimensional affine space cannot be surjective (Shafarevich, 1972). Thus we have proven the following proposition.

Proposition 4.10 If $f(n)$ is admissible, then there exists a number $r \in \mathbb{N}$ and a polynomial $p \in \mathbb{K}[x_0,\dots,x_r] \setminus \{0\}$ such that

$$p\,(f(n), f(n+1)\dots, f(n+r)) = 0 \qquad (n \geq 0).$$

Given an admissible system for $f(n)$, such a polynomial can be computed. ■

This proposition admits a criterion for proving that certain sequences are *not* admissible.

Example 4.11 The sequence $f(n) = 2^{n!}$ is not admissible. Assume otherwise. Then, by the proposition, there would exist a polynomial $p \in \mathbb{Q}[x_0,\dots,x_r]$ such that

$$p\,(f(n), f(n+1), \dots, f(n+r)) = 0 \qquad (n \geq 0).$$

For $e_0,\dots,e_s \in \mathbb{Z}$ with $e_s \neq 0$, we have

$$(2^{(n+s)!})^{e_s} \cdots (2^{(n+1)!})^{e_1} (2^{n!})^{e_0} = 2^{(e_s n^s + \text{lower terms})n!} \xrightarrow{n\to\infty} \begin{cases} 0 & \text{if } e_s < 0 \\ \infty & \text{if } e_s > 0. \end{cases}$$

Hence, if we divide p by its leading term w.r.t. the lexicographic order $x_r > x_{r-1} > \cdots > x_0$, substitute the $f(n+i)$ and let n go to infinity, we arrive at the contradiction $\textsc{Lc}(p) = 0$.

The termination of Algorithm 4.1 also implies that a sequence $f(n)$, which is not ultimately constant, can only be admissible if it does not possess arbitrarily long "runs" of constant values. This means: A sequence $f(n)$ with the property that for every length M there exists a number $N > M$ and a number $n_0 \in \mathbb{N}$ such that

$$f(n_0) = f(n_0+1) = f(n_0+2) = \cdots = f(n_0+N) \neq f(n_0+N+1)$$

is not admissible. For every admissible sequence $f(n)$ and every value $v \in \mathbb{K}$, Algorithm 4.1 applied to $f(n) - v$ delivers an upper bound for the length of a run

$$f(n_0) = f(n_0+1) = f(n_0+2) = \cdots = f(n_0+N-1) = v.$$

Any longer run would imply that $f(n) = v$ for all $n \geq n_0$.

As an example, we find that $(-1)^{\lfloor \log n \rfloor}$ is not admissible, because this sequence consists of runs of -1 and 1 that become longer and longer as n grows.

4.4 Remarks on Complexity

We have no results about the time and space complexity of Algorithm 4.1, and just make some general remarks in this section. A complexity analysis would have to focus on two questions: First, how much time is consumed by the radical membership test in line 4, and secondly, how many iterations of the while loop might be necessary.

It is generally hard to make statements about the time complexity of algorithms in commutative algebra. It was mentioned that the radical membership test can be done by a standard application of Gröbner basis techniques, but the computation of Gröbner bases is known to be expensive: doubly exponential runtime and exponential space requirements have to be assumed in general for the worst case. The special problem of radical membership can, however, be decided with polynomial space Mayr (1997).

As for the number of iterations, it is not likely that a reasonable bound depending on, say, the depth m, the order r and/or the maximum total degree d of the admissible system S could be established. In fact, any such bound $\kappa(m,r,d)$ would presumably give rise to a much faster algorithm, provided that κ itself can be computed in reasonable time: just return $N := \kappa(m,r,d)$. The while loop with its expensive radical membership test could be discarded altogether.

Despite the poor worst case complexity, it turns out that Algorithm 4.1 performs quite well in practice. It is well known that Gröbner basis computations perform far better than suggested by the worst case complexity analysis on problems arising in practice. A similar remark applies to the number of iterations of the while loop. Though it is easy to construct input which requires any prescribed number of iterations (Example 4.8.(1)), already two or three iterations suffice for most examples from practice. If special purpose software is used most identities to which the algorithm is applicable, in particular those listed in Example 4.9, can be done in less than a few seconds on contemporary hardware. The use of Faugère's system Gb (Faugère, 1999, 2002a,b) has turned out to be very efficient.

It is possible to improve the efficiency of the algorithm in the case where algebraic dependencies are known in addition to the defining recurrences. The most typical example is $t^2 - 1$ if t represents $(-1)^n$. In Chapters 6f we will discuss methods for computing algebraic dependencies among given admissible sequences. Once and for all computed, such relations may be freely added to the associated difference ideal $\mathfrak{a}$ in Algorithm 4.1. Correctness is obviously not hurt if the relations indeed hold. Also the termination is not affected, because if the algorithm terminates for an ideal $\mathfrak{a}$ then it certainly also terminates for every overideal $\mathfrak{a}' \supseteq \mathfrak{a}$ in place of $\mathfrak{a}$.

Adding relations might seem counterproductive, because the radical membership tests might slow down if additional nonlinear generators are present. However, the number N of iterations actually needed might be considerably smaller.

Example 4.12 In order to prove the Fibonacci identity

$$F_n^5(1 - F_{n+1}) + F_n^4(2 + 2F_{n+1} - 4F_{n+1}^2) - \cdots$$

$$\begin{aligned}
&\cdots - F_n^3(5-9F_{n+1}+4F_{n+1}^3) - F_n^2(6-8F_{n+1}^2+3F_{n+1}^3-F_{n+1}^4)\\
&\quad - F_nF_{n+1}(6-10F_{n+1}+4F_{n+1}^2+2F_{n+1}^3-2F_{n+1}^4) + F_{n+1}^2(6-5F_{n+1}-3F_{n+1}^2+2F_{n+1}^3)\\
&= (-1)^n\big(F_n^3(F_{n+1}-1) + F_n^2(3F_{n+1}^2-F_{n+1}-2) + F_n(2F_{n+1}^3-7F_{n+1}-5)\\
&\qquad + (2F_{n+1}^3-3F_{n+1}^2-5F_{n+1}+6)\big)
\end{aligned}$$

directly, Algorithm 4.2 requires four iterations. The identity was obtained by disturbing Cassini's identity with a multiplicative factor. (As we will see in Chapter 7, all Fibonacci identities are obtained in this way.) If Cassini's identity is taken for granted and the corresponding difference polynomial is added to the ideal in line 4 of Algorithm 4.1, then no iteration is needed at all. For proving Cassini's identity, one single iteration is necessary.

A speed up by homomorphic images is not so easy to achieve. The essence of a radical membership test is to decide whether $\mathfrak{a} = \langle 1\rangle$ for a certain ideal $\mathfrak{a}$. For a given ideal $\mathfrak{a} = \langle a_1,\dots,a_m\rangle \trianglelefteq \mathbb{Q}[x_1,\dots,x_n]$, let $\bar{\mathfrak{a}} = \langle \bar{a}_1,\dots,\bar{a}_m\rangle \trianglelefteq \mathbb{Z}_p[x_1,\dots,x_n]$ be the ideal generated by the images $\bar{a}_i$ of the a_i under the canonical homomorphism $\mathbb{Q}\to\mathbb{Z}_p$ (p prime). It is easy to find examples where $\mathfrak{a} = \langle 1\rangle$, but $\bar{\mathfrak{a}} \neq \langle 1\rangle$, and conversely, where $\mathfrak{a} \neq \langle 1\rangle$ but $\bar{\mathfrak{a}} = \langle 1\rangle$. Hence, from the result of a radical membership in a homomorphic image, we cannot draw conclusions about the radical membership in the original field.

With high probability, however, we have $\mathfrak{a} = \langle 1\rangle$ and $\bar{\mathfrak{a}} = \langle 1\rangle$ or $\mathfrak{a} \neq \langle 1\rangle$ and $\bar{\mathfrak{a}} \neq \langle 1\rangle$ (Arnold, 2003). This suggests to first run the radical membership test modulo a prime p. If it returns False, iterate. Only if it returns True, check the result by recomputing the test in $\mathbb{Q}$.

4.5 Proof Certificates

Results obtained by computer algebra software might be incorrect. Even though the underlying algorithms are provably correct, implementations consisting of several thousand lines of code are likely to contain bugs. Wester (1999) studies the behavior of computer algebra systems and comes to the conclusion that it is indispensable to check the result of a nontrivial computation for plausibility.

The output of a decision procedure is just "yes" or "no," and there is little possibility to check such a result for plausibility. In view of the possibility of bugs we might ask what is the practical value of a proving procedure. For hypergeometric identities, Wilf and Zeilberger (1990) devised the concept of *proof certificates.* Instead of just saying "yes" or "no," the proving algorithm supplies a formal proof of the claim which can be verified independently of the algorithm by simple arithmetic.

Our zero equivalence test (Algorithm 4.2) can be adapted in such a way that it returns a proof along with its output. For the case where $f(n)\neq 0$, this is easy. The algorithm has found a witness n_0 which may serve as certificate. In the case $f(n) = 0$ we have to justify the induction step property for the number N returned by Algorithm 4.1, and for this it is evidently sufficient to justify the radical membership test in the final iteration.

In order to justify $p \in \operatorname{Rad}\mathfrak{a}$ for a given $p\in\mathbb{K}[X]$ and $\mathfrak{a} = \langle p_1,\dots,p_m\rangle \trianglelefteq \mathbb{K}[X]$, recall that $p\in\operatorname{Rad}\mathfrak{a}$ iff $p^n\in\mathfrak{a}$ for some n, by definition of the radical. In this case, there exist cofactors

$q_1, \dots, q_m$ such that

$$p^n = q_1 p_1 + \dots + q_m p_m.$$

The exponent n as well as the cofactors q_i can be computed, and from this data a proof of the zero equivalence by induction can be constructed, as shown in the following simple example.

Example 4.13 Consider Cassini's identity

$$F_n^2 + F_n F_{n+1} - F_{n+1}^2 + (-1)^n = 0 \qquad (n \geq 0).$$

We have

$$s^2 t_3 \in \operatorname{Rad}(\langle t_3, st_3 \rangle + \langle\langle s^2 t_1 - st_1 - t_1, t_2^2 - 1, st_3 - t_1^2 - t_1 st_1 + st_1^2 - t_2 \rangle\rangle \cap \mathbb{K}\{t_1, t_2, t_3\}_2),$$

and so the identity is proved by checking two initial conditions. The radical membership is asserted by the polynomial identity

$$\begin{aligned}(s^2 t_3)^1 = {} & (-t_1 - s^2 t_1) \cdot (s^2 t_1 - st_1 - t_1) \\ & + 1 \cdot (st_2 + t_2) \\ & + (-1) \cdot (st_3) \\ & + 1 \cdot (st_3 - t_1^2 - t_1 st_1 + st_1^2 - t_2) \\ & + 1 \cdot (s^2 t_3 - st_1^2 - st_1 s^2 t_1 + s^2 t_1^2 - st_2),\end{aligned}$$

which is easily verified.

The computation of the cofactors is rather costly. For nontrivial examples, it will most often be the case that Algorithm 4.1 terminates after a reasonable time, but the extra effort required for computing the proof certificate is beyond the scope of state-of-the-art computing environments.

4.6 Zero Equivalence of Non-Admissible Sequences

In this section, we present a generalization of the zero equivalence test which can be applied to certain problems involving sequences that are not admissible (Kauers, 2004). We have met examples of such sequences in Section 4.3.

Let $g_1(n), \dots, g_\ell(n)$ be arbitrary (not necessarily admissible) sequences in $\mathbb{K}$, define

$$\varphi \colon \mathbb{K}\{t_1, \dots, t_\ell\} \to \mathbb{K}^{\mathbb{N}}$$

by $\varphi(x) = x$ ($x \in \mathbb{K}$) and $\varphi(t_i) = g_i(n)$ ($i = 1, \dots, \ell$). We assume that $g_1(n), \dots, g_\ell(n)$ are such that membership $p \overset{?}{\in} \ker \varphi \trianglelefteq \mathbb{K}\{t_1, \dots, t_m\}$ can be decided for any given $p \in \mathbb{K}\{t_1, \dots, t_m\}$.

For the sake of theoretic soundness, we assume further that $g_1(n), \dots, g_\ell(n)$ are chosen such that for every $p \in \mathbb{K}\{t_1, \dots, t_m\} \setminus \ker \varphi$, the sequence $\varphi(p)$ has finitely many roots only, and an algorithm is known for computing an upper bound for the largest root. This is a severe restriction, but we may well ignore it in practice and prompt the user to supply the required information instead of insisting in answering such questions algorithmically.

Definition 4.14 Let $g_1(n),\ldots,g_\ell(n)$ be arbitrary sequences in $\mathbb{K}$. A system $S=\{rec_1,\ldots,rec_m\}$ of difference equations for sequences $f_1(n),\ldots,f_m(n)$ is *admissible* w.r.t. $g_1(n),\ldots,g_\ell(n)$ if each rec_i is of the form

$$\begin{aligned}
f_i(n+r_i) = \mathrm{rat}_i\big(&g_1(n), \quad g_1(n+1), \quad \ldots, g_1(n+r_i-1), \quad g_1(n+r_i), \quad g_1(n+r_i+1), \ \ldots,\\
&\vdots \\
&g_\ell(n), \quad g_\ell(n+1), \quad \ldots, g_1(n+r_i-1), \quad g_\ell(n+r_i), \quad g_\ell(n+r_i+1), \ \ldots,\\
&f_1(n), \quad f_1(n+1), \quad \ldots, f_1(n+r_i-1), \quad f_1(n+r_i),\\
&\vdots \\
&f_{i-1}(n), f_{i-1}(n+1), \ \ldots, f_{i-1}(n+r_i-1), f_{i-1}(n+r_i),\\
&f_i(n), \quad f_i(n+1), \quad \ldots, f_i(n+r_i-1),\\
&\vdots \\
&f_m(n), \quad f_m(n+1), \quad \ldots, f_m(n+r_i-1)\big) \qquad\qquad (n\ge 1)
\end{aligned}$$

where $r_i\ge 0$ is fixed and rat_i is some fixed rational function with coefficients in $\mathbb{K}$.

A sequence $f(n)$ in $\mathbb{K}$ is called *admissible* w.r.t. $g_1(n),\ldots,g_\ell(n)$ if there exists an admissible system of recurrences $S=\{rec_1,\ldots,rec_m\}$ w.r.t. $g_1(n),\ldots,g_\ell(n)$ and solutions $f_1(n),\ldots,f_m(n)$ of S with $f(n)=f_i(n)$ for some i.

Note that the $g_i(n)$ may appear with arbitrary order in the recurrences.

The original definition of admissible systems and sequences is contained in this definition as the special case $\ell=0$. We leave it to the reader to verify that Theorems 3.2 and 3.5 can be generalized to the present situation.

Let S be an admissible system for $f_1(n),\ldots,f_m(n)$ w.r.t. $g_1(n),\ldots,g_\ell(n)$. The *associated difference ideal* $\mathfrak{a}\trianglelefteq\mathbb{K}\{t_1,\ldots,t_{m+\ell}\}$ of S is defined by $\mathfrak{a}:=\langle P_1,\ldots,P_m\rangle$, where $P_1,\ldots,P_m$ are as in Definition 3.12 (with s^it_j replaced by $s^it_{j+\ell}$ for all i and j). In order to generalize the definition of the associated polynomial ideal, we introduce the rings

$$\begin{aligned}
\mathbb{K}\{t_1,\ldots,t_{m+\ell}\}_r^\ell := \mathbb{K}\big[&t_1, st_1, \ldots, \qquad s^rt_1, s^{r+1}t_1, \ldots\ldots,\\
&\vdots \qquad\qquad\quad \vdots\\
&t_\ell, st_\ell, \ldots, \qquad s^rt_\ell, s^{r+1}t_\ell, \ldots\ldots,\\
&t_{\ell+1}, st_{\ell+1}, \ldots, \ s^rt_{\ell+1},\\
&\vdots \qquad\qquad\quad \vdots\\
&t_{m+\ell}, st_{m+\ell}, \ldots, s^rt_{m+\ell}\big].
\end{aligned}$$

The associated polynomial ideal $\mathfrak{a}$ of order r is defined as

$$\langle P_1, sP_1, \ldots, s^{r-r_1}P_1, P_2, sP_2, \ldots, s^{r-r_2}P_2, \ldots\ldots, P_m, sP_m, \ldots, s^{r-r_m}P_m\rangle \trianglelefteq \mathbb{K}\{t_1,\ldots,t_{m+\ell}\}_r^\ell$$

where the P_i are as above and r_i is the order of P_i (cf. Definition 3.13). Theorems 3.15 and 3.16 carry over to the present situation.

We can now give the following zero equivalence testing algorithm for sequences which are admissible w.r.t. some given sequences $g_1(n),\dots,g_\ell(n)$. The basic idea is the same as for Algorithms 4.1 and 4.2, only the technicalities are a bit more involved.

Algorithm 4.15 (Zero Equivalence of Admissible Sequences w.r.t. $g_1(n),\dots,g_\ell(n)$)
Input: An admissible system S for $f_1(n),\dots,f_m(n)$ w.r.t. $g_1(n),\dots,g_\ell(n)$; initial values; an index $i \in \{1,\dots,m\}$
Output: True, False, or Failed
Assumption: For every $p \in \mathbb{K}\{t_1,\dots,t_m\}$, it can be decided whether $p \in \ker\varphi$ and an upper bound for the maximum root $M \in \mathbb{N}$ of $\varphi(p)$ can be computed, where $\varphi\colon \mathbb{K}\{t_1,\dots,t_m\} \to \mathbb{K}^{\mathbb{N}}$ is defined via $t_i \mapsto g_i(n)$. It is further assumed that the $g_i(n)$ are computable.

1 **if** $i \leq \ell$ **then**
2   **if** $t_i \in \ker\varphi$ **then return** True **else return** False
3 Bring S into the form of Theorem 3.2.(3)
4 Let $\mathfrak{a} \trianglelefteq \mathbb{K}\{t_1,\dots,t_{m+\ell}\}$ be the associated difference ideal of S
5 Set N to the order of S
6 **if not** $f_i(1) = f_i(2) = \dots = f_i(N-1) = 0$ **then**
7   **return** False
8 **repeat**
9   **if** $f_i(N) \neq 0$ **then**
10   **return** False
11   Let X be a new indeterminate.
12    $\mathfrak{a}_0 := (\langle t_i, st_i, \dots, s^{N-1}t_i, Xs^N t_i - 1\rangle + \mathfrak{a} \cap \mathbb{K}\{t_1,\dots,t_{m+\ell}\}_N^\ell) \cap \mathbb{K}\{t_1,\dots,t_\ell\}_N^\ell$
13    $N := N+1$
14 **while** $\mathfrak{a}_0 = \{0\}$
15 **if** $\mathfrak{a}_0 \subseteq \ker\varphi$ **then**
16   **return** Failed
17 Let $p \in \mathfrak{a}_0 \setminus \ker\varphi$
18 Let M be an upper bound for the roots of $\varphi(p)$
19 **if** $f_i(N) = f_i(N+1) = \dots = f_i(M) = f_i(M+1) = 0$ **then**
20   **return** True
21 **else**
22   **return** False

The steps in this algorithm are computable. The elimination ideals $\mathfrak{a} \cap \mathbb{K}\{t_1,\dots,t_{m+\ell}\}_N^\ell$ are precisely the associated polynomial ideals by the generalized Theorem 3.16. The elimination of X in line 12 can be done by Gröbner bases. It does not matter that the polynomial rings $\mathbb{K}\{t_1,\dots,t_{m+\ell}\}_N^\ell$ have infinitely many indeterminates if $\ell > 0$, because the ideal at hand is finitely generated and only the indeterminates appearing in the finite basis have to be taken into account. The result of the elimination is a finite basis for $\mathfrak{a}_0$. Whether we have $\mathfrak{a}_0 \subseteq \ker\varphi$ can be decided by checking whether $b \in \ker\varphi$ for all b in the basis of $\mathfrak{a}_0$. It is assumed that this can be done. If line 17 is reached, then $b \notin \ker\varphi$ for at least one basis element b of $\mathfrak{a}_0$. This b is a suitable choice for p. Line 18 is computable by assumption, and the remaining lines only require evaluation of $f_i(n)$ at particular points $n \in \mathbb{N}$ and zero equivalence decision in $\mathbb{K}$.

Theorem 4.16 Algorithm 4.15 is correct. That is, if it returns True then $f_i(n) = 0$ for all $n \geq 1$, and if it returns False then $f_i(n) \neq 0$ for at least one $n \in \mathbb{N}$. No statement is made about the case when the return value is Failed.

Proof The algorithm returns False only if it encounters a counterexample, so it remains to prove correctness for the case where the algorithm returns True. Suppose now that this is the case.

For every $n, N \in \mathbb{N}$, we have

$$\begin{aligned}&\mathfrak{a} + \langle t_{\ell+1} - f_1(n), \dots, s^N t_{\ell+1} - f_1(n+N), \dots\dots, t_{\ell+m} - f_m(n), \dots, s^N t_{\ell+m} - f_m(n+N)\rangle \\ &\qquad \cap \mathbb{K}\{t_1, \dots, t_\ell\} \\ &\subseteq \{p \in \mathbb{K}\{t_1, \dots, t_\ell\} : \varphi(p)(n) = 0\}\end{aligned} \tag{4.1}$$

Let $N_0 \in \mathbb{N}$ be the value of N at the termination of the loop in lines 8–14. We show that for all $n > M$ (where M is as in line 19),

$$f_i(n) = \cdots = f_i(n + N_0 - 1) \implies f_i(n + N_0) = 0. \tag{4.2}$$

Assume, for the contrary, that there exists an $n > M$ such that

$$f_i(n) = f_i(n+1) = \cdots = f_i(n + N_0 - 1) = 0 \quad \text{and} \quad f_i(n + N_0) \neq 0.$$

By (4.1) and line 12,

$$\mathfrak{a}_0 \subseteq \{p \in \mathbb{K}\{t_1, \dots, t_\ell\} : \varphi(p)(n) = 0\}.$$

Hence, for the $p \in \mathfrak{a}_0$ of line 17, we have $p(n) = 0$ in contradiction to the choice of M (line 18), and the proof of (4.2) is complete.

Using this implication and induction step, and line 19 as induction base, it follows by complete induction on n that $f_i(n) = 0$ for all $n > M$. In addition, by the tests in lines 6, 9 and 19, we have $f_i(n) = 0$ for $n = 1, \dots, M$, and thus $f_i(n) = 0$ for all $n \geq 1$, as claimed. ∎

Theorem 4.17 Algorithm 4.15 terminates.

Proof The only critical part is the loop in lines 8–14. Let $\bar{\mathbb{K}} := \mathbb{K}\langle t_1, \dots, t_\ell\rangle = Q(\mathbb{K}\{t_1, \dots, t_\ell\})$ and $\bar{\mathfrak{a}}$ be the image of $\mathfrak{a}$ under the canonical embedding $\mathbb{K} \hookrightarrow \bar{\mathbb{K}}$. $\bar{\mathfrak{a}}$ may be seen as the associated difference ideal of some admissible system $\bar{S}$ for sequences $f_1(n), \dots, f_m(n)$ in $\bar{\mathbb{K}}$. Consider the application of Algorithm 4.1 to $\bar{S}$ and i.

According to Theorem 4.5, there exists a finite number $N \in \mathbb{N}$ such that

$$s^{N+1} t_i \in \operatorname{Rad}(\langle t_i, \dots, s^N t_i\rangle + \bar{\mathfrak{a}} \cap \bar{\mathbb{K}}\{t_{\ell+1}, \dots, t_{\ell+m}\}_{N+1}).$$

By Theorem 2.7.(2), this is equivalent to

$$\langle t_1, \dots, s^N t_i, X s^{N+1} t_i - 1\rangle + \bar{\mathfrak{a}} \cap \bar{\mathbb{K}}\{t_{\ell+1}, \dots, t_{\ell+m}\}_{N+1} = \langle 1\rangle,$$

which in turn is equivalent to

$$\big(\langle t_1, \dots, s^N t_i, X s^{N+1} t_i - 1\rangle + \mathfrak{a} \cap \mathbb{K}\{t_1, \dots, t_\ell\}\{t_{\ell+1}, \dots, t_{\ell+m}\}_{N+1}\big) \cap \mathbb{K}\{t_1, \dots, t_\ell\} \neq \{0\}. \quad ∎$$

Algorithm 4.15 is useful for proving identities involving expressions which do not give rise to admissible sequences.

Example 4.18

(1) The identity

$$\sum_{k=0}^{n} \frac{1}{\sqrt{k+1}+\sqrt{k}} = \sqrt{n+1} \qquad (n \geq 0)$$

can be shown by Algorithm 4.15. Take $g_1(n) = n$, $g_2(n) = \sqrt{n}$. Then, by a result of Gerhold (2004),

$$\ker \varphi = \langle\langle st_1 - t_1 - 1, t_2^2 - t_1 \rangle\rangle$$

and membership for this ideal can be easily checked. We apply the algorithm to the admissible system

$$S = \Big\{ f_1(n+1) = \frac{1}{g_2(n+1)+g_2(n)}, f_2(n+1) = f_2(n) + f_1(n+1),$$
$$f_3(n+1) = f_2(n) - g_2(n+1) \Big\}$$

for $f_3(n)$. The loop in lines 8–14 terminates with $\mathfrak{a}_0 = \langle s^2t_2st_2 + s^2t_2t_2 - st_2^2 - st_2t_2 - 1 \rangle$ when $N = 3$. By a procedure to be introduced in the following chapter, it can be quickly asserted that

$$\sqrt{n+2}\sqrt{n+1} + \sqrt{n+2}\sqrt{n} - (n+1) - \sqrt{n+1}\sqrt{n} - 1 < 0$$

for all $n \geq 0$, in particular the left hand does not evaluate to zero for any natural n. Hence we may choose $M = 0$ in line 18. Then no further checks are necessary in line 19 and the algorithm returns True.

(2) An unlucky choice of the defining admissible system might lead to a failure of the algorithm. Consider again the identity

$$\sum_{k=0}^{n} \frac{1}{\sqrt{k+1}+\sqrt{k}} = \sqrt{n+1} \qquad (n \geq 0),$$

now using the admissible system

$$S = \big\{ f_1(n) = 1/(g_2(n+1)+g_2(n)), f_2(n+1) = f_2(n) + f_1(n+1),$$
$$f_3(n) = f_2(n) - g_2(n+1) \big\}$$

with $g_1(n)$ and $g_2(n)$ as above. Then the loop in lines 8–14 terminates with $\mathfrak{a}_0 = \langle st_1 - st_2^2 \rangle \subseteq \ker \varphi$.

(3) We have seen in Example 4.11 that the shifts of $g_1(n) := 2^{n!}$ do not satisfy an algebraic relation. That is, for

$$\varphi\colon \mathbb{Q}\{t\} \to \mathbb{Q}^{\mathbb{N}}, \qquad t \mapsto 2^{n!}$$

we have $\ker \varphi = \{0\}$. With this knowledge, we may prove the identity

$$\sum_{k=1}^{n} \frac{1}{2^{k!}} \prod_{i=1}^{k} \frac{2^{i!}}{1+2^{i!}} = 1 - \prod_{k=1}^{n+1} \frac{2^{k!}}{1+2^{k!}} \qquad (n \geq 0)$$

by means of Algorithm 4.15. Using the admissible system

$$\Big\{\, f_1(n+1) = \frac{g_1(n+1)}{1+g_1(n+1)} f_1(n), f_2(n+1) = f_2(n) + \frac{f_1(n+1)}{g_1(n+1)},$$
$$f_3(n+1) = f_2(n) - (1 - f_1(n+1)) \,\Big\}$$

the loop in lines 8–14 terminates with $\mathfrak{a}_0 = \langle 1 \rangle$ when $N = 1$. If we choose $p = 1$ in line 17, then we may choose $M = 0$ in line 18.

In the case of $\ker \varphi = \{0\}$ in the last example above, where there are no polynomial relations available for the sequence $2^{n!}$, one might ask why the algorithm is able to prove something about it. In fact, the appearance of $2^{n!}$ is immaterial in this identity, we have

$$\sum_{k=1}^{n} \frac{1}{x_k} \prod_{i=1}^{k} \frac{x_i}{a+x_i} = \frac{1}{a}\Big(1 - \prod_{k=1}^{n+1} \frac{x_k}{a+x_k}\Big) \qquad (n \geq 0, a \in \mathbb{K} \setminus \{0\})$$

for *every* sequence $x_1, x_2, \ldots \in \mathbb{K} \setminus \{-a\}$ (van der Poorten, 1979). Let us consider such general identities in some more detail.

Consider the field $\mathbb{K}' := \mathbb{K}(x_1, x_2, x_3, \ldots)$ and define

$$\varphi \colon \mathbb{K}\{t\} \to \mathbb{K}(x_1, x_2, x_3, \ldots)^{\mathbb{N}}$$

by $\varphi(t) := (x_1, x_2, x_3, \ldots)$. Such a sequence is called *free* over $\mathbb{K}$. By construction we have $\ker \varphi = \{0\}$. We can thus apply Algorithm 4.15 to identities involving sequences which are admissible with respect to a sequence $x_1, x_2, x_3, \ldots$ of indeterminates. Observe that as a consequence of $\ker \varphi = \{0\}$, the algorithm will never end up in a failure. If the x_i are understood as formal variables rather than as placeholders of some particular sequences, we may well choose $M = 0$ in line 18. We restrict here our attention to the question of proving conjectured identities involving free sequences. For finding such identities, we were able to give a generalization of Karr's algorithm (Karr, 1981, 1985) that allows free sequences to appear in the summand expression in a collaboration with C. Schneider (Kauers and Schneider, 2004). We will not comment on this in the present text.

One source of identities involving free sequences is the theory of *symmetric functions* (Stanley, 1999). A function $f(x_1, \ldots, x_n)$ is called symmetric if

$$\forall\, \pi \in S_n : f(x_1, \ldots, x_n) = f(x_{\pi 1}, \ldots, x_{\pi n}).$$

If, in addition, $f(x_1, \ldots, x_n) \in \mathbb{K}[x_1, \ldots, x_n]$, we call it a *symmetric polynomial.* There are special purpose algorithms available for the treatment of symmetric polynomials.

Example 4.19 Consider the identity

$$\Big(\sum_{k=1}^{n} x_i\Big)^2 = \sum_{k=1}^{n} x_k^2 + 2 \sum_{k=1}^{n} x_k \sum_{i=1}^{k-1} x_i, \tag{4.3}$$

which holds for all $n \geq 0$ and all $x_1, x_2, \ldots, x_n \in \mathbb{K}$. Using the standard definitions

$$p_1(n) = \sum_{k=1}^{n} x_k, \quad p_2(n) = \sum_{k=1}^{n} x_k^2, \quad e_2(n) = \sum_{k=1}^{n} x_k \sum_{i=1}^{k-1} x_i$$

the identity (4.3) may be rewritten as $p_1(n)^2 = p_2(n) + 2e_2(n)$. Stembridge (1995) describes a Maple package for doing computations with symmetric function identities of this form. The above identity can be quickly proven with his package.

Identity (4.3) can also be proven by an application of Algorithm 4.15. A suitable admissible system defining the quantities $p_1(n)$, $p_2(n)$, and $e_2(n)$ is given by

$$\{\, p_1(n+1) = p_1(n) + x_{n+1},\ p_2(n+1) = p_2(n) + x_{n+1}^2,\ e_2(n+1) = e_2(n) + p_1(n)\,\}.$$

There are several identities in the literature which are out of the scope of algorithms for symmetric functions, but to which our algorithm applies as well.

Example 4.20

(1) $\displaystyle\sum_{k=1}^{n} \frac{\prod_{i=1}^{k-1}(x_i+\alpha)}{\prod_{i=1}^{k} x_i} = \frac{1}{\alpha}\Big(\prod_{k=1}^{n} \frac{x_k+\alpha}{x_k} - 1\Big)$ (Gosper, 1978)

(2) The theorem on summation by parts (Graham *et al.*, 1994),

$$\sum_{k=1}^{n-1} x_k \Delta y_k = x_n y_n - x_1 y_1 - \sum_{k=1}^{n-1} y_{k+1} \Delta x_k,$$

where $\Delta f(n) := f(n+1) - f(n)$ denotes the forward difference operator.

(3) The Christoffel-Darboux identity (Chihara, 1978, Theorems 4.5 and 4.6): Let $\lambda(n)$ and $c(n)$ be arbitrary sequences ($\lambda(n) \neq 0$ for all n) and define sequence $p_n(x)$ of polynomials via

$$p_n(x) = (x - c(n))p_{n-1}(x) - \lambda(n)p_{n-2}(x) \quad (n \geq 0), \qquad p_{-1}(x) = 0, p_0(x) = 1.$$

Then

$$\sum_{k=0}^{n} \frac{p_k(x)p_k(u)}{\prod_{i=1}^{k+1}\lambda(i)} = \frac{p_{n+1}(x)p_n(u) - p_n(x)p_{n+1}(u)}{(x-u)\prod_{k=1}^{n+1}\lambda(k)},$$
$$\sum_{k=0}^{n} \frac{p_k(x)^2}{\prod_{i=1}^{k+1}\lambda(i)} = \frac{p_n(x)p'_{n+1}(x) - p_{n+1}(x)p'_n(x)}{\prod_{k=1}^{n+1}\lambda(k)}$$

(4) (Andrews *et al.*, 1999, Exercise 5.12) Consider the general continued fraction

$$\operatorname*{K}_{k=1}^{n}(b(k)/a(k)) = \frac{p(n)}{q(n)}$$

with $p(n)$ and $q(n)$ being the corresponding continuant polynomials (cf. p. 12). Then

$$\operatorname*{K}_{k=1}^{n}(b(k)/a(k)) = a(0) + \sum_{k=1}^{n}(-1)^{k+1}\frac{\prod_{i=1}^{k} b(i)}{q(k)q(k+1)} \qquad (n \geq 1).$$

(5) (Bowman and Laughlin, 2002; Chrystal, 1922) For arbitrary sequences $a(n)$ and fixed $x \in \mathbb{K}$,

$$\operatorname*{K}_{k=0}^{n}(a(k-1)x/(a(k)-x)) = \frac{1}{\sum_{k=0}^{n}(-x)^k/\prod_{i=0}^{k} a(i)} - x \qquad (n \geq 0)$$

5 Inequalities involving Admissible Sequences[1]

5.1 Motivation

So far in this thesis, we have been dealing with identities $f(n) = g(n)$ only. In fact, most of the available algorithms for special functions are restricted to the treatment of identities. Despite their importance in all branches of mathematics, *inequalities* have received only very little attention so far. Textbooks on inequalities (Hardy *et al.*, 1952; Mitrinović, 1964, 1970) contain a lot of entries, from which even simple ones could not be verified by symbolic computation so far.

Some individual inequalities can be proven using algorithms for proving identities. For instance, Lagrange's identity

$$\sum_{k=1}^{n} x_k^2 \sum_{k=1}^{n} y_k^2 - \Big(\sum_{k=1}^{n} x_k y_k\Big)^2 = \sum_{k=1}^{n}\sum_{i=1}^{k}(x_k y_i - x_i y_k)^2 \qquad (n \geq 1),$$

which can be proven with the algorithm of Section 4.6 or by symmetric function algorithms, immediately implies the Cauchy-Schwarz inequality,

$$\Big(\sum_{k=1}^{n} x_k y_k\Big)^2 \leq \sum_{k=1}^{n} x_k^2 \sum_{k=1}^{n} y_k^2 \qquad (n \geq 1).$$

Similarly, a hypergeometric identity due to Askey and Gasper (1976), which can also be proven algorithmically (Ekhad, 1993), implies the inequality

$$\sum_{k=0}^{n} P_k^{(\alpha,0)}(x) > 0 \quad (\alpha > -1, -1 < x \leq 1, n \geq 0),$$

where $P_k^{(\alpha,0)}(x)$ denotes the Jacobi polynomials (cf. p. 11). This inequality was used in the first proof of the Bieberbach conjecture (de Branges, 1985).

Paule (2005) proves the inequality

$$\sum_{k=n}^{\infty} \frac{1}{k^2\binom{n+k}{k}} < \frac{2}{n\binom{2n}{n}} \qquad (n \geq 1)$$

(Schur and Knopp, 1918) by an application of the extended Gosper algorithm (Petkovšek *et al.*, 1997).

All these proofs still have to be considered "hand-made", even though symbolic computation gives some assistance. As opposed to this, the procedure we are proposing in this chapter is able to prove many inequalities entirely automatically—or with little human support only. Though

[1] The results in this chapter originate from a collaboration with S. Gerhold (Gerhold and Kauers, 2005).

the procedure does not terminate in general, it is surprisingly successful on a lot of examples. Section 5.5 contains an extensive list of inequalities that were proven by our method.

Unless otherwise stated, we assume throughout this chapter that $\mathbb{K} = \mathbb{R} \cap \bar{\mathbb{Q}}$ is the field of real algebraic numbers, or a subfield thereof.

5.2 Polynomial Inequalities and CAD

In this section, we introduce some notation about inequalities, and recall some fundamental facts about their algorithmic treatment.

A *polynomial inequality* over $\mathbb{K}[x_1,\dots,x_n]$ is a formula of the form

$$p(x_1,\dots,x_n) \diamondsuit q(x_1,\dots,x_n)$$

where $p,q \in \mathbb{K}[x_1,\dots,x_n]$ and $\diamondsuit \in \{=,\neq,<,>,\leq,\geq\}$. Without loss of generality, we may assume that q is the zero polynomial. A *Tarski formula* (or *formula* for short) over $\mathbb{K}[x_1,\dots,x_n]$ is a boolean combination of polynomial inequalities. A finite set of formulas over $\mathbb{K}[x_1,\dots,x_n]$ is called a system of formulas.

Given a specific system of formulas $S = \{A_1(x_1,\dots,x_n),\dots,A_s(x_1,\dots,x_n)\}$, we may wonder about the *consistency* of this system. The system S is called *consistent* if

$$\exists\, x_1,\dots,x_n \in \mathbb{R} : A_1(x_1,\dots,x_n) \wedge \dots \wedge A_s(x_1,\dots,x_n),$$

and *inconsistent* otherwise. It is called *universally valid* if

$$\forall\, x_1,\dots,x_n \in \mathbb{R} : A_1(x_1,\dots,x_n) \wedge \dots \wedge A_s(x_1,\dots,x_n).$$

Tarski (1951) has shown that the more general question about the truth of a given formula

$$Q_1x_1\ Q_2x_2\ \dots\ Q_nx_n : A_1(x_1,\dots,x_n) \wedge \dots \wedge A_s(x_1,\dots,x_n) \tag{5.1}$$

in the theory of the real numbers is decidable for any given system S of formulas over $\mathbb{K}[x_1,\dots,x_n]$ and any choice of quantifiers $Q_1,\dots,Q_n \in \{\forall,\exists\}$. In particular, it is decidable whether a given system is inconsistent, or universally valid.

Tarski's algorithm is impractical. The method of *cylindrical algebraic decomposition* (CAD) introduced by Collins (1975) gives a more efficient algorithm for deciding formulas like (5.1). For a thorough introduction into the field, we refer the interested reader to the book of Caviness and Johnson (1998). For our purpose, it is not necessary to understand in detail how CAD works. It is sufficient to know that it is capable of deciding consistency of a set of Tarski formulas. However, for the sake of completeness, let us briefly sketch the underlying principles.

Given a finite set $P = \{p_1,\dots,p_m\} \subseteq \mathbb{K}[x_1,\dots,x_n]$, we associate to every point $x = (x_1,\dots,x_n) \in \mathbb{R}^n$ the sign pattern $(\operatorname{sgn}(p_1(x_1,\dots,x_n)),\dots,\operatorname{sgn}(p_m(x_1,\dots,x_n))) \in \{-,0,+\}^m$. A *cell* is a connected subset $C \subseteq \mathbb{R}^n$ with invariant sign pattern, which is maximal in the sense that for all $x \notin C$, $C \cup \{x\}$ is not connected or not sign pattern invariant. No matter how P is chosen, there will be finitely many different cells only, and all these cells together form a disjoint cover of $\mathbb{R}^n$, called an *algebraic decomposition* of $\mathbb{R}^n$. Figure 5.1.a) shows the algebraic decomposition defined by $P = \{x^2+y^2-4, x+y\} \subseteq \mathbb{K}[x,y]$.

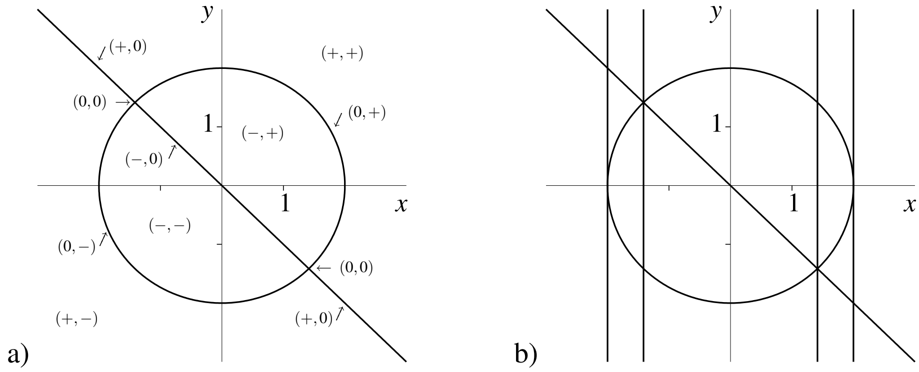

Figure 5.1 a) algebraic decomposition of $\{x^2+y^2-4, x+y\}$
b) extension of the decomposition in a) to a cylindrical algebraic decomposition: $\{x^2+y^2-4,\, x+y,\, x-2,\, x-\sqrt{2},\, x+\sqrt{2},\, x+2\}$

Now consider the formula (5.1) and assume that the atoms of $A_i(x_1,\dots,x_n)$ $(i=1,\dots,s)$ are of the form $p_j(x_1,\dots,x_n) \diamondsuit 0$ for polynomials $p_1,\dots,p_m \in \mathbb{K}[x_1,\dots,x_n]$. Truth of (5.1) could be decided by inspection if for each cell C in the algebraic decomposition induced by $P=\{p_1,\dots,p_m\}$ a sample point $x=(x_1,\dots,x_n)\in C$ were known. The idea of CAD is to supplement the polynomial list P with additional polynomials in such a way that such sample points can be easily obtained.

For $n>1$, the projection maps π_n are defined via

$$\pi_n \colon \mathbb{R}^n \to \mathbb{R}^{n-1}, \qquad \pi_n(x_1,\dots,x_n) := (x_1,\dots,x_{n-1}).$$

An algebraic decomposition of $\mathbb{R}^n$ is called *cylindrical* if its image under π_n is again a cylindrical algebraic decomposition. Equivalently, a decomposition is called cylindrical if for any two cells C_1, C_2 of this decomposition, the cells $\pi_n(C_1)$ and $\pi_n(C_2)$ are either identical or disjoint.

Any set $P=\{p_1,\dots,p_s\}\in\mathbb{K}[x_1,\dots,x_n]$ can be supplemented with polynomials $q_1,\dots,q_t$ such that the decomposition induced by $P\cup\{q_1,\dots,q_t\}$ is cylindrical. The computation of suitable polynomials q_i is the first phase of the CAD algorithm, called the *projection phase.* Figure 5.1.b) shows a cylindrical algebraic decomposition for $\{x^2+y^2-4, x+y\}$. In this example, the cylindrical decomposition consists of 47 cells, while the decomposition in a) has only got 11 cells.

The second phase of the CAD algorithm consists of the computation of sample points, one for each cell of the decomposition. This is done bottom-up, starting with the projection of the cylindrical decomposition to the one-dimensional real line. In this situation, we have univariate polynomials only. Suppose that $\xi_1,\xi_2,\dots,\xi_r$ are the real roots of these polynomials, labeled such that $\xi_i<\xi_{i+1}$ for $0<i<r$. Then suitable sample points for the projected decomposition are

$$\xi_1-1,\ \xi_1,\ \tfrac{1}{2}(\xi_2+\xi_1),\ \xi_2,\ \dots\dots,\ \xi_{r-1},\ \tfrac{1}{2}(\xi_r+\xi_{r-1}), \xi_r,\ \xi_{r+1}+1.$$

In order to get sample points for the projection to the two-dimensional case, consider these sample points, one after the other. For each sample point x, consider the the projection of the decomposition to the two-dimensional space, and substitute x for x_1. Then the resulting polynomials are univariate in x_2, and sample points can be found as before. In the same way, sample points for the d-dimensional projection can be obtained from sample points for the $(d-1)$-dimensional projection for $d = 2,\dots,n$.

For further algorithmic details, we refer to the literature.

Several implementations of the CAD algorithm are available. Besides special purpose software like QEPCAD (Collins and Hong, 1991; Brown, 2003a), there is a builtin implementation of CAD in Mathematica due to Strzeboński (2000).

5.3 The Proving Procedure

We are interested in proving inequalities involving sequences that can be defined via difference equations. Our main interest are admissible sequences, but the method also applies to objects that do not constitute admissible sequences. We will describe the method in full generality, and only afterwards discuss the most relevant special cases.

Definition 5.2 Let $f_1,\dots,f_m$ be function symbols. An *atomic formula* for $f_1,\dots,f_m$ (in n) is a formula of the form

$$p(f_1(n),\dots,f_1(n+r_1-1),\dots\dots,f_m(n),\dots,f_m(n+r_m-1)) \diamondsuit 0$$

with a multivariate polynomial p over $\mathbb{K}$, a relation $\diamondsuit \in \{=,\neq,<,>,\leq,\geq\}$, and $r_1,\dots,r_m \in \mathbb{N}_0$ fixed.

A *formula* for $f_1,\dots,f_m$ is defined by structural induction: every atomic formula is a formula, and if $A(n)$ and $B(n)$ are formulas then so are $\neg A(n)$, $A(n) \wedge B(n)$, $A(n) \vee B(n)$, $A(n) \Rightarrow B(n)$, $A(n) \Leftrightarrow B(n)$.

Let $A(n)$ be a formula for $f_1,\dots,f_m$. The *associated Tarski formula* of $A(n)$ is defined as the Tarski formula obtained form $A(n)$ by replacing each occurrence $f_i(n+j)$ by the variable $s^j t_i$. The maximum number r such that $A(n)$ contains a subexpression $f_i(n+r)$ is called the *order* of $A(n)$.

Let $f_1(n),\dots,f_m(n)$ be sequences over $\mathbb{R}$ and $A(n)$ be a formula for $f_1,\dots,f_m$. By $A(N)$ ($N \in \mathbb{N}$ fixed) we denote the truth value of the formula obtained by $A(n)$ upon replacing each occurrence of $f_i(n+j)$ by the value of the sequence $f_i(n)$ at $n = N+j$. We say that $A(n)$ is *valid* if $A(N)$ is true for all $N \geq 1$.

The goal of the proving procedure is to find out whether a given formula $A(n)$ is valid. To this end, a number $N \in \mathbb{N}$ is computed which has the property that

$$(\forall\, n \in \mathbb{N} : A(n)) \iff A(1) \wedge A(2) \wedge \dots \wedge A(N). \tag{5.2}$$

That is, the universal quantifier binding n is eliminated. The underlying correctness argument of the procedure is a complete induction on n, similar as in Section 4.1. Certainly, a number N that satisfies the induction step formula

$$\forall\, n \in \mathbb{N} : A(n) \wedge A(n+1) \wedge \dots \wedge A(n+N-1) \Rightarrow A(n+N) \tag{5.3}$$

would perfectly qualify for establishing validity of (5.2). Consider the associated Tarski formula of (5.3), being of the form

$$\forall\, t_1, st_1, \ldots, s^r t_1, \ldots\ldots, t_m, st_m, \ldots, s^r t_m \in \mathbb{R} : B(t_1, st_1, \ldots, s^r t_1, \ldots\ldots, t_m, st_m, \ldots, s^r t_m) \qquad (5.4)$$

for some Tarski formula B. Whether or not this formula is true in the theory of real numbers—that is, regarding the $s^i t_j$ as polynomial ring variables only—can be determined by CAD (Section 5.2). If (5.4) holds, then in particular (5.3) holds, and we are done. On the other hand, if (5.4) does not hold, then we are not entitled to draw any conclusion about the validity of (5.3). In this case, we proceed to check the next larger value of N. We continue to iterate until eventually (5.4) becomes true.

This reasoning is, of course, in full analogy to the zero equivalence algorithm presented in the previous chapter. The main technical difference is that CAD is used instead of Gröbner basis computations for establishing the induction step.

The details of the procedure are below. We extend the shift function s defined for difference polynomials $p \in \mathbb{K}\{t_1, \ldots, t_m\}$ naturally to Tarski formulas: $s(A \diamondsuit B) := s(A) \diamondsuit s(B)$ for any logical junctor $\diamondsuit$ and $s(p \diamondsuit 0) := s(p) \diamondsuit 0$ for every $\diamondsuit \in \{=, \neq, <, >, \leq, \geq\}$.

Procedure 5.3 (Proving Formulas)
Input: Computable sequences $f_1(n), \ldots, f_m(n)$ in $\mathbb{K}$, a set R of formulas for $f_1, \ldots, f_m$ which are valid, a formula $A(n)$ for $f_1, \ldots, f_m$.
Output: True, if $A(n)$ is valid, False otherwise
Assumption: Truth of $A(N)$ can be decided for every particular $N \in \mathbb{N}$

Let r be the maximum of the orders of $A(n)$ and the formulas in R
$R' := \emptyset$
forall $B(n) \in R$ **do**
  Let r_0 be the order of $B(n)$ and B' be the associated Tarski formula of $B(n)$
  $R' := R' \cup \{\, s^i B' : i = 0, \ldots, r - r_0 \,\}$
Let r_0 be the order of $A(n)$ and A' be the associated Tarski formula of $A(n)$
if $A(1) \wedge A(2) \wedge \ldots \wedge A(r - r_0)$ is False **then**
  return False
$N := r - r_0$
while $R' \cup \{A', \ldots, s^{N-1} A', \neg s^N A'\}$ is satisfiable **do**
  $N := N + 1$
  if $A(N)$ is False **then**
    return False
  $R' := R' \cup sR'$
return True

With respect to complexity, CAD is even more sensitive than Gröbner bases computations. Therefore, an actual implementation of Procedure 5.3 should carefully preprocess the formula system on which CAD is invoked. For instance, if R contains formulas of the form $f_i(n + r_i) = \text{poly}$ (e.g., recurrences), these should be used to eliminate all variables $s^j t_i$ ($j \geq r_i$) using this relation.

Theorem 5.4 Procedure 5.3 is correct.

Proof The procedure is evidently correct whenever it returns False, because this only happens when a counterexample N, where the formula does not hold, is actually found. Assume now that the procedure returns True.

We will show that then $A(n)$ is true for all $n \geq 1$. First, we have that $A(1) \wedge A(2) \wedge \ldots \wedge A(N)$ is true by lines 7 and 12 (otherwise the procedure would have stopped before and returned False). In addition, we have

$$\forall\, n \in \mathbb{N} : A(n) \wedge A(n+1) \wedge \cdots \wedge A(n+N-1) \implies A(n+N).$$

For, suppose this implication does not hold. Then there would be a number $n \in \mathbb{N}$ with $A(n)$, $A(n+1), \ldots, A(n+N-1)$, $\neg A(n+N)$ all being true. In addition, substituting the values $f_i(n+j)$ for $s^j t_i$ in R' will give true, because R only contains formulas which are valid by assumption. Hence, all formulas in $R' \cup \{A', \ldots, s^{N-1}A', \neg s^N A'\}$ become true upon this substitution. This is in contradiction to line 10. ∎

Example 5.5 Let us execute the procedure in detail on the simple inequality

$$\sum_{k=1}^{n} k > \tfrac{1}{2}n^2 \qquad (n \geq 1).$$

(See Section 5.5 for more interesting examples.) Take

$$R = \{f_1(n+1) = f_1(n) + 1, f_2(n+1) = f_2(n) + f_1(n+1)\}.$$

The admissible formula to be shown is $A(n) := (f_2(n) > \frac{1}{2} f_1(n)^2)$.

Lines 1–5 only adjust the orders of R and A, and turn the recurrences into formulas. We obtain $R' = \{st_1 = t_1 + 1, st_2 = t_2 + st_1\}$ and $A' = (t_2 > \frac{1}{2}t_1^2)$.

In line 7, we check that $A(1) = (1 > \frac{1}{2})$ holds. (Note that $r = 1$ and $r_0 = 0$).

For $N = 1$, we next apply CAD to determine whether the system

$$\{\, st_1 = t_1 + 1, st_2 = t_2 + st_1, t_2 > \tfrac{1}{2}t_1^2, st_2 \leq \tfrac{1}{2}st_1^2 \,\}$$

is satisfiable over the reals. As it turns out that it is not, the while loop terminates and the procedure returns True according to line 15.

We have deliberately not imposed restrictions on the set R of known facts about $f_1(n), \ldots, f_m(n)$. Of course, no reasonable output can be expected if, for instance, $R = \emptyset$. In fact, the procedure does not terminate if $R = \emptyset$ and $A(n)$ is not tautologic.

Most commonly, the set R will contain an admissible system (Def. 3.1) for $f_1(n), \ldots, f_m(n)$ as a subset. However, quite in contrast to Algorithm 4.2, there is no guarantee for termination even in this case. Let us illustrate the situation with the simple exponential inequality $3^n > 2^n$ $(n \geq 1)$. We take $R = \{f_1(n+1) = 3f_1(n), f_2(n+1) = 2f_2(n)\}$ and $A(n) = (f_1(n) > f_2(n))$.

Applying the procedure gives in the N-th iteration of the while loop a system of inequalities that is equivalent to $3^N t_1 > 2^N t_2 \wedge 3^{N+1} t_1 \leq 2^{N+1} t_2$. For every value of N, the corresponding cylindrical algebraic decomposition looks like the special case $N = 1$ which is depicted in Figure 5.6. The

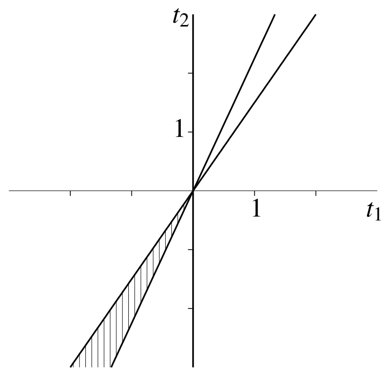

Figure 5.6 Cylindrical Algebraic Decomposition for $3^N t_1 > 2^N t_2 \wedge 3^{N+1} t_1 \leq 2^{N+1} t_2$, for $N = 1$.

formula is valid for every point (t_1, t_2) in the shaded cell, so it is in particular not inconsistent. As N grows, the shaded sector of exceptional points becomes more and more narrow, yet it is nonempty for every N, and therefore the procedure does not terminate.

In order to make the proof of $3^n > 2^n$ go through, we have to supply additional information. For example, putting additionally $2^n > 0$ or $3^n > 0$ into the knowledge base R makes the procedure terminate already after the first iteration. This is typical. The proof of almost every nontrivial inequality requires some—usually trivial—facts as additional knowledge besides the recurrences defining the involved sequences. Remarkably enough, formulas specified as additional knowledge can almost always be proven independently by the same procedure.

In cases where Procedure 5.3 does not terminate, there is no way to prove the desired inequality by induction, in the following sense. For every $N > 0$, there exist sequences $f_1(n), \ldots, f_m(n)$ for which all formulas in R are valid, but $A(n)$ is true only for $1 \leq n < N$ and false for $n = N$. This rules out the possibility to do an induction proof solely based on the knowledge supplied in R. In the example $3^n > 2^n$, using

$$R = \{f_1(n+1) = 3f_1(n), f_2(n+1) = 2f_2(n)\},$$

the solutions

$$f_1(n) = -2^N \cdot 3^n, \qquad f_2(n) = -(3^N + \varepsilon) \cdot 2^n$$

for sufficiently small $\varepsilon > 0$ are such that $f_1(n) > f_2(n)$ holds for $n = 1, \ldots, N$, but no longer for $n = N+1$ and larger N. An induction proof using only the recurrences is therefore not possible in this case.

To conclude the discussion about termination, let us remark that a decision procedure capable of proving inequalities of the type we are considering is a quite unreasonable thing to hope for. For, suppose such an algorithm exited. Then, for any given admissible sequence $f(n)$, we could apply this algorithm to the inequality $f(n)^2 > 0$. The algorithm would return False if and only if there exists an $n \in \mathbb{N}$ with $f(n) = 0$. Deciding existence of such n is, however, a reputedly difficult problem. Already for the class of C-finite sequences, which constitutes only a very small subset of all admissible sequences, it is a well-known open problem whether this question is algorithmically decidable (Everest *et al.*, 2003, Section 2.3).

5.4 Variations on the Scheme

Typically, Procedure 5.3 will be applied to inequalities about admissible sequences over $\mathbb{K}$. In this section, we consider some classes of inequalities that do not directly match this pattern, but to which the procedure applies as well.

1 Inequalities involving Parameters

We have restricted the ground field $\mathbb{K}$ to a subfield of $\mathbb{R}$ in the present chapter, but inequalities involving parameters need not be excluded from consideration. Instead of treating a parameter, x, by extending the ground field $\mathbb{K}$ by a new transcendental element for x, as usual, we may encode parameters as constant functions. The knowledge base R may be extended by an equation of the form $f(n+1) = f(n)$, the initial value of $f(n)$ being $f(1) = x$. The parameter is then hidden from the termination condition of the main loop in Procedure 5.3.

Example 5.7 Bernoulli's inequality

$$(x+1)^n \geq 1+nx \qquad (n \geq 0,\ x \geq -1)$$

can be proven by applying the procedure to

$$\begin{aligned} R = \{&f_1(n+1) = f_1(n), f_2(n+1) = f_2(n)+1, f_3(n+1) = (f_1(n)+1)f_3(n),\\ &f_1(n) \geq -1, f_2(n) \geq 0\},\\ A(n) = f_3&(n) \geq 1+f_2(n)f_1(n). \end{aligned}$$

One initial value has to be checked: $A(1) = (x+1) \geq (x+1)$ is true.
It is only little known but easy to verify that Bernoulli's inequality already holds for $x \geq -2$. Our procedure confirms this fact, using two more iterations.

If the parameters only occur polynomially in the admissible system and/or in the initial values, as in the example above, then the initial values can be checked by another application of CAD. Otherwise, if there are more delicate relations amongst the parameters (e.g., if the parameters x and y are related via $y = \sin(x)$), checking initial values might become difficult for a machine. In this case, the procedure may just be used for providing the induction step, and leave the verification of the initial values to the user.

2 Inequalities involving Free Sequences

By using function symbols for which no recurrence is present in the set R, inequalities involving free sequences can be treated. This is in analogy to Section 4.6. The standard example is the Cauchy-Schwarz inequality

$$\Big(\sum_{k=1}^{n} x_k y_k\Big)^2 \leq \sum_{k=1}^{n} x_k^2 \sum_{k=1}^{n} y_k^2 \qquad (n \geq 1).$$

This inequality can be proven by Procedure 5.3 if R contains, besides the recurrences defining the sums, knowledge about the nonnegativity of the sums on the right hand side.
An inequality depending on a free sequence $f(n)$ may hold only when $f(n)$ is monotonic, or positive, or bounded, etc. Such restrictions can easily be formalized by formulas $f(n+1) > f(n)$, or $f(n) > 0$, or $-M < f(n) < M$ that can be put into R.

Example 5.8 The procedure succeeds in proving the inequality

$$\sum_{k=1}^{n}(-1)^{k-1}a(k)^2 \geq \Big(\sum_{k=1}^{n}(-1)^{k-1}a(k)\Big)^2 \qquad (n \geq 1)$$

which holds for any positive and decreasing sequence $a(n)$. (Mitrinović, 1970, Thm. 6, p. 112).

3 Inequalities involving Algebraic Sequences

Though algebraic sequences are originally defined via algebraic equations, it is not easy to handle them with Gröbner basis. Consider, for example, the sequence of square roots, $f(n) = \sqrt{n}$. Clearly, a defining equation for this sequence is $f(n)^2 = n$. But this difference equation has got a continuum of solutions: each sequence $s(n)\sqrt{n}$ where $s(n)$ is an arbitrary sequence in $\{-1,1\}$ solves this equation. As CAD permits the use of inequalities, it is possible to specify which solution of the equation $f(n)^2 = n$ we mean, for instance by imposing the additional constraint $f(n) \geq 0$.

It is therefore possible to prove inequalities involving algebraic sequences, such as the square root function. One example of such an inequality is

$$\Big(\sum_{k=1}^{n}\sqrt{k}\Big)^2 \leq \Big(\sum_{k=1}^{n}\sqrt[3]{k}\Big)^3 \qquad (n \geq 0),$$

which was to our knowledge first proven by Procedure 5.3.

4 Inequalities requiring Preprocessing by other Algorithms

Inequalities which are not amenable to Procedure 5.3 can sometimes be rewritten such as to make the proving procedure applicable. As an example, consider the inequality

$$\sum_{k=n}^{\infty}\frac{1}{k^2\binom{n+k}{k}} < \frac{2}{n\binom{2n}{n}} \qquad (n \geq 1)$$

of Knopp and Schur mentioned in Section 5.1. We have

$$\sum_{k=1}^{\infty}\frac{1}{k^2\binom{n+k}{k}} = \psi'(n+1) = \frac{\pi^2}{6} - \mathrm{H}_n^{(2)} \qquad (n \geq 1),$$

where $\mathrm{H}_n^{(2)} = \sum_{k=1}^{n} 1/k^2$ is the nth second order Harmonic number, and ψ' denotes the digamma function (Andrews *et al.*, 1999).

Using Zeilberger's algorithm, we obtain

$$\sum_{k=1}^{n-1}\frac{1}{k^2\binom{k+n}{k}} = \sum_{k=1}^{n-1}\frac{3k^2+3k+1}{2k^2(k+1)(2k+1)\binom{2k-1}{k-1}} - \frac{1}{(k+1)^2} \qquad (n \geq 1).$$

By this identity, the original inequality can be simplified to

$$\frac{\pi^2}{6} - 1 - \sum_{k=1}^{n-1}\frac{3k^2+3k+1}{k^2(k+1)(2k+1)\binom{2k}{k}} < \frac{2}{n\binom{2n}{n}} \qquad (n \geq 1).$$

To this latter inequality, Procedure 5.3 is applicable.

A minor technical problem might be the occurrence of π, which is not a real algebraic number, as required by CAD. However, we can easily circumvent this problem by regarding π as a continuous parameter subject to the restriction $3 < \pi < 4$. In this setup the proof is successful.

5.5 Examples

We have seen that Procedure 5.3 does not terminate in general. Nevertheless, it is successful on a large number of inequalities appearing in the literature. Many of them can now be done without any human support, and where human support is necessary, it is only of trivial nature. As evidence for the practical relevance of the method, we give a list of inequalities which we could verify by our procedure. We are not aware of any algorithmic method which could prove any of the inequalities listed in this section.

The textbooks of Mitrinović (1964, 1970) contain a vast collection of inequalities, many of which are in the scope of our method. Table 5.10 on the following double page contains a collection of some inequalities from the 1964 book that we were able to verify by means of Procedure 5.3. Some further example are in order.

Example 5.9

(1) Turan's inequality

$$P_n(x)^2 - P_{n-1}(x)P_{n+1}(x) \geq 0 \qquad (-1 < x < 1, n > 1)$$

for the Legendre polynomials $P_n(x)$ (cf. p. 11) (Andrews *et al.*, 1999).

(2) A family $p_n(x)$ of monic orthogonal polynomials defined via

$$p_{n+2}(x) = (x - c_n)p_{n+1}(x) + \lambda_n p_n(x) \qquad p_{-1}(x) = 0, p_0(x) = 1$$

for some sequences c_n and λ_n is said to be *positive definite* if the c_n are real and the λ_n are positive. Such polynomials satisfy the inequality

$$p'_{n+1}(x)p_n(x) - p_{n+1}(x)p'_n(x) > 0 \qquad (x \in \mathbb{R}, n \geq 0)$$

(Chihara, 1978; Andrews *et al.*, 1999).

(3) Let $a(n)$ be an arbitrary sequence in $\mathbb{N}$, and $p(n), q(n)$ be the continuants of the continued fraction $\mathbf{K}_{k=0}^{n}(1/a(k))$. Then

$$q(n) < 2^{(n+1)/2} \qquad (n \geq 1)$$

(Khinchin, 1964).

(4) Levin's inequality (Mitrinović, 1970, 3.2.13)

$$1 \leq \frac{nx^{n+1} + 1}{x^n(n+1)} \leq \tfrac{1}{2}n(1-x)^2x^{-n} + 1 \qquad (0 < x \leq 1, n > 0)$$

(5) The inequality

$$\frac{nF_{n+6}}{2^{n+1}} + \frac{F_{n+8}}{2^n} - F_8 < 0 \qquad (n \geq 1)$$

proposed by Janous (2002)

(6) The inequality

$$\sum_{k=1}^{n} \frac{L_k^2}{F_k} \geq \frac{(L_{n+2}-3)^2}{F_{n+2}-1} \qquad (n \geq 2)$$

proposed by Diaz and Egozcue (2002b), using as additional knowledge that $F_n \geq 1$ $(n \geq 2)$.

(7) If $a \geq 1, a+b \geq 1$ and $x_n > 0$ $(n \geq 1)$, then

$$\sum_{k=1}^{n} x_k^a \Big(\sum_{i=1}^{k} x_i\Big)^b \leq \Big(\sum_{k=1}^{n} x_k\Big)^{a+b} \qquad (n \geq 1)$$

(Beesack, 1969). This inequality can be done for any specific $a,b \in \mathbb{Z}$, using positivity of the involved sums as knowledge in addition to the defining recurrences.

(8) If $f(n)$ is defined via

$$f(n+1) = 1 + \frac{n}{f(n)} \quad (n \geq 1), \qquad f(1) = 1,$$

then

$$\sqrt{n - \tfrac{3}{4}} \leq f(n) - \tfrac{1}{2} \leq \sqrt{n + \tfrac{1}{4}} \qquad (n \geq 1)$$

(Nanjundiah, 1993)

(9) The inequality

$$F_n(x)^2 \leq (x^2+1)^2(x^2+2)^{n-3} \qquad (n \geq 3)$$

for the *Fibonacci polynomials* $F_n(x)$ defined via

$$F_{n+2}(x) = xF_{n+1}(x) + F_n(x) \quad (n \geq 0), \qquad F_1(x) = 1, F_2(x) = x.$$

(Mitrinović, 1970, 3.3.38)

There are, of course, also inequalities on which the procedure fails. The inequality of Vietoris (Andrews *et al.*, 1999), for instance, says that for every sequence $a(n)$ with

$$a(2n) \leq \frac{2n-1}{2n} a(2n-1),$$

we have

$$\sum_{k=1}^{n} a(k) \sin(kx) > 0 \qquad (0 < x < \pi, n \geq 1).$$

The Fejér-Jackson inequality

$$\sum_{k=1}^{n} \frac{1}{k} \sin(kx) > 0 \qquad (0 < x < \pi, n \geq 1)$$

Inequality	Domain of validity	Source
• $\frac{1}{2\sqrt{n}} < 4^{-n}\binom{2n}{n} < \frac{1}{\sqrt{3n+1}}$	$n \geq 2$	3.27
• $\prod_{k=1}^{n} \frac{4k-1}{4k+1} < \sqrt{\frac{3}{4n+3}}$	$n \geq 1$	3.28
• $\frac{1}{2\sqrt{n}} < \prod_{k=1}^{n} \frac{2k-1}{2k} < \frac{1}{\sqrt{2n+1}}$	$n > 1$	3.29
• $\prod_{k=0}^{n} \frac{3k+4}{3k+2} > 1 + \frac{2}{3}\mathrm{H}_{n+1}$	$n \geq 1$ (using $\mathrm{H}_n \geq 0$)	3.31
• $(n+a)! + n! > (n+a-1)! + (n+1)!$	$n \geq 1, a \geq 1$ (using $n! > 0, (n+a)! > 0$)	3.57
• $\sum_{k=1}^{n} \frac{1}{\sqrt{k}} > 2(\sqrt{n+1} - 1)$	$n \geq 1$	4.1
• $\mathrm{H}_{2n} > \frac{1}{2} + \mathrm{H}_n$	$n > 1$	4.3
• $\mathrm{H}_{3n+1} > 1 + \mathrm{H}_n$	$n \geq 1$	4.4
• $\sum_{k=1}^{n} a^{2k} \leq n(a^{2n+1} + 1)$	$n \geq 1, a \geq 1$ (using $a^{2n} \geq 0, \sum_{k=1}^{n} a^{2k} \geq 0$)	4.8
• $\sum_{k=1}^{n} \frac{1}{\sqrt{k}} > \sqrt{n}$	$n \geq 2$ (using $\sum_{k=1}^{n} \frac{1}{\sqrt{k}} \geq 0$)	4.14
• $\sqrt{\sum_{k=1}^{n} k^2} \geq \sqrt[3]{\sum_{k=1}^{n} k^3}$	$n \geq 0$	4.15
• $\prod_{k=1}^{n} (a_k + 1) > 1 + \sum_{k=1}^{n} a_k$	$a_k > 0, n \geq 1$ (using $\sum_{k=1}^{n} a_k \geq 0$)	4.16
• $\prod_{k=1}^{n} (1 - a_k) > 1 - \sum_{k=1}^{n} a_k$	$0 < a_k < 1, \sum_{k=1}^{n} a_k < 1, n \geq 1$ (using $\sum_{k=1}^{n} a_k > 0$)	4.17
• $\prod_{k=1}^{n} (a_k + 1) < \frac{1}{1 - \sum_{k=1}^{n} a_k}$	$0 < a_k < 1, \sum_{k=1}^{n} a_k < 1, n \geq 1$ (using $\sum_{k=1}^{n} a_k > 0$)	4.18
• $\prod_{k=1}^{n} (1 - a_k) < \frac{1}{1 + \sum_{k=1}^{n} a_k}$	$0 < a_k < 1, n \geq 1$ (using $1 > \prod_{k=1}^{n} (1 - a_k) > 0$)	4.19

Inequality	Domain of validity	Source
• $\prod_{k=1}^{n-1}(a^k+1) < \frac{1-a}{a^n-2a+1}$	$0 < a < \frac{1}{2}, n \geq 1$ (using $a \geq a^n > 0$)	4.22
• $(n+1)\prod_{k=1}^{n}(a_k+1) \geq 2^n\Big(\sum_{k=1}^{n} a_k+1\Big)$	$a_k > 0, n \geq 1$ (using $2^{-n}\prod_{k=1}^{n}(a_k+1) > 1, \sum_{k=1}^{n} a_k > 1$)	4.23
• $\frac{1}{2}-\frac{1}{n+1} < \sum_{k=2}^{n}\frac{1}{k^2} < 1-\frac{1}{n}$	$n \geq 2$	4.29
• $\frac{1}{a+1}-\frac{1}{a+n+1} < \sum_{k=1}^{n}\frac{1}{(a+k)^2} < \frac{1}{a}-\frac{1}{a+n}$	$a > 0, n \geq 1$	4.30
• $(2n)!\sum_{k=2}^{2n}\frac{(-1)^k}{k!} \geq (2n-1)!!^2$	$n > 1$ (using $(2n)! > 1, (2n-1)!! > 1$)	4.41
• $(2n+1)x^n \leq \sum_{k=0}^{2n} x^k$	$n \geq 0, x \geq 0$ (using $\sum_{k=0}^{2n} x^k = \frac{x^{2n+1}-1}{x-1}, x^n \geq 0$)	7.35
• $1-x^{2n} \geq 2nx^n(1-x)$	$n \geq 0, 0 \leq x \leq 1$	7.36
• $(n-1)\Big(\sum_{k=1}^{n} a_k\Big)^2 \geq 2n\sum_{k=1}^{n} a_k \sum_{i=1}^{k-1} a_i$	$n \geq 1$	7.44
• $\Big(\sum_{k=1}^{n}\frac{1}{a_k}\Big)\sum_{k=1}^{n} a_k \geq n^2$	$a_k > 0, n > 0$ (using $\sum_{k=1}^{n} a_k \geq 0$)	8.3
• $\Big(\sum_{k=1}^{n} a_k b_k\Big)^2 \leq \Big(\sum_{k=1}^{n} k a_k^2\Big)\sum_{k=1}^{n}\frac{1}{k}b_k^2$	$n \geq 1$ (using $\sum_{k=1}^{n} k a_k^2 \geq 0, \sum_{k=1}^{n}\frac{1}{k}b_k^2 \geq 0$)	8.4
• $\Big(\sum_{k=1}^{n}\frac{1}{k}a_k\Big)^2 \leq \Big(\sum_{k=1}^{n} k^3 a_k^2\Big)\sum_{k=1}^{n}\frac{1}{k^5}$	$n \geq 1$ (using $\sum_{k=1}^{n} k^3 a_k^2 \geq 0, \sum_{k=1}^{n}\frac{1}{k^5} \geq 0$)	8.8
• $\Big(\sum_{k=1}^{n} a_k\Big)^2 \leq n\sum_{k=1}^{n} a_k^2$	$n \geq 0$	8.25
• $\sqrt{n+\sqrt{(n-1)+\sqrt{\cdots+\sqrt{2+\sqrt{1}}}}} < \sqrt{n}+1$	$n \geq 1$	11.1

Table 5.10 Inequalities from the book of Mitrinović (1964), which have been verified by Procedure 5.3. The inequalities appear in the books under the label given in the source column. Besides the stated knowledge, only defining recurrences were used.

is a famous special case. Both inequalities fit well into the specification of Procedure 5.3. Also to the inequality

$$\sum_{k=0}^{n} P_k^{(\alpha,0)}(x) > 0 \qquad (\alpha > -1, -1 < x < 1, n \geq 0)$$

mentioned in Section 5.1 the procedure is in principle applicable.

These are outstanding inequalities, fairly hard to prove even by hand (Andrews *et al.*, 1999, give proofs), and some of these inequalities remained open problems for quite some time. We were not able to verify any of these inequalities by using our procedure. Though theoretically applicable, already in the first iteration the CAD computations become too complex, and we ran out of memory (1GB) before a result was obtained.

It would be interesting to know whether the procedure terminates on any of these examples, or—if it does not—which additional knowledge could be supplied to obtain termination.

6 Algebraic Dependencies of Admissible Sequences

6.1 The Annihilating Ideal

Let $f_1(n), \ldots, f_m(n)$ be sequences in $\mathbb{K}$. An *algebraic dependency* (or *algebraic relation*) of these sequences is defined as a polynomial p such that

$$p\,(f_1(n), \ldots, f_1(n+r_1), \ldots\ldots, f_m(n), \ldots, f_m(n+r_m)) = 0 \qquad (n \in \mathbb{N}).$$

(Other authors may prefer a different definition of the term "algebraic dependency".) As an example, $x^2 - 1$ is an algebraic relation of the sequence $f(n) = (-1)^n$.

Consider the difference homomorphism

$$\varphi\colon\ (\mathbb{K}\{t_1, \ldots, t_m\}, s) \to (\mathbb{K}^{\mathbb{N}}, \mathbf{E})$$

mapping t_i to $f_i(n)$ $(i = 1, \ldots, m)$ and elements of $\mathbb{K}$ to the corresponding constant sequences. Evidently, the set of algebraic dependencies of $f_1(n), \ldots, f_m(n)$ is precisely the difference ideal $\ker\varphi$.

Example 6.1 Let $\mathfrak{a} := \ker\varphi$ for $\varphi\colon \mathbb{K}\{t\} \to \mathbb{K}^{\mathbb{N}}$ defined by $t \mapsto f(n)$. The following statements can be easily verified by hand.

(1) If $f(n) = 2^n$, then $\mathfrak{a} = \langle\langle st - 2t \rangle\rangle$.

(2) If $f(n) = (-1)^n$, then $\mathfrak{a} = \langle\langle t^2 - 1, st + t \rangle\rangle$.

(3) If $f(n) = (-1)^{\lfloor \log n \rfloor}$, then $\mathfrak{a} = \langle\langle t^2 - 1 \rangle\rangle$ (cf. Section 4.3).

(4) If $f(n) = 2^{n!}$, then $\mathfrak{a} = \{0\}$ (cf. Example 4.11).

For any given sequences $f_1(n), \ldots, f_m(n)$ over $\mathbb{K}$, we call the ideal of their algebraic dependencies the *annihilating ideal* (or *annihilator*) of these sequences, and write

$$\operatorname{ann}(f_1(n), \ldots, f_m(n)) := \ker\varphi \trianglelefteq \mathbb{K}\{t_1, \ldots, t_m\},$$

where $\varphi\colon \mathbb{K}\{t_1, \ldots, t_m\} \to \mathbb{K}^{\mathbb{N}}$ maps t_i to $f_i(n)$ and elements of $\mathbb{K}$ to the corresponding constant sequences. This notion of an annihilating ideal is to be carefully distinguished from ideals of annihilating operators, as they are used for instance in the work of Zeilberger (1990).

For the rest of this chapter, let $f_1(n), \ldots, f_m(n)$ be solutions of a normal admissible system S. Our goal is to develop methods for determining generators of the annihilator of $f_1(n), \ldots, f_m(n)$.

Definition 6.2 If r is the maximum order of the recurrences in S, then

$$\operatorname{alg}(f_1(n), \ldots, f_m(n)) := \operatorname{ann}(f_1(n), \ldots, f_m(n)) \cap \mathbb{K}\{t_1, \ldots, t_m\}_{r-1}$$

is called the *algebraic part* of $f_1(n), \ldots, f_m(n)$.

We have deliberately defined the algebraic part $\operatorname{alg}(f_1(n),\dots,f_m(n))$ as a function of the actual sequences $f_i(n)$. Observe that this is in contrast to the definition of the associated difference ideal (Def. 3.12), which only depends on an admissible system. For instance, the recurrence $f(n+1) = -f(n)$ has the solution $f(n) = \alpha(-1)^n$ for arbitrary $\alpha \in \mathbb{K}$, and we have $\operatorname{alg}(\alpha(-1)^n) = \langle t^2 - \alpha^2 \rangle$. The algebraic part depends on the initial value α, even though the recurrence does not.

Asking for computing a set of generators for the difference ideal of all algebraic dependencies only makes sense if that ideal is finitely generated. Fortunately, finite generation can be asserted.

Theorem 6.3 The difference ideal $\operatorname{ann}(f_1(n),\dots,f_m(n)) \trianglelefteq \mathbb{K}\{t_1,\dots,t_m\}$ is finitely generated. In fact, we have

$$\operatorname{ann}(f_1(n),\dots,f_m(n)) = \langle\langle \operatorname{alg}(f_1(n),\dots,f_m(n))\rangle\rangle + \mathfrak{a},$$

where $\mathfrak{a}$ is the associated difference ideal of S.

Proof "$\supseteq$" Obvious. "$\subseteq$" Let $p \in \operatorname{ann}(f_1(n),\dots,f_m(n))$ be of order $\tilde{r}$. For $\tilde{r} < r$ there is nothing to prove, so let $\tilde{r} \geq r$. Let $\ell \in \{1,\dots,m\}$ be the maximum index such that $s^{\tilde{r}}t_\ell$ effectively occurs in p, say

$$p = p_0 + p_1 s^{\tilde{r}}t_\ell + \cdots + p_{d-1}(s^{\tilde{r}}t_\ell)^{d-1} + p_d(s^{\tilde{r}}t_\ell)^d$$

for some d. The inclusion follows if we can construct an equivalent polynomial which is free of $s^{\tilde{r}}t_\ell$, by repeating the argument.

As S is normal, $\mathfrak{a}$ contains a difference polynomial either of the form $s^{\tilde{r}}t_\ell - q$ or of the form $qs^{\tilde{r}}t_\ell - 1$, with q depending only on variables $s^i t_j$ with $i < \tilde{r}$ or $j < \ell$. In the first case, we have

$$p \equiv p_0 + p_1 q + \cdots + p_{d-1}q^{d-1} + p_d q^d \bmod \mathfrak{a},$$

and we are done. For the second case, observe that by $qs^{\tilde{r}}t_\ell - 1 \in \mathfrak{a} \subseteq \operatorname{ann}(f_1(n),\dots,f_m(n))$ we have

$$p \in \operatorname{ann}(f_1(n),\dots,f_m(n)) \iff q^d p \in \operatorname{ann}(f_1(n),\dots,f_m(n)),$$

so it suffices to find a representation of the desired form for $q^d p$. Such a representation is given by

$$q^d p \equiv p_0 q^d + p_1 q^{d-1} + \cdots + p_{d-1}q + p_d \bmod \mathfrak{a}.$$

The proof of the ideal identity is now complete. As $\operatorname{alg}(f_1(n),\dots,f_m(n))$ is an ideal of the polynomial ring $\mathbb{K}\{t_1,\dots,t_m\}_{r-1}$ with finitely many variables, it is finitely generated by Hilbert's basis theorem. Furthermore, $\mathfrak{a}$ is finitely generated by definition. It follows that the annihilating ideal is finitely generated. ■

According to the theorem above, in order to find generators for an annihilator, it is sufficient to compute a basis of its algebraic part. As the algebraic part is finitely generated, it is meaningful to ask for a basis. If a finite basis is known, it can be used for computations. For instance, it is often necessary to know a basis for certain elimination ideals. Like in the proof of the theorem above, we can show that for every $p \in \operatorname{ann}(f_1(n),\dots,f_m(n)) \cap \mathbb{K}\{t_1,\dots,t_m\}_\ell$ (ℓ at least the order of S) there exist $q_{i,j} \in \mathbb{K}\{t_1,\dots,t_m\}_\ell$ such that

$$p = \sum_{i,j} q_{i,j} s^j p_i$$

where p_i belong to the algebraic part, or to the basis of the associated difference ideal, and all $s^j p_i$ belong to $\mathbb{K}\{t_1,\dots,t_m\}_\ell$. The $s^j p_i$ therefore form a polynomial ideal basis of the elimination ideal $\operatorname{ann}(f_1(n),\dots,f_m(n)) \cap \mathbb{K}\{t_1,\dots,t_m\}_\ell$. The following corollary is an immediate consequence, which is often needed in practice.

Corollary 6.4 Let $\Lambda \subseteq \mathbb{N} \times \{1,\dots,m\}$ be finite. Then a basis of the polynomial ideal

$$\operatorname{ann}(f_1(n),\dots,f_m(n)) \cap \mathbb{K}[s^i t_j : (i,j) \in \Lambda]$$

can be computed from a difference ideal basis of the annihilating ideal that consists of a basis of the algebraic part and a basis of the associated difference ideal.

Proof First compose a basis of $\operatorname{ann}(f_1(n),\dots,f_m(n)) \cap \mathbb{K}\{t_1,\dots,t_m\}_\ell$ where ℓ is the maximal number i such that $(i,j) \in \Lambda$ for some j, as described before. The task is completed by a Gröbner basis computation. ■

Although Theorem 6.3 asserts that the annihilating ideals that we consider are always finitely generated, there is little hope that a complete algorithmic way for computing generators of the algebraic part can be found. We have the following reduction.

Theorem 6.5 Let $f_1(n),\dots,f_m(n)$ be admissible sequences. Suppose there exists an algorithm for computing a finite basis of $\operatorname{alg}(f_1(n),\dots,f_m(n))$ from an admissible system S for $f_1(n),\dots,f_m(n)$. Then it is decidable whether there exists a point $n \in \mathbb{N}$ with $f(n) = 0$, for any given admissible sequence $f(n)$.

Proof Consider the indefinite product $F(n) := \prod_{i=1}^n f(i)$. The sequence $F(n)$ is admissible by Theorem 3.5.(2), and we have the equivalence

$$N \in \mathbb{N} \text{ is a root of } f(n) \iff (n-1)(n-2)\cdots(n-N+1)F(n) \text{ is the zero sequence.}$$

Let S be an admissible system in $g_1(n),\dots,g_\ell(n)$ with solutions $g_1(n) = n$, $g_2(n) = F(n)$. By assumption we can compute a basis for $\mathfrak{a} := \operatorname{alg}(g_1(n),\dots,g_\ell(n))$

Compute a Gröbner basis G for $\mathfrak{a}$ with respect to a lexicographic ordering where t_2 is the heaviest variable, t_1 is the lightest, and all other variables are arranged arbitrarily in between. Then $f(n)$ has a root if and only if G contains a polynomial of the form

$$(t_1-1)(t_1-2)\cdots(t_1-N+1)t_2,$$

which can be determined by inspection. ■

Theorem 6.5 says that finding a root of an admissible sequence is computationally at least as difficult as finding a generating set for the annihilator of a tuple of admissible sequences. Finding a root of an admissible sequence is, however, a very difficult problem. Already for the special case of C-finite sequences, i.e., determining whether a C-finite sequence $f(n)$ possesses a root, it is still not known whether this question is decidable (Everest *et al.*, 2003, Section 2.3).

Instead of an algorithm for actually computing the annihilating ideal, we present algorithms that "approximate" that ideal in a sense that will be made precise.

6.2 Geometric Interpretation

As $\mathbb{K}\{t_1,\ldots,t_m\}_{r-1}$ is only a polynomial ring in $r \times m$ variables, we may well say that the algebraic part $\operatorname{alg}(f_1(n),\ldots,f_m(n))$ of an annihilator is nothing else than an ordinary polynomial ideal. And just like every other polynomial ideal, it defines an algebraic variety $V \subseteq \mathbb{K}^{rm}$. In the present section, we will study the relationship of this variety with the sequences $f_1(n),\ldots,f_m(n)$. First we consider a concrete example.

Example 6.6 What are the algebraic dependencies of the Fibonacci sequence? The annihilator $\operatorname{ann}(F_n) \trianglelefteq \mathbb{Q}\{t\}$ is generated by $sst - st - t$ and possibly some difference polynomials depending on t and st only. These polynomials form the ideal $\operatorname{alg}(F_n) \trianglelefteq \mathbb{Q}[t, st]$. Consider the Fibonacci identity

$$F_n^4 + 2F_n^3F_{n+1} - F_n^2F_{n+1}^2 - 2F_nF_{n+1}^3 + F_{n+1}^4 - 1 = 0 \qquad (n \geq 0),$$

which is routinely verifiable by, e.g., Algorithm 4.2.

The corresponding difference polynomial, $p = t^4 + 2t^3st - t^2st^2 - 2tst^3 + st^4 - 1$, must belong to $\operatorname{alg}(F_n)$, because it is obviously not contained in $\langle\langle sst - st - t\rangle\rangle$. In fact, we claim that $\operatorname{alg}(F_n) = \langle p\rangle$, i.e.,

$$\operatorname{ann}(F_n) = \langle\langle p, sst - st - t\rangle\rangle.$$

This statement is contained in Exercise 6.81 of Graham *et al.* (1994). Its relevance was brought to our attention in a seminal seminar talk by Zimmermann (2002). Here we give a geometric proof which is independent of Zimmermann's.

Consider the variety $V(p)$ of $\mathfrak{a} := \langle p\rangle \trianglelefteq \mathbb{Q}[t, st]$, depicted in Figure 6.7. The variety consists of two irreducible components, each component corresponding to one irreducible factor of

$$p = (t^2 + tst - st^2 - 1)(t^2 + tst - st^2 + 1).$$

It follows from Cassini's identity

$$F_n^2 + F_nF_{n+1} - F_{n+1}^2 = (-1)^n \qquad (n \in \mathbb{N})$$

that for all points $(t, st) \in \mathbb{Q}^2$ of the form (F_{2n}, F_{2n+1}), the first factor of p vanishes, and for (t, st) of the form (F_{2n+1}, F_{2n+2}), the second factor does. In sloppy words, every branch of the variety

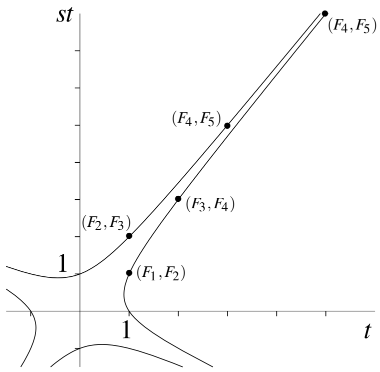

Figure 6.7 The algebraic dependencies of the Fibonacci sequence

$V(\mathfrak{a})$ carries half of the points (F_n, F_{n+1}). Conversely, every point (F_n, F_{n+1}) must belong to the variety V of $\text{alg}(F_n)$.

Now suppose that $\langle p \rangle \neq \text{alg}(F_n)$, i.e., $\langle p \rangle \subsetneq \text{alg}(F_n)$. If V denotes the variety of $\text{alg}(F_n)$, this implies the proper inclusion $V \subsetneq V(p)$. (Observe that p is square free.) Now, a theorem in algebraic geometry (Shafarevich, 1972, Thm. I.6.1) implies that V cannot have two one-dimensional components—every proper subvariety of $V(p)$ must have replaced at least one component by a collection of zero-dimensional varieties. But every zero-dimensional variety consists of finitely many points only (Eisenbud, 1995, Cor. 9.1), in contradiction to our observation that every irreducible component of $V(p)$ carries infinitely many points (F_n, F_{n+1}).

The reasoning in the previous example can be generalized to the following theorem.

Theorem 6.8 $\text{alg}(f_1(n), \ldots, f_m(n)) = \bigcap_{n=1}^{\infty} \langle s^i t_j - f_j(n+i) : i = 0, \ldots, r-1,\ j = 1, \ldots, m \rangle$

Proof "$\subseteq$" Let $p \in \text{alg}(f_1(n), \ldots, f_m(n))$ and let $N \in \mathbb{N}$ be arbitrary. By definition of the algebraic part $\text{alg}(f_1(n), \ldots, f_m(n))$, we have

$$p(f_1(n), \ldots, f_1(n+r-1), \ldots\ldots, f_m(n), \ldots, f_m(n+r-1)) = 0 \qquad (n \in \mathbb{N}),$$

so this relation holds in particular for $n = N$. Hence p reduces to zero modulo

$$\langle s^i t_j - f_j(N+i) : i = 0, \ldots, r-1,\ j = 1, \ldots, m \rangle,$$

and hence p belongs to that ideal. As N was arbitrary, the inclusion follows.
"$\supseteq$" Let $N \in \mathbb{N}$ be fixed. For all

$$p \in \langle s^i t_j - f_j(N+i) : i = 0, \ldots, r-1,\ j = 1, \ldots, m \rangle,$$

we have

$$p(f_1(N), \ldots, f_1(N+r-1), \ldots\ldots, f_m(N), \ldots, f_m(N+r-1)) = 0$$

Now if p belongs to the infinite intersection, then

$$p(f_1(n), \ldots, f_1(n+r-1), \ldots\ldots, f_m(n), \ldots, f_m(n+r-1)) = 0 \qquad (n \in \mathbb{N}),$$

and hence, by definition, p belongs to $\text{alg}(f_1(n), \ldots, f_m(n))$. ■

Corollary 6.9 The variety of $\text{alg}(f_1(n), \ldots, f_m(n))$ is precisely the Zariski closure of the set

$$\{\, (f_1(n), \ldots, f_1(n+r-1), \ldots\ldots, f_m(n), \ldots, f_m(n+r-1)) : n \in \mathbb{N}\} \subseteq \mathbb{K}^{rm}$$

of all function values of these sequences. ■

Corollary 6.10 The algebraic part $\text{alg}(f_1(n), \ldots, f_m(n))$ is a zero-dimensional ideal if and only if all the sequences $f_i(n)$ are periodic.

Proof Let us denote the point

$$(f_1(n),\dots,f_1(n+r-1),\dots\dots,f_m(n),\dots,f_m(n+r-1))$$

in the affine space $\mathbb{K}^{rm}$ by x_n.

"$\Leftarrow$" Suppose first that all the $f_i(n)$ are periodic, say $f_i(n)=f_i(n+p_i)$ for all $n\in\mathbb{N}$. Then for $p:=\operatorname{lcm}(p_1,\dots,p_m)$ we have $f_i(n)=f_i(n+p)$ ($n\in\mathbb{N}$, $i=1,\dots,m$). Hence the set $\{x_n : n\in\mathbb{N}\}$ is finite, and so it is Zariski closed. By Cor. 6.9, this set is the variety of $\operatorname{alg}(f_1(n),\dots,f_m(n))$, and hence this ideal is zero-dimensional.

"$\Rightarrow$" Now suppose that $\operatorname{alg}(f_1(n),\dots,f_m(n))$ is zero-dimensional. Then, by Cor. 6.9, the set $\{x_n : n\in\mathbb{N}\}$ is finite, say of cardinality $p\in\mathbb{N}$.

As the $f_i(n)$ are solutions of the admissible system S, we have that $x_n=x_{\tilde n}$ implies $x_{n+1}=x_{\tilde n+1}$ for all $n,\tilde n\in\mathbb{N}$. This implies that all the $f_i(n)$ are periodic, with a period at most p. ■

6.3 Approximation from Above

Theorem 6.8 gives rise to a method for approximating the algebraic part of an annihilator. As before, fix a tuple $f_1(n),\dots,f_m(n)$ of solutions of some normal admissible system S of order r.

Consider the sequence of ideals defined by

$$\begin{aligned}\mathfrak{a}_n &:= \bigcap_{\ell=1}^{n}\langle s^i t_j - f_j(\ell+i) : i=0,\dots,r-1,\ j=1,\dots,m\rangle \trianglelefteq \mathbb{K}\{t_1,\dots,t_m\}_{r-1}\\ &= \mathfrak{a}_{n-1}\cap\langle s^i t_j - f_j(n+i) : i=0,\dots,r-1,\ j=1,\dots,m\rangle\end{aligned}$$

Bases for these ideals are easy to compute. No more is necessary than evaluating the $f_i(\ell)$ for $\ell=1,\dots,n+r-1$ and computing the intersection of the corresponding maximal ideals. In this particular situation, the intersection can be computed efficiently. There are two cases to distinguish. Either the point

$$(f_1(n),\dots,f_1(n+r-1),\dots\dots,f_m(n),\dots,f_m(n+r-1))\in\mathbb{K}^{rm} \tag{6.1}$$

is identical to

$$(f_1(\ell),\dots,f_1(\ell+r-1),\dots\dots,f_m(\ell),\dots,f_m(\ell+r-1))$$

for some $\ell<n$. Then, by Cor. 6.10, we have $\mathfrak{a}_n=\operatorname{alg}(f_1(n),\dots,f_m(n))$ and we are done. Or, the point in (6.1) has not been encountered before. In this case, the ideals $\mathfrak{a}_{n-1}$ and $\langle s^i t_j - f_j(n+i) : i,j\rangle$ are comaximal, and their intersection $\mathfrak{a}_n$ is equal to their product (Bourbaki, 1970, I.8.6, Prop. 7). This is advantageous because computing the product of two ideals is less expensive than computing an intersection.

The ideals $\mathfrak{a}_n$ are related to $\operatorname{alg}(f_1(n),\dots,f_m(n))$ via

$$\mathfrak{a}_1\supseteq\mathfrak{a}_2\supseteq\mathfrak{a}_3\supseteq\cdots\supseteq\operatorname{alg}(f_1(n),\dots,f_m(n)).$$

It is easy to see that the sequences $\mathfrak{a}_n$ actually converges to the algebraic part, in the following sense.

Theorem 6.11 Let the ideals $\mathfrak{a}_n$ be defined as above. If $\mathfrak{a} \trianglelefteq \mathbb{K}\{t_1,\dots,t_m\}_{r-1}$ is such that

$$\operatorname{alg}(f_1(n),\dots,f_m(n)) \subseteq \mathfrak{a} \subseteq \mathfrak{a}_n$$

for all $n \in \mathbb{N}$, then $\mathfrak{a} = \operatorname{alg}(f_1(n),\dots,f_m(n))$.

Proof If $p \in \mathfrak{a}_n$ for all $n \in \mathbb{N}$, then, by definition of $\mathfrak{a}_n$, $p \in \bigcap_{i=1}^n \mathfrak{a}_i$ for all $n \in \mathbb{N}$. Hence $p \in \bigcap_{i=1}^\infty \mathfrak{a}_i$, and the claim follows by Theorem 6.8. ■

The ideal sequence $\mathfrak{a}_n$ can be used to determine candidates for algebraic relations. Unless the $f_i(n)$ are all periodic, we will have $\operatorname{alg}(f_1(n),\dots,f_m(n)) \subsetneq \mathfrak{a}_n$ for all N. Every $\mathfrak{a}_n$ is generated by $\operatorname{alg}(f_1(n),\dots,f_m(n))$ and some additional "parasite" polynomials $p_1,\dots,p_\ell \in \mathbb{K}\{t_1,\dots,t_m\}_{r-1}$. As n grows, so does the degree of the parasite polynomials, and therefore, elements of $\mathfrak{a}_n$ with small total degree may be regarded as candidates for polynomials that indeed belong to the algebraic part.

Example 6.12 Consider again the Fibonacci sequence. Define

$$\mathfrak{a}_n := \bigcap_{i=1}^n \langle t - F_i, st - F_{i+1}\rangle \trianglelefteq \mathbb{K}[t, st] \qquad (n \in \mathbb{N}).$$

Consider the Gröbner bases of $\mathfrak{a}_1, \mathfrak{a}_2, \dots$ with respect to degree reverse lexicographic ordering with $t \succ st$.

$$\begin{aligned}
\mathfrak{a}_1 &= \langle t-1, st-1\rangle \\
\mathfrak{a}_2 &= \langle st^2 - 3st + 2, t-1\rangle \\
\mathfrak{a}_3 &= \langle t^2 - 3t + 2, t\,st - 3t - st + 3, st^2 - 3st - 2t + 4\rangle \\
&\ \ \vdots \\
\mathfrak{a}_{13} &= \langle 2573916t^3 st - 1655101t^2 st^2 + (\dots 12 \text{ more terms}\dots), \\
&\qquad 1286958t^4 + 368143t^2 st^2 + (\dots 12 \text{ more terms}\dots), \\
&\qquad 428986st^5 + 876600896999484155t^2 st^2 + (\dots 12 \text{ more terms}\dots), \\
&\qquad 2198835588991693436t\,st^4 - 1357584802697081162st^5 + (\dots 13 \text{ more terms}\dots), \\
&\qquad 428986t^2 st^3 + 3348738948723536 53t^2 st^2 + (\dots 12 \text{ more terms}\dots)\rangle \\
\mathfrak{a}_{14} &= \langle t^4 + 2t^3 st - t^2 st^2 - 2t\,st^3 + st^4 - 1, \\
&\qquad 87st^5 - 45128478224686650t^3 st + (\dots 13 \text{ more terms}\dots), \\
&\qquad 148644086475454t\,st^4 - 91853803918593st^5 + (\dots 14 \text{ more terms}\dots), \\
&\qquad 87t^2 st^3 - 17236804823885190t^3 st + (\dots 13 \text{ more terms}\dots), \\
&\qquad 87t^3 st^2 - 10652702566841358t^3 st + (\dots 13 \text{ more terms}\dots)\rangle
\end{aligned}$$

The polynomial $t^4 + 2t^3 st - t^2 st^2 - 2t\,st^3 + st^4 - 1$, which generates $\operatorname{alg}(F_n)$ according to Example 6.6, appears first explicitly in the Gröbner basis of $\mathfrak{a}_{14}$. The other polynomials are parasite. It seems to be typical that parasite polynomials have more ugly coefficients than the actual generators of the algebraic part.

The fact that the generator of $\operatorname{alg}(F_n)$ appeared in the approximation ideals $\mathfrak{a}_n$ in the example above was not a coincidence. We will show next that generators of the algebraic part will always appear in the Gröbner basis of $\mathfrak{a}_n$ with respect to an appropriate term order when n is sufficiently large. Let $\mathbb{K}\{t_1,\dots,t_m\}_{r-1}$ be equipped with a term order $\prec$, which is such that for every term τ, there are only finitely many terms $\tilde{\tau}$ with $\tilde{\tau} \prec \tau$. For example, degree orderings have this property, but the lexicographic ordering does not.

Theorem 6.13 Let $f_1(n),\dots,f_m(n)$ be solutions of a normal admissible system S of order r. For $n \in \mathbb{N}$, define

$$\mathfrak{a}_n := \bigcap_{\ell=1}^{n} \langle s^i t_j - f_j(\ell+i) : i = 0,\dots,r-1,\ j = 1,\dots,m\rangle.$$

Then for all sufficiently large $n \in \mathbb{N}$, the Gröbner basis G of $\mathfrak{a} := \operatorname{alg}(f_1(n),\dots,f_m(n))$ with respect to $\prec$ appears as a subset of the Gröbner basis G_n of $\mathfrak{a}_n$ with respect to $\prec$.

Proof For $\mathfrak{a} = \{0\}$ there is nothing to prove. If $\dim \mathfrak{a} = 0$, then $\mathfrak{a} = \mathfrak{a}_n$ for sufficiently large n, so there is nothing to prove in this case either.

Now exclude these cases from consideration. The chain $\mathfrak{a}_1 \supseteq \mathfrak{a}_2 \supseteq \mathfrak{a}_3 \supseteq \cdots \supseteq \mathfrak{a}$ implies the chain

$$\operatorname{LT}(\mathfrak{a}_1) \supseteq \operatorname{LT}(\mathfrak{a}_2) \supseteq \operatorname{LT}(\mathfrak{a}_3) \supseteq \cdots \supseteq \operatorname{LT}(\mathfrak{a}),$$

which in turn implies $\bigcap_{i=1}^{\infty} \operatorname{LT}(\mathfrak{a}_i) \supseteq \operatorname{LT}(\mathfrak{a}) = \operatorname{LT}(\bigcap_{i=1}^{\infty} \mathfrak{a}_i)$.

Also the other inclusion holds. Let $\tau \in \bigcap_{i=1}^{\infty} \operatorname{LT}(\mathfrak{a}_i)$. Then for all $n \in \mathbb{N}$ there exists $p_n \in \mathfrak{a}_n$ with $\operatorname{LT}(p_n) = \tau$. We may assume that the p_n are chosen such that $p_{n+1} \neq p_n$ only if $p_n \notin \mathfrak{a}_{n+1}$. By discarding some of the ideals $\mathfrak{a}_n$, we may assume without loss of generality that $p_n \notin \mathfrak{a}_{n+1}$ for all $n \in \mathbb{N}$. By r_n denote the reductum of p_n modulo $\mathfrak{a}_{n+1}$. Then we have $\operatorname{LT}(r_n) \notin \operatorname{LT}(\mathfrak{a}_{n+1})$ but $\operatorname{LT}(r_n) \in \operatorname{LT}(\mathfrak{a}_n)$, because $\mathfrak{a}_{n+1} \subseteq \mathfrak{a}_n$ implies $r_n \in \mathfrak{a}_n$. With $T := \{\, \tilde{\tau} : \tilde{\tau} \prec \tau \,\}$ we get the chain

$$\operatorname{LT}(\mathfrak{a}_n) \cap T \supsetneq \operatorname{LT}(\mathfrak{a}_{n+1}) \cap T \supsetneq \operatorname{LT}(\mathfrak{a}_{n+2}) \cap T \supsetneq \cdots,$$

which cannot be infinite because T is finite by assumption on the term order. It follows that the set $\{\, p_n : n \in \mathbb{N} \,\}$ is finite, and hence there must be a p with $p = p_n$ for infinitely many $n \in \mathbb{N}$. By the inclusions $\mathfrak{a}_n \supseteq \mathfrak{a}_{n+1}$ ($n \in \mathbb{N}$) we obtain that $p \in \mathfrak{a}_n$ for all $n \in \mathbb{N}$, and therefore $p \in \mathfrak{a}$ and $\tau = \operatorname{LT}(p) \in \operatorname{LT}(\mathfrak{a})$. We have thus established

$$\bigcap_{n=1}^{\infty} \operatorname{LT}(\mathfrak{a}_n) = \operatorname{LT}(\bigcap_{n=1}^{\infty} \mathfrak{a}_n). \tag{6.2}$$

Now consider the Gröbner basis G of $\mathfrak{a}$. Let $\tau := \max \operatorname{LT}(G)$ and $T := \{\, \tilde{\tau} : \tilde{\tau} \preccurlyeq \tau \,\}$. As a consequence of (6.2), we have $\operatorname{LT}(G_n) \cap T = \operatorname{LT}(G)$ for sufficiently large n. For these n, we have $G \subseteq G_n$: Assume for the contrary the existence of some $p \in G \setminus G_n$. Then, by $\operatorname{LT}(G) \subseteq \operatorname{LT}(G_n)$, there is some $\tilde{p} \in G_n$, $p \neq \tilde{p}$ with $\operatorname{LT}(p) = \operatorname{LT}(\tilde{p})$. Then

$$q := \operatorname{LC}(p)\tilde{p} - \operatorname{LC}(\tilde{p})p \in \mathfrak{a}_n \setminus \mathfrak{a}$$

and some divisor of $\operatorname{LT}(q) \prec \operatorname{LT}(p) \preccurlyeq \tau$ would belong to G_n, in contradiction to $\operatorname{LT}(G_n) \cap T = \operatorname{LT}(G)$. ∎

Summarizing, we are able to find algebraic dependencies by computing a Gröbner basis of the product of sufficiently many maximal ideals corresponding to evaluations of the sequences at hand. The procedure delivers ideals which containing all algebraic dependencies, plus some additional parasite polynomials. By the zero equivalence algorithm (Algorithm 4.2), parasite polynomials can automatically be distinguished from actual algebraic relations.

6.4 Approximation from Below

We now turn to a different way of finding algebraic dependencies, which can be formulated as an approximation of the algebraic part "from below". Again, let $f_1(n),\dots,f_m(n)$ be solutions of a normal admissible system S. Suppose that a finite set $P=\{p_1,\dots,p_\ell\}\subseteq\mathbb{K}\{t_1,\dots,t_m\}$ is given, and assume we are interested only in algebraic relations p of $f_1(n),\dots,f_m(n)$ which are $\mathbb{K}$-linear combinations of the p_i. These relations form a finite dimensional vector space over $\mathbb{K}$. A basis of this space can be computed by an undetermined ansatz, as described next.

Algorithm 6.14 (Computing Algebraic Dependencies of Prescribed Shape)
Input: Admissible sequences $f_1(n),\dots,f_m(n)$, defined by a normal admissible system S of order r and initial values, a set $P=\{p_1,\dots,p_\ell\}\subseteq\mathbb{K}\{t_1,\dots,t_m\}$
Output: A basis of the $\mathbb{K}$-vector space of all algebraic dependencies of $f_1(n),\dots,f_m(n)$ of the form $c_1p_1+\dots+c_\ell p_\ell$ with $c_i\in\mathbb{K}$

$S:=S\cup\{f_{m+1}(n+1)=f_{m+1}(n),\dots,f_{m+\ell}(n+1)=f_m(n)\}$
ansatz $:=t_{m+1}p_1+t_{m+2}p_2+\dots+t_{m+\ell}p_\ell$
Apply Algorithm 4.1 to ansatz, obtaining a number N.
Use Algorithm 3.9 to evaluate each p_i for $n=1,2,\dots,N$, obtaining values $v_{n,i}\in\mathbb{K}$
Compute a basis $B\subseteq\mathbb{K}^\ell$ of the solution space of the linear system

$$\begin{pmatrix} v_{1,1} & \cdots & v_{1,\ell}\\ \vdots & \ddots & \vdots\\ v_{N,1} & \cdots & v_{N,\ell}\end{pmatrix}\begin{pmatrix} t_{m+1}\\ \vdots\\ t_{m+\ell}\end{pmatrix}=0$$

<u>**return**</u> $\{b_1p_1+\dots+b_\ell p_\ell : b=(b_1,\dots,b_\ell)\in B\}$

Theorem 6.15 Algorithm 6.14 is correct.

Proof Let $\varphi\colon\mathbb{K}\{t_1,\dots,t_m\}\to\mathbb{K}^{\mathbb{N}}$ be the difference homomorphism mapping t_i to $f_i(n)$ and elements of $\mathbb{K}$ to the corresponding constant sequences. Let $p=\sum_{i=1}^{\ell}c_ip_i$ for some $c_1,\dots,c_\ell\in\mathbb{K}$. Suppose that $p\in\operatorname{ann}(f_1(n),\dots,f_m(n))$. Then $\varphi(p)$ is the zero sequence. In particular, this sequence is zero for $n=1,2,\dots,N$, and hence the coefficient vector $(c_1,\dots,c_\ell)$ must belong to the solution space of the linear system.

On the other hand, suppose that p is a linear combination of the polynomials computed by the algorithm. Then the sequence $\varphi(p)$ vanishes for $n=1,2,\dots,N$. By the specification of Algorithm 4.1, it follows that $\varphi(p)$ vanishes for all $n\in\mathbb{N}$, and hence $p\in\operatorname{ann}(f_1(n),\dots,f_m(n))$. ∎

It is pretty inconvenient from a complexity viewpoint to implement Algorithm 6.14 as described above. The additional difference variables needed to represent the coefficients in the ansatz blow up the difference ring and make the Gröbner basis computations infeasible. Typically, N will be roughly of the same size as ℓ, and this is much larger than the N required for a particular field elements in place of the ansatz indeterminates. A much more efficient variant proceeds as follows. Instead of executing line 3, simply set $N = \ell$, then execute lines 4f and form the set

$$\{\sum_{i=1}^{\ell} b_i\tau_i : b = (b_1,\dots,b_\ell) \in B\}.$$

Apply the zero equivalence tester (Algorithm 4.2) to all elements of this set. If the algorithm yields True for all elements, return this set as result. Otherwise, increment N and go to line 4. Correctness of this variant follows from Theorem 6.15, and for termination it is only necessary that there exists a number N such that all solutions of the linear system correspond to the zero sequence. Existence is guaranteed, in Algorithm 6.14 such an N is actually computed.

The linear systems in Algorithm 6.14 can become quite large. Typically, these systems are dense, so that heuristics exploiting sparsity cannot be used for speeding up the solving of the system. A drastic speed-up can be achieved by mapping the system homomorphically to a finite field $\mathbb{Z}_p$ and solve it there. Classically, one would recover the solution of the original system from several homomorphic images via Chinese remaindering and/or Newton interpolation, but this might be tedious if $\mathbb{K}$ is complicated. An alternative approach is to interpret the basis vectors in the homomorphic image of the solution space only as a *structure set* determining the polynomials which are actually in relation with each other. A structure set is a small set of terms which are conjectured to be dependent. For every structure set, an individual system can be set up over $\mathbb{K}$, which can be solved to determine the actual coefficients of the relation. This technique is widely used in symbolic summation.

Structure set considerations can reduce the size of the linear systems considerably, but if $\mathbb{K} = \mathbb{Q}(x_1,\dots,x_s)$, solving these systems can still be the runtime bottleneck of the entire computation, even though such systems can be solved in polynomial time using Bareiss elimination (Geddes *et al.*, 1992, Section 9.3). For the case $\mathbb{K} = \mathbb{Q}(x)$, Storjohann and Villard (2005) give an efficient algorithm that outperforms Bareiss' elimination both in practice and in theory. The case $s > 1$ remains challenging.

In order to approximate an algebraic part by subideals, we employ the algorithm just described. The whole polynomial ring $\mathbb{K}\{t_1,\dots,t_m\}_{r-1}$ can be viewed as a vector space over $\mathbb{K}$ of infinite dimension. The terms $\tau = \prod_{i=0}^{r-1}\prod_{j=1}^{m}(s^i t_j)^{e_{i,j}}$ form a basis of this space. The algebraic part $\operatorname{alg}(f_1(n),\dots,f_m(n))$ can be approximated from below by applying Algorithm 6.14 to larger and larger sets of terms. For $d \in \mathbb{N}$, let P_d denote the set of all terms τ with $\deg\tau \le d$. If $\mathfrak{b}_d \trianglelefteq \mathbb{K}\{t_1,\dots,t_m\}_{r-1}$ denotes the ideal generated by the polynomials which Algorithm 6.14 returns for $f_1(n),\dots,f_m(n)$ and P_d then

$$\mathfrak{b}_1 \subseteq \mathfrak{b}_2 \subseteq \mathfrak{b}_3 \subseteq \cdots \subseteq \operatorname{alg}(f_1(n),\dots,f_m(n))$$

provides an approximation of the algebraic part from below. Using Hilbert's basis theorem, it can be shown that the algebraic part is eventually identical to $\mathfrak{b}_d$ for sufficiently large d. This is what makes these ideals more interesting than the superideals $\mathfrak{a}_i$ discussed in the previous section.

Theorem 6.16 Let $f_1(n),\dots,f_m(n)$ be solutions of a normal admissible system S of order r. Let P_d be the set of all terms $\tau = \prod_{i=0}^{r-1}\prod_{j=1}^{m}(s^i t_j)^{e_{i,j}}$ with $\deg\tau \le d$ and $\mathfrak{b}_d \trianglelefteq \mathbb{K}\{t_1,\dots,t_m\}_{r-1}$ be the ideal generated by the polynomials, which Algorithm 6.14 computes for P_d and $f_1(n),\dots,f_m(n)$. Then

$$\mathfrak{b}_1 \subseteq \mathfrak{b}_2 \subseteq \mathfrak{b}_3 \subseteq \cdots \subseteq \operatorname{alg}(f_1(n),\dots,f_m(n)),$$

and $\mathfrak{b}_d = \operatorname{alg}(f_1(n),\dots,f_m(n))$ for sufficiently large d.

Proof The inclusions $\mathfrak{b}_d \subseteq \mathfrak{b}_{d+1}$ ($d \ge 1$) directly follow from the corresponding inclusions of the vector spaces of Algorithm 6.14, which are evident. Furthermore, $\mathfrak{b}_d \subseteq \operatorname{alg}(f_1(n),\dots,f_m(n))$ ($d \ge 1$) directly follows from the correctness of Algorithm 6.14.

The algebraic part $\operatorname{alg}(f_1(n),\dots,f_m(n))$ is finitely generated as ideal of $\mathbb{K}\{t_1,\dots,t_m\}_{r-1}$, let $B = \{b_1,\dots,b_\ell\}$ be a basis. For $d_0 = \max_{i=1}^{\ell}\deg b_i$, we have $b_i \in \mathfrak{b}_{d_0}$ for all i, and hence

$$\operatorname{alg}(f_1(n),\dots,f_m(n)) \subseteq \mathfrak{b}_{d_0}.$$

It follows that $\mathfrak{b}_d = \operatorname{alg}(f_1(n),\dots,f_m(n))$ for $d \ge d_0$. ■

According to the theorem above, we can compute the algebraic part—and hence the annihilating ideal—of some admissible sequences $f_1(n),\dots,f_m(n)$ by approximating it sufficiently accurate from below. In order to devise a complete algorithm for computing generators of the algebraic part, it is now sufficient to know a general criterion by which it could be decided whether an approximation at hand is already complete. We do not know of any such criterion, and in view of Theorem 6.5 it does not seem likely that such a criterion could be found.

The set of terms of a prescribed total degree d grows exponentially in d. Therefore, the linear systems of Algorithm 6.14 become quite big already for moderate size of d. It is advisable to spend some extra efforts in order to keep the ansatz polynomials as small as possible. One way of doing so is by leaving out redundant terms. If bases of the ideals $\mathfrak{b}_1,\mathfrak{b}_2,\dots$ are computed in order, then in the computation of $\mathfrak{b}_d$ only relations are of interest which do not already belong to $\mathfrak{b}_{d-1}$. The following algorithm computes a basis for $\mathfrak{b}_d$ incrementally, keeping the ansatz polynomial of Algorithm 6.14 small.

Algorithm 6.17 (Approximation of the Algebraic Part from Below)
Input: Admissible sequences $f_1(n),\dots,f_m(n)$, defined by a normal admissible system S of order r and initial values, a number $D \in \mathbb{N}$
Output: A basis for the ideal $\mathfrak{b}_D \trianglelefteq \mathbb{K}\{t_1,\dots,t_m\}$ generated by all algebraic dependencies of $f_1(n),\dots,f_m(n)$ of total degree at most D
Assumption: $\mathbb{K}\{t_1,\dots,t_m\}_{r-1}$ is equipped with a total degree term ordering

$G = \emptyset$
<u>for</u> $d = 1$ **<u>to</u>** D **<u>do</u>**
  Let T be the set of all terms τ in $\mathbb{K}\{t_1,\dots,t_m\}_{r-1}$ with $\deg\tau \le d$
  Delete from T all terms which are divisible by some term in $\mathrm{LT}(G)$
  Apply Algorithm 6.14 to $f_1(n),\dots,f_m(n)$ and T, obtaining $B \subseteq \mathbb{K}\{t_1,\dots,t_m\}_{r-1}$
  $G := \text{GröbnerBasis}(G \cup B)$
<u>return</u> G

Theorem 6.18 Algorithm 6.17 is correct.

Proof Let $G \subseteq \mathbb{K}\{t_1,\ldots,t_m\}_{r-1}$ be as computed by the algorithm. It is clear that $\langle G\rangle \subseteq \mathfrak{b}_d$ by the correctness of Algorithm 6.14. Without line 4, also the inclusion $\langle G\rangle \supseteq \mathfrak{b}_d$ would be clear. We have to prove that line 4 does not violate this inclusion.

Assume for the contrary that it does. Then there exists an algebraic dependency $p \in \mathfrak{b}_d \setminus \mathfrak{b}_{d-1}$ with $\operatorname{LT}(p) \in \langle \operatorname{LT}(G)\rangle = \operatorname{LT}(\mathfrak{b}_{d-1})$. We may assume that p is minimal in the sense that there is no $\tilde{p} \in \mathfrak{b}_d \setminus \mathfrak{b}_{d-1}$ with $\operatorname{LT}(\tilde{p}) \in \langle \operatorname{LT}(G)\rangle$ and $\operatorname{LT}(\tilde{p}) \prec \operatorname{LT}(p)$. By $\operatorname{LT}(p) \in \langle \operatorname{LT}(G)\rangle$, there exists $q \in \mathfrak{b}_{d-1}$ with $\operatorname{LT}(p) = \operatorname{LT}(q)$. By $\mathfrak{b}_{d-1} \subseteq \mathfrak{b}_d$, we also have $q \in \mathfrak{b}_d$, and also

$$\tilde{q} := \operatorname{LC}(q)p - \operatorname{LC}(p)q \in \mathfrak{b}_d.$$

But $\operatorname{LT}(\tilde{q}) \prec \operatorname{LT}(p)$, and by minimality assumption on p, it follows that $\tilde{q} \in \mathfrak{b}_{d-1}$, which in turn implies that

$$p = \frac{1}{\operatorname{LC}(q)}(\operatorname{LC}(p)q + \tilde{q}) \in \mathfrak{b}_{d-1},$$

in contradiction to the choice of p. ∎

If some algebraic dependencies are already known, then G should be initialized to a Gröbner basis of those in line 1.

An additional refinement of the ansatz polynomial is possible by taking into account knowledge about close superideals of the algebraic part, as they appeared in Section 6.3. Suppose that $\mathfrak{a} \supseteq \operatorname{alg}(f_1(n),\ldots,f_m(n))$, and basis elements $q_1,\ldots,q_\ell$ of $\mathfrak{a}$ are known. Every p in the algebraic part also belongs to $\mathfrak{b}$, and hence there are $p_1,\ldots,p_\ell \in \mathbb{K}\{t_1,\ldots,t_m\}_{r-1}$ such that

$$p = p_1q_1 + p_2q_2 + \cdots + p_\ell q_\ell.$$

Instead of making an ansatz for p directly, one might prefer to make an ansatz for the cofactors p_i. In this case, in order not to unnecessarily increase the number of terms in the ansatz polynomial, one has to take into account the syzygies of the generators $q_1,\ldots,q_\ell$. This leads to the following refinement of Algorithm 6.17.

Algorithm 6.19 (Approximation of the Algebraic Part from Below)
Input: Admissible sequences $f_1(n),\ldots,f_m(n)$, defined by a normal admissible system S of order r and initial values, a number $D \in \mathbb{N}$, and difference polynomials $q_1,\ldots,q_\ell \in \mathbb{K}\{t_1,\ldots,t_m\}_{r-1}$ such that $\langle q_1,\ldots,q_\ell\rangle \supseteq \operatorname{alg}(f_1(n),\ldots,f_m(n))$
Output: A basis for the ideal $\mathfrak{b}_D \trianglelefteq \mathbb{K}\{t_1,\ldots,t_m\}$ generated by all algebraic dependencies of $f_1(n),\ldots,f_m(n)$ of total degree at most D
Assumption: $\mathbb{K}\{t_1,\ldots,t_m\}_{r-1}$ and $(\mathbb{K}\{t_1,\ldots,t_m\}_{r-1})^\ell$ are equipped with a total degree term ordering, the q_i form a Gröbner basis in $\mathbb{K}\{t_1,\ldots,t_m\}_{r-1}$, ε_i is the i-th unit vector in the module $(\mathbb{K}\{t_1,\ldots,t_m\}_{r-1})^\ell$

```
G = ∅
M = GröbnerBasis(Syz(q_1, ..., q_ℓ))
for d = 1 to D do
   for i = 1 to ℓ do
```

```
5       Let T_i be the set of all terms τ in K{t_1,...,t_m}_{r-1} with deg τ ≤ d
6       Delete from T_i all terms τ such that τε_i is not irreducible w.r.t. M
7     T := { p_i τ : τ ∈ T_i, i = 1,...,ℓ }
8     Apply Algorithm 6.14 to f_1(n),...,f_m(n) and T, obtaining B ⊆ K{t_1,...,t_m}_{r-1}
9     G := G ∪ B
10  return G
```

Theorem 6.20 Algorithm 6.19 is correct.

Proof We have to show that no dependencies are lost by lines 5f.

Line 5. Let $p \in \langle q_1,\dots,q_\ell\rangle \trianglelefteq \mathbb{K}\{t_1,\dots,t_m\}_{r-1}$ be such that $\deg p = d$. Then there exist cofactors $p_1,\dots,p_\ell$ with $p = p_1q_1+\dots+p_\ell q_\ell$. As the q_i form by assumption a Gröbner basis with respect to a degree order, p reduces to zero modulo $q_1,\dots,q_\ell$, and therefore the p_i can be chosen such that $\deg p_i \leq \deg p - \deg q_i \leq d$.

Line 6. Assume that $\tau\varepsilon_i$ is reducible w.r.t. M. Then there exist cofactors $c_j \in \mathbb{K}\{t_1,\dots,t_m\}_{r-1}$ with $\mathrm{LT}(c_j) \prec \mathrm{LT}(\tau)$ $(j = 1,\dots,\ell)$ such that $\tau q_i = c_1q_1+\dots+c_\ell q_\ell$. If a dependency $p = p_1q_1+\dots+p_\ell q_\ell \in \mathfrak{b}_d$ with $\deg p = d$ is such that p_i involves τ, then we have the alternative representation

$$p = (p_1+c_1)q_1+\dots+(p_\ell+c_\ell)q_\ell$$

in which the i-th cofactors is free of τ. Starting with the greatest reducible term appearing in the cofactors, one can thus inductively eliminate them all using the corresponding syzygies. Eventually, this will give a representation of p with cofactors only involving irreducible terms, and hence reducible terms need not be taken into account in the ansatz polynomial. ■

Suitable polynomials $q_1,\dots,q_\ell$ needed by Algorithm 6.19 are generators of the ideals $\mathfrak{a}_n$ discussed in the previous section. It turns out that taking for the q_i generators of the n-th approximation from above, the set T of line 7 has approximately n elements less than the corresponding set in Algorithm 6.17. This decrease does, however, in most cases not lead to an acceleration that equalizes the additional time needed to compute the syzygy module. In other words, Algorithm 6.17 performs better than Algorithm 6.19 in most cases, and it should therefore be preferred. The software described in Chapter 9 contains implementations of both algorithms.

6.5 Examples and Applications

We have discussed in this chapter ways to obtain algebraic dependencies between some given admissible sequences. Knowledge of an ideal of algebraic dependencies is useful for answering various questions about the relationships of admissible sequences at hand. In Chapter 8 below we will use knowledge about algebraic dependencies for computing closed forms of sums involving admissible sequences. Summation is no doubt the main application, but there are also other problems which can be solved with the aid of algebraic dependencies. Some examples are given in this sections, more examples can be found in the following chapters.

Given an admissible sequence $f(n)$, it might be of interest to know whether there exists a rational function r such that

$$f(n) = r(g_1(n),\dots,g_m(n)) \qquad (n \geq 1) \tag{6.3}$$

for some other given admissible sequences $g_1(n),\dots,g_m(n)$. In this case, we say that $f(n)$ is represented *in terms of* the $g_i(n)$. The question as to whether a representation of $f(n)$ in terms of the $g_i(n)$ exists, and if so, to compute it, is called the *representation problem.* Indefinite summation is the most commonly considered special case of this problem ("express the sum in terms of the summand").

By multiplying (6.3) with its denominator, we obtain an algebraic dependency of $f(n)$ and $g_1(n),\dots,g_m(n)$. Conversely, every algebraic dependency of this form provides a solution to the problem. Algorithm 6.17 can be turned into a semidecision procedure for the representation problem: Compute bases for the ideals $\mathfrak{b}_1,\mathfrak{b}_2,\dots$. For each basis $\mathfrak{b}_d$, eliminate all variables corresponding to $f(n+1), f(n+2),\dots$. For the resulting basis, compute a Gröbner basis with respect to a lexicographic ordering that weights the variable corresponding to $f(n)$ heaviest. A solution to the representation problem where numerator and denominator of the rational function have total degree at most d and $d-1$, respectively, exists if and only if a corresponding polynomial appears in the resulting Gröbner basis. This procedure terminates and returns a solution of the representation problem if and only if a solution exists. Otherwise, it does not terminate.

Example 6.21 Rabinowitz (2003) has proposed the following problem. Let $u(n)$ be defined via the nonlinear recurrence

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

Express $u(n)$ in terms of Fibonacci and/or Lucas numbers.

The ideal computed by Algorithm 6.17 for total degree 3 contains difference polynomials which immediately give rise to the representations

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

More generally than in the representation problem, we might be interested in an algebraic dependency of a special shape, for instance, a linear recurrence with coefficients from a prescribed domain.

Example 6.22 Consider the sum

$$s(n) := \sum_{k=1}^{n} \frac{2k+1}{k+1} P_k^{(1,-1)}(x)$$

where $P_n^{(1,-1)}(x)$ denotes the nth Jacobi polynomial (p. 11). We have the obvious recurrence

$$s(n+1) = s(n) + \frac{2(n+1)+1}{(n+1)+1} P_{n+1}^{(1,-1)}(x),$$

but does there also exist a linear recurrence for $s(n)$ whose coefficients only involve n?

From the closure of P-finite sequences under indefinite summation, it follows that such a recurrence must exist, and we can compute one with Algorithm 6.17. The software package described in Chapter 9 delivers the inhomogeneous second order recurrence

$$\begin{aligned}&(n+3)(2n+3)s(n+2)-((2n+3)(2n+5)x-1)s(n+1)+(n+1)(2n+5)s(n)\\=&(x+1)(2n+3)(2n+5)\end{aligned}$$

The packages gfun (Salvy and Zimmermann, 1994) and Mallinger's package (Mallinger, 1996) both give a homogeneous recurrence of order 3 as result.

Example 6.23 Recall the notion of a Somos sequence (Example 2.11.(6), p. 13). We consider the Somos-4 sequence C_n, defined by

$$C_{n+2}=\frac{1}{C_{n-2}}(C_{n-1}C_{n+1}+C_n^2)\quad(n\geq 2),\qquad C_{-2}=C_{-1}=C_0=C_1=1.$$

In Example 4.9.(6) (p. 34) we have verified some algebraic relations of C_n by van der Poorten (2004).

The ideal generated by all quadratic dependencies between $C_{n-4},C_{n-3},\dots,C_{n+3},C_{n+4}$ can be computed by means of Algorithm 6.17. The resulting basis is long, and we do not reproduce it here.

Having the basis at hand, it is an easy matter to find out which higher order Somos-like difference equations are satisfied by the sequence C_n. Besides the recurrence from the definition, we obtain

$$\begin{aligned}C_{n+3}C_{n-2}&=5C_{n+1}C_n-C_{n+2}C_{n-1}\\C_{n+3}C_{n-3}&=5C_n^2+C_{n+1}C_{n-1}\\C_{n+4}C_{n-3}&=5C_nC_{n+1}+4C_{n+1}C_n\\C_{n+4}C_{n-4}&=-4C_n^2+25C_{n+1}C_{n-1}.\end{aligned}$$

This list is exhaustive in the sense that every other Somos-like relation of order at most 8 is a $\mathbb{Q}$-linear combination of those. (Note that the third relation does not appear in van der Poorten's list.)

7 A Computable Subclass [2]

In the previous chapter, we have been concerned with algebraic dependencies of admissible sequences. We have described methods to enumerate a basis for the ideal of algebraic dependencies, and we have argued that it is presumably very difficult to find an algorithm which actually computes a complete basis for all dependencies in a finite number of steps. What makes this problem so difficult is that the class of admissible sequences is very large.

In the present chapter, we restrict our attention to a small subclass of the class of admissible sequences, namely the class of C-finite sequences, and we show that in this case, it is actually possible to algorithmically compute a basis for all algebraic dependencies.

7.1 C-Finite Sequences

According to Def. 2.10.(3), a sequence $f(n)$ in $\mathbb{K}$ is called C-finite if it satisfies a homogeneous linear recurrence with constant coefficients,

$$f(n+r) = a_0 f(n) + a_1 f(n+1) + \cdots + a_{r-1} f(n+r-1) \qquad (n \geq 1).$$

C-finite sequences have been studied deeply (Everest *et al.*, 2003). One crucial and elementary fact about C-finite recurrences is that they always admit a closed form solution. Let $\phi_1, \ldots, \phi_s \in \bar{\mathbb{K}}$ be the roots of the *characteristic polynomial*

$$c(x) := x^r - a_0 - a_1 x - \cdots - a_{r-1} x^{r-1}$$

of $f(n)$, and let e_i be the multiplicity of ϕ_i ($i = 1, \ldots, s$). Then we have

$$f(n) = p_1(n)\phi_1^n + \cdots + p_s(n)\phi_s^n \qquad (n \geq 1) \tag{7.1}$$

for certain polynomials $p_i \in \mathbb{K}[n]$ of degree at most e_i.

This closed form representation can obviously be computed easily from a given recurrence and initial values. We will describe in the following sections how the algebraic dependencies between some sequences $f_1(n), \ldots, f_m(n)$ can be computed by using these closed forms. It is not necessary that the $f_i(n)$ are independently defined by a C-finite recurrence. The only important requirement is that they admit a closed form representation of the form (7.1), and that they are defined by a system of recurrences from which such a representation can be computed. This is, for instance, possible for C-finite systems of recurrences, which we define as follows.

Definition 7.1 A system $S = \{rec_1, \ldots, rec_m\}$ of difference equations for $f_1(n), \ldots, f_m(n)$ is

[2] The results in this chapter originate from a collaboration with B. Zimmermann (Kauers and Zimmermann, 2005a).

called *C-finite*, if every rec_i has the form

$$\begin{aligned} f_i(n+r_i) = {} & a_{0,1}f_1(n) + a_{1,1}f_1(n+1) + \cdots + a_{r_1-1,1}f_1(n+r_i-1) \\ & + a_{0,2}f_2(n) + a_{1,2}f_2(n+1) + \cdots + a_{r_2-1,2}f_2(n+r_i-1) \\ & + \cdots \\ & + a_{0,m}f_m(n) + a_{1,m}f_m(n+1) + \cdots + a_{r_m-1,m}f_m(n+r_i-1) \qquad (n \geq 1) \end{aligned}$$

for certain fixed orders $r_i \in \mathbb{N}_0$ $(i = 1, \ldots, m)$ and constants $a_{i,j} \in \mathbb{K}$, not all $a_{0,i}$ being zero.

As a special case, any system $S = \{rec_1, \ldots, rec_m\}$ in which rec_i is a C-finite recurrence for $f_i(n)$ is such a system. The general case is referred to as a *coupled* system in the literature. For P-finite coupled systems there are "uncoupling" algorithms known which transform the system into an equivalent one consisting of independent P-finite recurrences only (Zürcher, 1994; Gerhold, 2002). These algorithms could also be applied to reduce C-finite systems to systems of independent C-finite recurrences, which then can be solved using the characteristic polynomial. A more direct way is given by the following algorithm.

Algorithm 7.2 (Solving C-finite Systems of Difference Equations)
Input: A C-finite system $S = \{rec_1, \ldots, rec_m\}$ for $f_1(n), \ldots, f_m(n)$, and initial values
Output: For each $f_i(n)$ a representation as linear combination of exponential terms with polynomial coefficients
Assumption: rec_i has the order $r_i \geq 0$

1 Write $x_n = (f_1(n), \ldots, f_1(n+r_1-1), \ldots\ldots, f_m(n), \ldots, f_m(n+r_m-1))$ for short.
2 Let $M \in \mathbb{K}^{(r_1+\cdots+r_m)\times(r_1+\cdots+r_m)}$ be such that $x_{n+1} = Mx_n$
3 Compute a Jordan decomposition $M = SJS^{-1}$ for M
4 Let J_n be the matrix obtained by J upon replacing each of its Jordan blocks

$$\begin{pmatrix} \phi & 1 & 0 & \cdots & 0 \\ 0 & \phi & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & 1 \\ 0 & \cdots & \cdots & 0 & \phi \end{pmatrix} \in \bar{\mathbb{K}}^{d\times d} \quad \text{by the block} \quad \begin{pmatrix} \phi^n & n\phi^{n-1} & \binom{n}{2}\phi^{n-2} & \cdots & \binom{n}{d-1}x^{n-d+1} \\ 0 & \phi^n & n\phi^{n-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \binom{n}{2}\phi^{n-2} \\ \vdots & & \ddots & \ddots & n\phi^{n-1} \\ 0 & \cdots & \cdots & 0 & \phi^n \end{pmatrix}$$

5 Compute the vector $c = SJ_nS^{-1}x_1 \in \mathbb{K}^{r_1+\cdots+r_m}$
6 <u>**return**</u> $\{\, f_i(n) = c_{r_1+\cdots+r_{i-1}+1} : i = 1, \ldots, m \,\}$

The coefficients of M in line 2 can be read of directly from the recurrences in S. The components of the vector x_1 needed in line 5 are the initial values of the $f_i(n)$. The symbol n is always understood symbolic. Note that in step 4, the value of d is known at runtime, so that the binomials appearing in $\tilde{J}$ are just polynomials in n of degree up to $d-1$.

Theorem 7.3 Algorithm 7.2 is correct.

Proof The only thing to show is that $M^n = SJ_nS^{-1}$ is true for all $n \in \mathbb{N}$. By

$$M^n = (SJS^{-1})^n = (SJS^{-1})(SJS^{-1})\cdots(SJS^{-1}) = SJ^nS^{-1} \qquad (n \in \mathbb{N})$$

it remains to show $J_n = J^n$. By the block structure of J, it suffices to show that the nth power of the matrix on the left in step 4 is equal to the matrix on the right. Denote the left matrix by B. For $n = 1$, the identity is evident. Suppose it is holds for some $n \in \mathbb{N}$, then for $n+1$ we have

$$B^{n+1} = \begin{pmatrix} \phi^n & n\phi^{n-1} & \binom{n}{2}\phi^{n-2} & \cdots & \binom{n}{d-1}x^{n-d+1} \\ 0 & \phi^n & n\phi^{n-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \binom{n}{2}\phi^{n-2} \\ \vdots & & \ddots & \ddots & n\phi^{n-1} \\ 0 & \cdots & \cdots & 0 & \phi^n \end{pmatrix} \begin{pmatrix} \phi & 1 & 0 & \cdots & 0 \\ 0 & \phi & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & 1 \\ 0 & \cdots & \cdots & 0 & \phi \end{pmatrix} = ((b_{i,j}))_{i,j=1}^d$$

where

$$b_{i,j} = \begin{cases} \binom{n}{j-i-1}\phi^{n-j+1} + \binom{n}{j-i}\phi^{n-j}\phi = \binom{n+1}{j-1}\phi^{(n+1)-j+1} & \text{if } i < j \\ \phi^{n+1} & \text{if } i = j \\ 0 & \text{if } i > j \end{cases},$$

as desired. ∎

7.2 Algebraic Dependencies of Exponentials

A sequence of the form $f(n) = \phi^n$ for some $\phi \in \mathbb{K}$ fixed is called an *exponential.* In this section, we study difference ideals of the form

$$\operatorname{ann}(n, \phi_1^n, \ldots, \phi_m^n) \trianglelefteq \mathbb{K}\{t_0, t_1, \ldots, t_m\}$$

for $\phi_1, \ldots, \phi_m \in \mathbb{K}$ fixed. (Observe that for the convenience of indexing, we label the difference variables starting from 0 in the present context.)

As discussed before, any solution of a C-finite recurrence is a linear combination of terms of the form $n^d\phi^n$. It is a classical result in the theory of difference equations that these terms indeed form a fundamental set of solutions to the recurrence. We need here the following linear independence statement.

Theorem 7.4 Let $d_1, \ldots, d_m \in \mathbb{N}_0$, $\phi_1, \ldots, \phi_m \in \mathbb{K} \setminus \{0\}$, and consider the sequences $f_i(n) := n^{d_i}\phi_i^n$ $(i = 1, \ldots, m)$. The set

$$\{f_1(n), f_2(n), \ldots, f_m(n)\}$$

is linearly independent over $\mathbb{K}$, provided that $(d_i, \phi_i) \neq (d_j, \phi_j)$ for $i \neq j$. ∎

For a proof, we refer to the literature (e.g. Milne-Thomson, 1933, Section 13.0). The theorem may be restated as linear independence of pairwise different exponentials over $\mathbb{K}[n]$.

Corollary 7.5 Let $\phi_1, \ldots, \phi_m \in \mathbb{K} \setminus \{0\}$ be pairwise different, and let $p_1, \ldots, p_m \in \mathbb{K}[n]$ be such that

$$p_1(n)\phi_1^n + p_2(n)\phi_2^n + \cdots + p_m(n)\phi_m^n = 0 \qquad (n \geq 1).$$

Then $p_1(n) = p_2(n) = \cdots = p_m(n) = 0$ $(n \geq 1)$.

Proof Any nontrivial relation would be in contradiction to Theorem 7.4. ■

Recall the shortcut notation $\Phi^e := \phi_1^{e_1}\cdots\phi_m^{e_m}$ for $\Phi = (\phi_1,\dots,\phi_m) \in \mathbb{K}^m$ and $e = (e_1,\dots,e_m) \in \mathbb{Z}^m$. Every *algebraic* dependency

$$a_1(n)\Phi^{e_1} + a_2(n)\Phi^{e_2} + \cdots + a_s(n)\Phi^{e_s} = 0 \qquad (n \geq 1)$$

of exponentials $\Phi := (\phi_1^n,\dots,\phi_m^n)$ can be read as a *linear* dependency

$$a_1(n)\psi_1^n + a_2(n)\psi_2^n + \cdots + a_s(n)\psi_s^n = 0 \qquad (n \geq 1)$$

of the exponentials $\psi_1^n,\dots,\psi_s^n$ with $\psi_i := (\phi_1^{e_{1,i}}\cdots\phi_m^{e_{m,i}})$. By the previous corollary, such a linear independence can only be nontrivial if some of the ψ_i^n coincide, i.e., if $\Phi^{e_i} = \Phi^{e_j}$ for some $i \neq j$. This motivates the following definition.

Definition 7.6

(1) Let $\phi_1,\dots,\phi_m \in \mathbb{K}\setminus\{0\}$. The *exponent lattice* of $\phi_1,\dots,\phi_m$ is defined as

$$L(\phi_1,\dots,\phi_m) := \{\,(e_1,\dots,e_m) \in \mathbb{Z}^m : \phi_1^{e_1}\phi_2^{e_2}\cdots\phi_m^{e_m} = 1\,\} \subseteq \mathbb{Z}^m.$$

(2) For $e = (e_1,\dots,e_m) \in \mathbb{Z}^m$, define

$$e_+ := (\max(0,e_1),\dots,\max(0,e_m)), \qquad e_- := (\max(0,-e_1),\dots,\max(0,-e_m)).$$

The *lattice ideal* of a lattice $L \subseteq \mathbb{Z}^m$ is defined as

$$I(L) := \langle T^{e_+} - T^{e_-} : e \in L \rangle \trianglelefteq \mathbb{K}[t_1,\dots,t_m],$$

where $T = (t_1,\dots,t_m)$.

Continuing in the notation of above, we have $\Phi^{e_i} = \Phi^{e_j}$ if and only if $e_i - e_j \in L(\phi_1,\dots,\phi_m)$. As a consequence, the algebraic dependencies of a set of exponentials $\phi_1^n,\dots,\phi_m^n$ is precisely described by the lattice ideal of the exponent lattice $L(\phi_1,\dots,\phi_m)$, as we will show next.

Theorem 7.7 Let $\phi_1,\dots,\phi_m \in \mathbb{K}\setminus\{0\}$. Then

$$\operatorname{ann}(n,\phi_1^n,\dots,\phi_m^n) = \langle\langle I(L(\phi_1,\dots,\phi_m))\rangle\rangle + \langle\langle st_0 - t_0 - 1, st_1 - \phi_1 t_1,\dots,st_m - \phi_m t_m\rangle\rangle.$$

Proof "$\supseteq$" Clear by definition of $L(\phi_1,\dots,\phi_m)$.
"$\subseteq$" Let $p \in \operatorname{ann}(f_1(n),\dots,f_m(n))$. We show that p reduces to zero modulo the ideal on the right hand side. By adding suitable elements of that ideal, p can be brought to a form

$$\bar{p} = a_1(t_0)T^{e_1} + \cdots + a_s(t_0)T^{e_s}$$

with $T = (t_1,\dots,t_m)$ and certain $a_i \in \mathbb{K}[t_0]$, where $e_i - e_j \notin L(\phi_1,\dots,\phi_m)$. Hence, writing $\Phi := (\phi_1,\dots,\phi_m)$, the exponentials $\psi_i := \Phi^{e_i}$ $(i = 1,\dots,m)$ are pairwise different. As $\bar{p}$ belongs to the ideal on the left, we have

$$a_1(n)\psi_1^n + a_2(n)\psi_2^n + \cdots + a_s(n)\psi_s^n = 0 \qquad (n \geq 0).$$

By Corollary 7.5, we obtain $a_1(n) = \cdots = a_s(n) = 0$, hence $\bar{p} = 0$, and hence p belongs to the ideal on the right hand side. ■

7.3 Ge's Algorithm

In the previous section, we have seen that the algebraic dependencies of sequences $\phi_1^n, \dots, \phi_s^n$ can be described by means of the integer lattice

$$L := L(\phi_1, \dots, \phi_s) := \{ (e_1, \dots, e_s) \in \mathbb{Z}^s : \phi_1^{e_1} \cdots \phi_s^{e_s} = 1 \} \subseteq \mathbb{Z}^s.$$

Most commonly, the bases $\phi_1, \dots, \phi_s$ belong to an algebraic extension field $\mathbb{Q}(\alpha)$ of $\mathbb{Q}$. For this case, Ge (1993) has given an efficient algorithm for computing a basis of L from given numbers $\phi_1, \dots, \phi_s \in \mathbb{Q}(\alpha)$.

Unfortunately, Ge's algorithm has apparently never been published in other form than his Ph.D. thesis. Articles referring to Ge's work (e.g. Derksen *et al.*, 2005; Cai *et al.*, 2000) give some hints about what the algorithm does, but details remain open. For the sake of completeness, we describe below an algorithm for computing a basis of L, based on integer relation algorithms and Diophantine approximation. The algorithm is likely to be very similar or even identical to Ge's approach, and we do not claim any sort of originality about the results presented in this section. Readers who wish to take Ge's algorithm for granted may skip this section and directly advance to Section 7.4.

1 Lattices

Definition 7.8 Let $b_1, \dots, b_m \in \mathbb{Q}^d$ be given. The set

$$[b_1, \dots, b_m] := \mathbb{Z}b_1 + \cdots + \mathbb{Z}b_m := \{ e_1 b_1 + \cdots + e_m b_m : e_1, \dots, e_m \in \mathbb{Z} \} \subseteq \mathbb{Q}^d$$

is called the *lattice* generated by $b_1, \dots, b_m$. The set $\{b_1, \dots, b_m\}$ is called a *basis* of that lattice.

By $||\cdot||$ and $||\cdot||_\infty$, we denote the Euclid norm and the maximum norm, respectively, of vectors in $\mathbb{Q}^d$, i.e.,

$$||x|| := \sqrt{x_1^2 + \cdots + x_d^2}, \quad ||x||_\infty := \max_{i=1}^{d} |x_i| \qquad (x = (x_1, \dots, x_d) \in \mathbb{Q}^d).$$

We are interested in bases of lattices whose elements are short w.r.t. the Euclid norm. Lenstra *et al.* (1982) proposed an efficient algorithm, now known as the LLL algorithm, which transforms a given basis of a lattice into such a basis. The vectors of the basis computed by LLL are not necessarily as short as can be, but they are off only by a constant factor. In particular, we have the following estimate.

Theorem 7.9 Let $L \subseteq \mathbb{Q}^d$ be a lattice and let $\{b_1, \dots, b_m\}$ be a basis computed by LLL from any given basis for L. Furthermore, let $c_1, \dots, c_\ell \in L$ be linearly independent. Then

$$||b_j|| \leq 2^{(n-1)/2} \max_{i=1}^{\ell} ||c_i|| \qquad (j = 1, \dots, \ell). \quad \blacksquare$$

Cohen (1993) has a detailed description of LLL and its applications. Also a proof of the theorem above can be found there.

2 *Continued Fractions*

Recall the definitions from page 12. We consider in the present context continued fractions of the form

$$\mathop{\mathbf{K}}_{i=0}^{n}(1/a(i)) = a(0) + \cfrac{1}{a(1) + \cfrac{1}{\cdots + \cfrac{1}{a(n)}}}$$

with $a(0) \in \mathbb{Z}$ and $a(n) \in \mathbb{N}$ for $n \geq 1$. For every n, the continued fraction is a rational number $p(n)/q(n)$, and it can be shown that $p(n)$ and $q(n)$ are precisely the continuants of this continued fraction (Perron, 1929).

Continued fractions of this form provide a unique representation of real numbers.

Theorem 7.10 Let $x \in \mathbb{R}$.

(1) If $x \in \mathbb{Q}$, then there exists $n \in \mathbb{N}$, $a(0) \in \mathbb{Z}$, and $a(1), \ldots, a(n) \in \mathbb{N}$, $a(n) \neq 1$, all uniquely determined, such that
$$x = \mathop{\mathbf{K}}_{i=0}^{n}(1/a(i)).$$

(2) If $x \notin \mathbb{Q}$, then there exists a uniquely determined sequence $a(n)$ with $a(0) \in \mathbb{Z}$ and $a(n) \in \mathbb{N}$ $(n \geq 1)$ such that
$$x = \mathop{\mathbf{K}}_{n=0}^{\infty}(1/a(i)) := \lim_{n\to\infty} \mathop{\mathbf{K}}_{i=0}^{n}(1/a(i)).$$

A proof can be found in Perron's book or any other introduction to the theory of continued fractions. The (possibly finite) sequence $a(n)$ in the theorem above is called the *continued fraction expansion* of x.

If $x \in \mathbb{R}$ and the sequence $a(n)$ is the continued fraction expansion of x, then the truncated continued fraction
$$\mathop{\mathbf{K}}_{i=0}^{N}(1/a(i)) = \frac{p(N)}{q(N)}$$
is called the Nth *approximant* (or *convergent*) to x. The name is justified by the following estimation, for a proof of which we again refer to Perron (1929).

Theorem 7.11 (Diophantine Approximation) Let $x \in \mathbb{R}$ and $a(n)$ be its continued fraction expansion. Let $p(n)/q(n)$ be the nth approximant to x. Then

$$\left| x - \frac{p(n)}{q(n)} \right| \leq \frac{1}{q(n)q(n+1)}$$

for all $n \geq 1$ for which $a(n+1)$ is defined. ■

3 Integer Relations

Given some real numbers $x_1,\dots,x_n \in \mathbb{R}\setminus\{0\}$, an *integer relation* of these numbers is an integer vector $(e_1,\dots,e_n) \in \mathbb{Z}^m$ such that

$$e_1x_1 + e_2x_2 + \cdots + e_nx_n = 0.$$

One way to find integer relations is based on continued fraction expansions and LLL (Borwein and Lisoněk, 2000).

Suppose that $\xi_1,\dots,\xi_n \in \mathbb{Q}$ are rational approximations to $x_1,\dots,x_n \in \mathbb{R}\setminus\{0\}$, with errors $\varepsilon_i := x_i - \xi_i$ $(i = 1,\dots,n)$. Choose a large number $c \in \mathbb{N}$, and consider the lattice

$$L := [\begin{pmatrix} 1\\ 0\\ \vdots\\ \vdots\\ 0\\ c\xi_1 \end{pmatrix}, \begin{pmatrix} 0\\ 1\\ 0\\ \vdots\\ 0\\ c\xi_2 \end{pmatrix}, \dots\dots, \begin{pmatrix} 0\\ \vdots\\ \vdots\\ 0\\ 1\\ c\xi_n \end{pmatrix}] \subseteq \mathbb{Q}^{n+1}. \tag{7.2}$$

Apply LLL to the basis above. The resulting basis will consist of vectors b of the form

$$b = (b_1,\dots,b_n, c(b_1\xi_1 + b_2\xi_2 + \cdots + b_n\xi_n))$$

with $b_i \in \mathbb{Z}$ $(i = 1,\dots,n)$. If $(b_1,\dots,b_n)$ is an integer relation for $x_1,\dots,x_n$, then it is likely to pop up in the basis computed by LLL, because then

$$\begin{aligned} b_1\xi_1 + b_2\xi_2 + \cdots + b_n\xi_n &= b_1(x_1-\varepsilon_1) + b_2(x_2-\varepsilon_2) + \cdots + b_n(x_n-\varepsilon_n) \\ &= \underbrace{b_1x_1 + b_2x_2 + \cdots + b_nx_n}_{=0} - (b_1\varepsilon_1 + b_2\varepsilon_2 + \cdots + b_n\varepsilon_n) \end{aligned}$$

is small. Whether or not an integer relation actually pops up in the LLL basis is governed by the the quality of the approximation and by the choice of c. Relations with large coefficients $b_1,\dots,b_n$ are only found if the vectors corresponding to the relation are actually shorter that the vectors in (7.2).

Now suppose that the lattice of integer relations for $x_1,\dots,x_n$ is generated by a basis $\{c_1,\dots,c_\ell\}$ with $||c_i||_\infty \le M$ $(i = 1,\dots,\ell)$, where $M \ge 0$ is known. Then, for all $e = (e_1,\dots,e_n) \in \{c_1,\dots,c_\ell\}$ we have

$$\begin{aligned} ||(e_1,\dots,e_n, ce_1\xi_1 + \cdots + ce_n\xi_n)|| &= ||(e_1,\dots,e_n, -c(e_1\varepsilon_1 + \cdots + e_n\varepsilon_n))|| \\ &\le ||e|| + |-c(e_1\varepsilon_1 + \cdots + e_n\varepsilon_n)| \\ &\le \sqrt{n}||e||_\infty + c|\sum_{i=1}^n e_i\varepsilon_i| \le \sqrt{n}M + cn\big(\max_{i=1}^n |e_i|\big)\big(\max_{i=1}^n |\varepsilon_i|\big) \\ &\le \big(\sqrt{n} + cn\max_{i=1}^n |\varepsilon_i|\big)M. \end{aligned}$$

By Theorem 7.9, it follows

$$||b_i|| \leq 2^{n/2} M(\sqrt{n} + \varepsilon_{\max} cn) \qquad (i = 1, \ldots, \ell)$$

where $\{b_1, \ldots, b_n\}$ is the basis obtained by applying LLL to the lattice L of (7.2) and $\varepsilon_{\max} := \max_{i=1}^{n} |\varepsilon_i|$. We want to choose the value c in such a way that the b_j are shorter than the vectors we start with. This is certainly the case whenever

$$2^{n/2} M(\sqrt{n} + \varepsilon_{\max} cn) < c \min_{i=1}^{n} |\xi_i|,$$

for the expression on the left is an upper bound for $||b_j||$ and the expression on the right is a lower bound for the length of the vectors used in (7.2) to define L. We have

$$2^{n/2} M(\sqrt{n} + \varepsilon_{\max} cn) < c \min_{i=1}^{n} |\xi_i| \quad \Longleftrightarrow \quad c > \frac{M\sqrt{n}}{2^{-n/2} \min_{i=1}^{n} |\xi_i| - nM\varepsilon_{\max}},$$

if the approximation is close enough to ensure $\min_{i=1}^{n} |\xi_i| > 2^{-n/2} nM\varepsilon_{\max}$.

Denote by $L_{M,c,\varepsilon_{\max}} \subseteq \mathbb{Z}^n$ the lattice generated by all the vectors $b = (b_1, \ldots, b_n) \in \mathbb{Z}^n$ which are obtained from vectors $(b_1, \ldots, b_n, b_{n+1}) \in \mathbb{Z}^n$ appearing in an LLL-basis of L and satisfying $|b_i| < M$ $(i = 1, \ldots, n)$ and

$$|b_{n+1}| < |b_1\varepsilon_1 + b_2\varepsilon_2 + \cdots + b_n\varepsilon_n|$$

by discarding the last coordinate b_{n+1}. Furthermore let $I_M \subseteq \mathbb{Z}^n$ be the lattice generated by all integer relations $e_1, \ldots, e_n$ for $x_1, \ldots, x_n$ with $|e_i| < M$ $(i = 1, \ldots, n)$.

Theorem 7.12 We have $I_M \subseteq L_{M,c,\varepsilon_{\max}}$ whenever $\varepsilon_{\max} < 2^{n/2} \min_{i=1}^{n} |\xi_i|/nM$ and

$$c > \frac{M\sqrt{n}}{2^{-n/2} \min_{i=1}^{n} |\xi_i| - nM\varepsilon_{\max}}.$$

Moreover, equality is obtained for sufficiently small $\varepsilon_{\max}$.

Proof The inclusion "$\subseteq$" follows from the discussion preceding the theorem. Now suppose we have "$\subsetneq$", i.e., there exists a vector $b = (b_1, \ldots, b_n) \in L_{M,c,\varepsilon_{\max}} \setminus I_M$. Without loss of generality, we may assume that b belongs to an LLL-basis of $L_{M,c,\varepsilon_{\max}}$. Then, by definition of $L_{M,c,\varepsilon_{\max}}$, we have $|b_i| < M$ $(i = 1, \ldots, n)$.

As $b_1, \ldots, b_n$ is not an integer relation for $x_1, \ldots, x_n$, we have

$$\delta := |b_1x_1 + b_2x_2 + \cdots + b_nx_n| > 0.$$

If $\varepsilon_{\max} < \delta/(1 + nM)$ then

$$\begin{aligned}
\delta &= |b_1x_1 + \cdots + b_nx_n| = |b_1(\xi_1 + \varepsilon_1) + \cdots + b_n(\xi_n + \varepsilon_n)| \\
&\leq |b_1\xi_1 + \cdots + b_n\xi_n| + |b_1\varepsilon_1 + \cdots + b_n\varepsilon_n| \\
&\leq |b_1\xi_1 + \cdots + b_n\xi_n| + nM\varepsilon_{\max} \\
\Longrightarrow \quad & |b_1\xi_1 + \cdots + b_n\xi_n| \geq \delta - nM\varepsilon_{\max} > \varepsilon_{\max}.
\end{aligned}$$

It follows that b cannot belong to an LLL-basis of $L_{M,c,\varepsilon_{\max}}$ if $\varepsilon_{\max} < \delta/(1 + nM)$.

The claim follows by observing that there are only finitely many integer vectors whose coordinates are bounded by M in absolute value. ∎

Under suitable assumptions about $x_1, \dots, x_n$, Theorem 7.12 gives rise to the following algorithm for computing a basis of I_M. The case where some of the x_i are zero has to be treated separately.

Algorithm 7.13 (Computation of Integer Relations)
Input: Real numbers $x_1, \dots, x_n \in \mathbb{R}$ whose continued fraction expansion can be effectively computed, a number $M > 0$.
Output: A basis for the smallest integer lattice containing all integer relations $e_1, \dots, e_n$ for $x_1, \dots, x_n$ with $|e_i| < M$ $(i = 1, \dots, n)$.
Assumption: For specific $e_1, \dots, e_n \in \mathbb{Z}$ it can be decided whether $e_1 x_1 + \cdots + e_n x_n = 0$.

if $x_i = 0$ for some i **then**
    Apply the algorithm to $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$, obtaining a basis B_0
    Obtain B from B_0 by inserting 0 after the $(i-1)$th entry in every $b \in B_0$
    return $B \cup \{(0, \dots, 0, 1, 0, \dots, 0)\}$ // ith unit vector
$B := \{(1, 0, \dots, 0), \dots\dots, (0, \dots, 0, 1)\}$
$\varepsilon_{\max} := 1$
while $e_1 x_1 + \cdots + e_n x_n \neq 0$ for at least one $(e_1, \dots, e_n) \in B$ **do**
    repeat
        $\varepsilon_{\max} := \varepsilon_{\max}/2$
        Let $\xi_i := p_i(n)/q_i(n)$ be convergent to ξ_i with $\varepsilon_i := 1/q_i(n)q_i(n+1) < \varepsilon_{\max}$
    while $\varepsilon_{\max} \geq 2^{n/2} \min_{i=1}^{n} |\xi_i|/n$
    Choose a number $c \in \mathbb{N}$ with $c > M\sqrt{n}/(2^{-n/2} \min_{i=1}^{n} |\xi_i| - nM\varepsilon_{\max})$
    $B := \{(1, 0, \dots, 0, c\xi_1), \dots\dots, (0, \dots, 0, 1, c\xi_n)\}$
    $B := \mathrm{LLL}(B)$
    Discard from B all $(e_1, \dots, e_n, e_{n+1})$ with $|e_{n+1}| \geq c|e_1 \varepsilon_1 + \cdots + e_n \varepsilon_n|$
    Discard the last coordinate from every vector in B
    Discard from B all vectors e with $||e||_\infty \geq M$
return B

Theorem 7.14 Algorithm 7.13 is correct and terminates.

Proof Follows from Theorems 7.11 and 7.12. ∎

4 *The Exponent Lattice*

Let $\mathbb{Q}(\alpha)$ be a simple algebraic extension of $\mathbb{Q}$, and suppose that $\phi_1, \dots, \phi_n \in \mathbb{Q}(\alpha)$ are given. Consider the exponent lattice

$$L := \{\, (e_1, \dots, e_n) \in \mathbb{Z}^n : \phi_1^{e_1} \phi_2^{e_2} \cdots \phi_n^{e_n} = 1 \,\}.$$

The following theorem due to Masser (1988) says that L has a basis consisting of "short" vectors only, with a precise meaning of "short".

Theorem 7.15 Let d be the degree of the extension $\mathbb{Q}(\alpha)/\mathbb{Q}$, and let

$$h := \max_{i=1}^{n} h(\phi_i),$$

where $h(\phi_i)$ denotes the *height* of ϕ_i, defined as the degree of its minimal polynomial plus the sum of the binary lengths of its coefficients.

Then there is a basis $B = \{b_1, \ldots, b_m\}$ of L with

$$||b_i||_\infty \leq d^2 \left[4n \max\left(hd\Big(\frac{\log(d+2)}{\log\log(d+2)}\Big)^3,\ 1\right)\right]^{n-1} \qquad (i = 1, \ldots, m) \quad \blacksquare.$$

The computation of a basis for L can be reduced to an integer relation problem which can be solved as described before. We have

$$\begin{aligned} \phi_1^{e_1} \cdots \phi_n^{e_n} = 1 \quad &\iff \quad e_1 \log\phi_1 + e_2 \log\phi_2 + \cdots + e_n \log\phi_n = 2\pi \mathrm{i} e_{n+1}. \\ &\iff \quad e_1 x_1 + \cdots + e_n x_n = 0 \\ &\qquad\quad \text{and} \quad e_1 y_1 + \cdots + e_n y_n = 2\pi e_{n+1}, \end{aligned}$$

with $x_i := \mathrm{Re}(\log\phi_i)$, $y_i := \mathrm{Im}(\log\phi_i)$ $(i = 1, \ldots, s)$ and a number e_{n+1} which is bounded by the degree of the field extension $\mathbb{Q}(\alpha)/\mathbb{Q}$. It is assumed that always the standard branch of the logarithm is taken.

Brent (1976) presents efficient algorithms for computing arbitrary precision approximation to the real numbers x_i and y_i, which can easily be turned into continued fraction expansions. Thus we can compute candidates for integer relations for $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ like in Algorithm 7.13. We might not be able to decide whether an integer relation candidate for $x_1, \ldots, x_n$ or $y_1, \ldots, y_n$ actually holds, but that is also not necessary to do. Instead, we check membership of the exponent lattice directly. The detailed algorithm is as follows.

Algorithm 7.16 (Computation of the Exponent Lattice)
Input: Algebraic numbers $\phi_1, \ldots, \phi_n \in \mathbb{Q}(\alpha)$
Output: A basis for the exponent lattice $L = \{\, (e_1, \ldots, e_n) \in \mathbb{Z}^n : \phi_1^{e_1} \cdots \phi_n^{e_n} = 1 \,\}$

$B := \{(1,0,\ldots,0), \ldots\ldots, (0,\ldots,0,1)\}$
Let $x_i := \mathrm{Re}(\log\phi_i)$, $y_i = \mathrm{Im}(\log\phi_i)$ $(i = 1, \ldots, n)$
Let M be the bound of Theorem 7.15
$\varepsilon_{\max} := 1$
while $\phi_1^{e_1} \cdots \phi_n^{e_n} \neq 1$ for at least one $(e_1, \ldots, e_n) \in B$ **do**
  Apply lines 8–17 of Algorithm 7.13 to $x_1, \ldots, x_n$, obtaining a basis X
  Apply lines 8–17 of Algorithm 7.13 to $y_1, \ldots, y_n, 2\pi$, obtaining a basis Y
  Discard the coordinates corresponding to 2π in the basis of Y
  Let B be a basis for $[X] \cap [Y]$
return B

Theorem 7.17 Algorithm 7.16 is correct and terminates.

Proof Clear by the preceding discussion. $\blacksquare$

While Ge's algorithm is proven to run in polynomial time, we have no estimate about the runtime required for the algorithm just described. However, an actual implementation in Mathematica suggests that the algorithm is very efficient. Indeed, in our implementation the runtime is usually negligible compared to the time needed for computing the primitive element α from minimal polynomials for the ϕ_i.

7.4 Dependencies of C-Finite Sequences

Using Ge's algorithm and Theorem 7.7, generators of the annihilating ideal

$$\operatorname{ann}(n, \phi_1^n, \ldots, \phi_s^n) \trianglelefteq \mathbb{K}\{t_0, t_1, \ldots, t_m\}$$

can effectively be computed. As the solutions $f_i(n)$ of a C-finite system can be written as a linear combination of exponentials ϕ_j^n, it is possible to compute a basis for the annihilator of the $f_i(n)$ from the annihilator of the exponentials. The algorithm is as follows.

Algorithm 7.18 (Annihilating ideal of C-finite sequences)
Input: A C-finite system S of order r for $f_1(n), \ldots, f_m(n)$ over $\mathbb{K} = \bar{\mathbb{Q}}$, and initial values of the $f_i(n)$.
Output: A difference ideal basis of $\operatorname{ann}(f_1(n), \ldots, f_m(n)) \trianglelefteq \mathbb{K}\{t_1, \ldots, t_m\}$

Apply Algorithm 7.2 to find $\phi_1, \ldots, \phi_s \in \mathbb{K}$ and $a_{i,j} \in \mathbb{K}[n]$ with

$$f_i(n) = a_{i,1}(n)\phi_1^n + \cdots + a_{i,s}(n)\phi_s^n \qquad (n \geq 1)$$

Compute a basis B of $L = L(\phi_1, \ldots, \phi_s)$, e.g., by Algorithm 7.16
Let $I(L) \trianglelefteq \mathbb{K}[x_1, \ldots, x_s]$ be the lattice ideal of L
Define $\mathfrak{b} \trianglelefteq \mathbb{K}[x_0, \ldots, x_s, t_1, \ldots, s^{r-1}t_1, \ldots\ldots, t_m, \ldots, s^{r-1}t_m]$ via

$$\mathfrak{b} := \langle\, s^\ell t_i - (a_{i,1}(x_0)\phi_1^\ell x_1 + \cdots + a_{i,s}(x_0)\phi_s^\ell x_s) : i = 1, \ldots, m, \ell = 0, \ldots, r-1 \,\rangle$$

$\mathfrak{c} = \langle c_1, \ldots, c_\ell \rangle := \mathfrak{b} + I(L) \cap \mathbb{K}[t_1, \ldots, s^{r-1}t_1, \ldots\ldots, t_m, \ldots, s^{r-1}t_m]$
Let $\mathfrak{a} = \langle\langle a_1, \ldots, a_m \rangle\rangle \trianglelefteq \mathbb{K}\{t_1, \ldots, t_m\}$ be the associated difference ideal of S
<u>return</u> $\{a_1, \ldots, a_m, c_1, \ldots, c_\ell\}$

Theorem 7.19 Algorithm 7.18 is correct.

Proof We have to show $\operatorname{ann}(f_1(n), \ldots, f_m(n)) = \mathfrak{a} + \langle\langle \mathfrak{c} \rangle\rangle$.
"$\supseteq$" Clearly $\mathfrak{a} \subseteq \operatorname{ann}(f_1(n), \ldots, f_m(n))$ and, using Theorem 7.7, $\mathfrak{c} \subseteq \operatorname{ann}(f_1(n), \ldots, f_m(n))$.
"$\subseteq$" Now let $p \in \operatorname{ann}(f_1(n), \ldots, f_m(n))$. By $\mathfrak{a} \subseteq \operatorname{ann}(f_1(n), \ldots, f_m(n))$, we may assume without loss of generality that the order of p is less than r, so it remains to show $p \in \mathfrak{c}$. Using the generators of $\mathfrak{b}$, p can be transformed to some $\bar{p} \in \mathbb{K}[x_0, \ldots, x_s]$ which is equivalent to p modulo $\mathfrak{c}$. Using $\mathfrak{c} \subseteq \operatorname{ann}(f_1(n), \ldots, f_m(n))$ and Theorem 7.7, it follows that $\bar{p} \in I(L)$, hence $\bar{p} \in \mathfrak{c}$, and hence $p \in \mathfrak{c}$, as claimed. ∎

Observe the difference to the algorithms of Chapter 6: The ideal computed by the algorithm above is *precisely* the annihilator, and not only an approximation to it. Using this algorithm, it is therefore also possible to obtain conclusive negative answers to, for instance, the representation problem.

Example 7.20 (Graham *et al.*, 1994, Exercise 7.26) The second order Fibonacci numbers are defined via the recurrence

$$f(n+2) = f(n+1) + f(n) + F_{n+2} \quad (n \geq 0), \qquad f(0) = 0, f(1) = 1.$$

Express $f(n)$ in terms of the ordinary Fibonacci numbers F_n and F_{n+1}.
Using Algorithm 7.18, we find

$$\operatorname{ann}(f(n), F_n) = \langle\!\langle t_2^4 + 2t_2^3 st_2 - t_2^2 st_2^2 - 2t_2 st_2^3 + st_2^4 - 1, sst_1 - st_1 - t_1 - sst_2, sst_2 - st_2 - t_2 \rangle\!\rangle.$$

It follows that there does *not* exist an algebraic function A with $f(n) = A(F_n, F_{n+1})$ $(n \geq 1)$.
Allowing also the term n to appear in the representation, we obtain

$$\begin{aligned}\operatorname{ann}(f(n), F_n, n) = \langle\!\langle & 5t_1 - (2st_3t_2 + t_3 st_2), t_2^4 + 2t_2^3 st_2 - t_2^2 st_2^2 - 2t_2 st_2^3 + st_2^4 - 1, \\ & sst_1 - st_1 - t_1 - sst_2, sst_2 - st_2 - t_2, st_3 - t_3 - 1 \rangle\!\rangle,\end{aligned}$$

the first generator of which implies the representation

$$f(n) = \tfrac{1}{5}(2(n+1)F_n + nF_{n+1}) \qquad (n \geq 0),$$

which is in accordance with the solution of Graham *et al.* (1994).

7.5 An Extension to the Multivariate Case

There are several ways to define multivariate C-finite sequences. Algorithms for proving identities involving a general sort of multivariate C-finite sequences are given by Bousquet-Mélou and Petkovšek (2000). In the present section, we will define a more restrictive class of multivariate C-finite sequences, and provide an extension of Algorithm 7.18 to this case.

The notion of an algebraic dependency of a set of multivariate sequences is defined in analogy to the univariate case. To give a precise definition, we first need to extend the notions of difference algebra to the multivariate case. A *multivariate* (d-variate) difference ring is a commutative ring R equipped with d distinguished endomorphisms $s_1, \ldots, s_d \colon R \to R$. For instance, the ring of d-variate sequences $f \colon \mathbb{N}^d \to \mathbb{K}$ equipped with $\mathbf{E}_1, \ldots, \mathbf{E}_d$, where $\mathbf{E}_i$ is defined via

$$\mathbf{E}_i f(n_1, \ldots, n_d) := f(n_1, \ldots, n_{i-1}, n_i + 1, n_{i+1}, \ldots, n_d),$$

is a d-variate difference ring. Considering $T = \{t_i^{(e_1,\ldots,e_d)} : e_1, \ldots, e_d \geq 0, i = 1, \ldots, m\}$ as a collection of indeterminates, the polynomial ring $\mathbb{K}[T]$ with infinitely many variables may be turned into a d-variate difference ring by declaring

$$s_i(t_j^{(e_1,\ldots,e_d)}) := t_j^{(e_1,\ldots,e_{i-1},e_i+1,e_{i+1},\ldots,e_d)}.$$

This difference ring is called the *free* multivariate difference ring in m (difference) variables and denoted by $\mathbb{K}\{t_1, \ldots, t_m\}$, as in the univariate case. The definitions of difference homomorphism, difference ideal, etc., extend literally to the present situation. Given some multivariate sequences

$f_1,\dots,f_m\colon \mathbb{N}^d \to \mathbb{K}$, the ideal of their algebraic dependencies is defined as the kernel of the difference homomorphism

$$\varphi\colon \mathbb{K}\{t_1,\dots,t_m\} \to \mathbb{K}^{\mathbb{N}^d}, \qquad t_i \mapsto f_i(n_1,\dots,n_d)$$

mapping elements of the ground field $\mathbb{K}$ to the corresponding constant sequences,

$$\operatorname{ann}(f_1(\mathbf{n}),\dots,f_m(\mathbf{n})) := \ker\varphi.$$

(Here and below, we use $\mathbf{n} := (n_1,\dots,n_d)$ as shortcut notation.) Note that all definitions agree with the univariate definitions for $d = 1$.

We consider sequences $f\colon \mathbb{N}^d \to \mathbb{K}$ which are defined by composition of an integer linear function with a univariate C-finite sequence, i.e., $f(n_1,\dots,n_d) = f'(e_1n_1 + e_2n_2 + \cdots + e_dn_d)$ for some fixed univariate C-finite sequence $f'(n)$ and some fixed $e_1,\dots,e_d \in \mathbb{Z}$. A typical example is the $(m+n)$th Fibonacci number F_{m+n}. C-finite sequences can be written as linear combination of exponentials, and by the exponential law $\phi^{n+m} = \phi^n\phi^m$, such a multivariate C-finite sequence can be disentangled into an equivalent expression involving univariate sequences only.

The key observation is that sequences $f(n)$ and $g(m)$, depending on "different" variables, are algebraically independent. More precisely, we have the following theorem.

Theorem 7.21 Let $f_{1,1},\dots,f_{i,j},\dots,f_{d,m}\colon \mathbb{N} \to \mathbb{K}$ be univariate sequences, and let

$$\mathfrak{a}_i^0 := \operatorname{ann}(f_{i,1}(n),\dots,f_{i,m}(n)) \trianglelefteq \mathbb{K}\{t_{i,1},\dots,t_{i,m}\} \qquad (i = 1,\dots,d)$$

be their univariate annihilating ideals and

$$\mathfrak{a}_i := \langle\langle \mathfrak{a}_i^0 \rangle\rangle + \langle\langle s_jt_{i,\ell} - t_{i,\ell} : j \neq i, \ell = 1,\dots,m \rangle\rangle \trianglelefteq \mathbb{K}\{t_{1,1},\dots,t_{d,m}\} \qquad (i = 1,\dots,d).$$

Then we have

$$\begin{aligned}&\operatorname{ann}\big(f_{1,1}(n_1),\dots,f_{1,m}(n_1),f_{2,1}(n_2),\dots,f_{2,m}(n_2),\dots\dots,f_{d,1}(n_d),\dots,f_{d,m}(n_d)\big)\\ &= \langle\langle \mathfrak{a}_1 \rangle\rangle + \langle\langle \mathfrak{a}_2 \rangle\rangle + \cdots + \langle\langle \mathfrak{a}_d \rangle\rangle \trianglelefteq \mathbb{K}\{t_{1,1},\dots,t_{d,m}\}\end{aligned}$$

for the multivariate annihilator of all the $f_{i,j}(n)$.

Proof Induction to d. For $d = 1$ there is nothing to prove. Assume the claim holds for $d-1$.

Let the difference homomorphism $\varphi\colon \mathbb{K}\{t_{1,1},\dots,t_{d-1,m}\} \to \mathbb{K}^{\mathbb{N}^d}$ be defined as usual. We have to show that $\ker\varphi = \langle\langle \mathfrak{a}_1 \rangle\rangle + \cdots + \langle\langle \mathfrak{a}_d \rangle\rangle$.

"$\supseteq$" clear by the definition of the $\mathfrak{a}_i$.

"$\subseteq$" Let $p \in \ker\varphi$ be fully reduced w.r.t. the ideal on the right. We have to show $p = 0$. For $n_1,\dots,n_{d-1} \in \mathbb{N}$, let

$$p' := p'(n_1,\dots,n_{d-1}) \in \mathbb{K}\{t_{d,1},\dots,t_{d,m}\}$$

be the polynomial obtained from p by substituting $f_{i,j}(n_i+\ell) \in \mathbb{K}$ for $s_i^\ell t_j$ in p. Then each p' is also reduced w.r.t. the ideal on the right, and by the independence of n_d and $n_1,\dots,n_{d-1}$ we have

$$p' \in \ker\varphi \cap \mathbb{K}\{t_{d,1},\dots,t_{d,m}\} = \ker\varphi|_{\mathbb{K}\{t_{d,1},\dots,t_{d,m}\}} = \mathfrak{a}_d,$$

hence $p' = 0$. As $n_1,\dots,n_{d-1}$ were arbitrary, it follows

$$p \in \ker\varphi|_{\mathbb{K}\{t_{1,1},\dots,t_{d-1,m}\}} = \langle\langle \mathfrak{a}_1 \rangle\rangle + \cdots + \langle\langle \mathfrak{a}_{d-1} \rangle\rangle,$$

hence $p = 0$. ∎

Using the theorem above, we can compute the ideal of algebraic dependencies of multivariate C-finite sequences similar as in Algorithm 7.18 from the annihilating ideal of the involved exponentials by elimination. The algorithm is as follows.

Algorithm 7.22 (Annihilating ideal of multivariate C-finite sequences)
Input: A C-finite system S of order r for $f_1(n),\dots,f_m(n)$ over $\mathbb{K}=\bar{\mathbb{Q}}$, and initial values of the $f_i(n)$, integer vectors $e_i=(e_{i,1},\dots,e_{i,d})\in\mathbb{Z}^d$ $(i=1,\dots,m)$.
Output: A difference ideal basis of the multivariate difference ideal $\operatorname{ann}(g_1(\mathbf{n}),\dots,g_m(\mathbf{n}))\trianglelefteq\mathbb{K}\{t_1,\dots,t_m\}$, where $g_i(\mathbf{n}):=f_i(e_{i,1}n_1+\dots+e_{i,d}n_d)$ $(i=1,\dots,m)$

1 Apply Algorithm 7.2 to find $\phi_1,\dots,\phi_s\in\mathbb{K}\setminus\{0\}$ and $a_{i,j}\in\mathbb{K}[n]$ with

$$f_i(n)=a_{i,1}(n)\phi_1^n+\dots+a_{i,s}(n)\phi_s^n \qquad (n\geq 1)$$

2 Compute a basis for $\mathfrak{a}:=\operatorname{ann}(\phi_1^{n_1},\dots,\phi_s^{n_1},\dots\dots,\phi_1^{n_d},\dots,\phi_s^{n_d})\trianglelefteq\mathbb{K}\{x_{1,1},\dots,x_{d,s}\}$

3 Define $\mathfrak{b}\trianglelefteq\mathbb{K}\{x_{1,0},\dots,x_{d,s},t_1,\dots,t_m\}$ via

$$\mathfrak{b}:=\langle\langle t_i-\sum_{j=1}^{s}a_{i,j}(e_{i,1}x_{1,0}+e_{i,2}x_{2,0}+\dots+e_{i,d}x_{d,0})x_{1,j}^{e_{i,1}}x_{2,j}^{e_{i,2}}\cdots x_{d,j}^{e_{i,d}}:i=1,\dots,m\rangle\rangle$$

4 **return** a basis of $(\mathfrak{a}+\mathfrak{b})\cap\mathbb{K}\{t_1,\dots,t_m\}$

We leave it to the reader to verify correctness. The argument is the same as for Algorithm 7.18. However, for the sake of readability, the listing above is not as explicit as the listing of Algorithm 7.18, so we should argue that all steps are actually computable. There is nothing special about line 1. As for line 2, a basis for the ideal can be computed from the respective univariate annihilators by means of Theorem 7.21 above. The univariate annihilators can be obtained using Ge's algorithm, as discussed earlier. In line 3, a technical difficulty arises in the case of a negative exponent $e_{i,\ell}$. This is easily circumvented by modifying step 1 such as to replace $\phi_1,\dots,\phi_s$ by $\phi_1,\dots,\phi_s,\phi_1^{-1},\dots,\phi_s^{-1}$, so that in line 3, there are difference variables for both $\phi_j^{e_{i,\ell}n_\ell}$ and $\phi_j^{-e_{i,\ell}n_\ell}$ available. Finally, we have to compute the elimination ideal in line 4. This can be done like in Chapter 6 by some distinction into algebraic part and recurrences: determine bounds for the order of each $f_i(e_{i,1}n_1+\dots+e_{i,d}n_d)$ with respect to each direction, and cut of the higher order indeterminates of $\mathbb{K}\{t_1,\dots,t_m\}$ accordingly. The result is a ring with finitely many indeterminates, and the elimination ideal can be computed via Gröbner bases.

7.6 Examples and Applications

1 Proving Identities

Proving identities involving C-finite sequences, and finding closed forms in terms of exponentials has been done for some while in symbolic computation (Nemes and Petkovšek, 1995). Also the algorithms of Chapter 4 can of course be employed for proving particular identities at hand. If the ideal of algebraic dependencies is explicitly known, then proving becomes even easier than that: an identity holds if and only if it belongs to the ideal of algebraic dependencies, and ideal membership can be easily decided. A lot of puzzles appearing in contemporary mathematical journals have therefore become routine.

Example 7.23

(1) $10F_{10n-5} + 12F_{10n-10} + F_{10n-15} = 25F_{2n}^5 + 25F_{2n}^3 + 5F_{2n}$ (Brown, 2003b)
(2) $F_n^2 + F_{n+1}^2 + 4F_{n+2}^2 = F_{n+3}^2 + L_{n+1}^2$ (Bruckman, 2002a)
(3) $L_n^2 + L_{n+1}^2 + 4L_{n+2}^2 = L_{n+3}^2 + (5F_{n+1})^2$ (Bruckman, 2002a)
(4) $(F_{n+4} + L_{n+3})^5 + (F_n + L_{n+1})^5 + (2F_{n+1} + L_{n+2})^5$
$= (2F_{n+3} + L_{n+2})^5 + F_{n+2}^5 + 5F_{n+2}^5 + 1920F_nF_{n+1}F_{n+2}F_{n+3}F_{n+4}$ (Bruckman, 2002b)

Textbooks on Fibonacci numbers (Hoggatt, 1979, e.g.) contain vast collections of identities like this, including also multivariate examples like Catalan's identity

$$F_n^2 - F_{n+m}F_{n-m} = (-1)^{n-m}F_m^2.$$

All these identities are easy to verify automatically, by Algorithm 4.2, by computing the annihilating ideal and checking ideal membership, or by any of the classic methods.

2 *Finding and Manipulating Identities*

More interesting might be that identities can also be *found* in an automated way, provided that it is specified where to search. For instance, Algorithm 7.18 delivers

$$\operatorname{ann}(F_n, (-1)^n) = \langle\langle sst_1 - st_1 - t_1, st_2 + t_2, t_2^2 - 1, st_1^2 - t_1st_1 - t_1^2 - t_2\rangle\rangle,$$

the last generator of which can be recognized as Cassini's identity.

In order to find, for instance, an identity that relates F_n, F_{2n}, and F_{3n} to each other, it is sufficient to inspect the ideal basis of $\operatorname{ann}(F_n, F_{2n}, F_{3n})$ computed by Algorithm 7.18. It contains a difference polynomial corresponding to the nice identity $4F_nF_{3n} = 5F_n^4 + 3F_{2n}^2$.

Example 7.24 Consider the sequence of twin primes 3, 5, 7, 11, 17, 19, 29, ... Problem 10844 in the American Math. Monthly asks which numbers are both a twin prime and a Fibonacci number.

In his solution, Watt (2002) proves that these are precisely the number 3, 5, and 13. The main part of the proof consists of showing that $F_n \pm 2$ is composite for all odd $n \geq 3$.

Our algorithm can assist in the search for a factorization. We only have to specify which terms we expect to appear in the factorization. For example, we have

$$\operatorname{ann}(F_{2n-1} - 2, (-1)^n, F_n) = \langle\langle t_1 + 2 + 2t_2 + 4t_3st_3 - 3st_3^2, t_1 + 2 - 2t_2 - 4t_3^2 + st_4^2, \dots\rangle\rangle.$$

It follows that

$$F_{2n-1} - 2 = \begin{cases} F_{n+1}(4F_{n+1} - 3F_n) & \text{if } n \text{ is odd} \\ (4F_n - F_{n+1})(4F_n + F_{n+1}) & \text{if } n \text{ is even} \end{cases}.$$

Similarly, the factorization

$$F_{2n-1} + 2 = \begin{cases} (2F_n + F_{n+1})(2F_n - F_{n+1}) & \text{if } n \text{ is odd} \\ F_{n+1}(3F_{n+1} - 4F_n) & \text{if } n \text{ is even} \end{cases}$$

can be found for $F_{2n-1} + 2$. (It does of course not matter that our factorizations differ from Watt's.)

The ideal of algebraic dependencies can also be used to design problems.

Example 7.25 The polynomial

$$p_n := (x - F_n)(x + F_n)(x - L_n)(x + L_n)(x - F_{2n})$$

obviously has the roots $\pm F_n, \pm L_n$, and F_{2n}. The elimination ideal

$$(\operatorname{ann}(F_n, L_n, F_{2n}) + \langle (x - t_1)(x + t_1)(x - t_2)(x + t_2)(x - t_3) \rangle) \cap \mathbb{Q}\{t_1, t_3\}[x]$$

contains the polynomial $x^5 + t_3x^4 + 2(t_3 - 2st_1^2)x^3 + 2t_3(t_3 - 2st_1^2)x^2 + t_3^2x + t_3^3$, and therefore

$$p_n = x^5 + F_{2n}x^4 + 2(F_{2n} - 2F_{n+1}^2)x^3 + 2F_{2n}(F_{2n} - 2F_{n+1}^2)x^2 + F_{2n}^2x + F_{2n}^3$$

for all n. In this representation the roots of p_n are no longer obvious. Diaz and Egozcue (2002a) have given p_n in this way and asked for its roots.

We are by no means restricted to the Fibonacci numbers. Some other combinatorial sequences also obey C-finite recurrences, and Algorithm 7.18 may be used to study their algebraic dependencies.

Example 7.26

(1) The sequence $f(n)$ defined via

$$f(n) = f(n-1) + 2f(n-2) - f(n-3) \quad (n \geq 3), \qquad f(0) = 1, f(1) = 3, f(2) = 7$$

describes the number of strings of length n over the alphabet $\{a, b, c\}$ containing neither aa nor bc as a substring (Stanley, 1997, Example 4.7.4). With our algorithm, we find that $f(n), f(n+1), f(n+2)$ are algebraically dependent with the alternating sign $(-1)^n$ via

$$\begin{aligned} 29(-1)^n = {} & f(n)^3 - 4f(n)^2f(n+1) + 3f(n)f(n+1)^2 + f(n+1)^3 \\ & - f(n)^2f(n+2) + 5f(n)f(n+1)f(n+2) - f(n+1)^2f(n+2) \\ & - 2f(n)f(n+2)^2 - 2f(n+1)f(n+2)^2 + f(n+2)^3 \quad (n \geq 0). \end{aligned}$$

(2) The sequence $f(n)$ defined via

$$f(n) = 5f(n-1) - 7f(n-2) + 4f(n-3) \quad (n \geq 3), \qquad f(0) = \tfrac{5}{16}, f(1) = \tfrac{3}{4}, f(2) = 2$$

describes the number of HC-polyominoes for $n \geq 2$ (Stanley, 1997, Example 4.7.18). With our algorithm, we find that $f(n), f(n+1), f(n+2)$ are algebraically dependent with 2^n via

$$\begin{aligned} 2^{2n} = {} & 256f(n)^3 - 896f(n)^2f(n+1) + 1104f(n)f(n+1)^2 - 496f(n+1)^3 \\ & + 320f(n)^2f(n+2) - 752f(n)f(n+1)f(n+2) + 512f(n+1)^2f(n+2) \\ & + 112f(n)f(n+2)^2 - 160f(n+1)f(n+2)^2 + 16f(n+2)^3 \quad (n \geq 0). \end{aligned}$$

Both relations were probably unknown before, and seem very difficult to justify by means of a combinatorial argument. Also the facts that the sequences $f(n)$ are algebraically independent with each other, with the Pell numbers, and, e.g., with the exponential 3^n are easily proved by Algorithm 7.18.

It is well known that the value $f(n)$ of a sequence satisfying a C-finite recurrence can be computed with a number of field operations which is only logarithmic in n. (As opposed to Algorithm 3.9, which requires a linear number of operations.) For example, the Fibonacci numbers satisfy the identities

$$F_{2n} = (2F_{n-1}+F_n)F_n \quad \text{and} \quad F_{2n+1} = F_{n-1}^2 + F_n^2 \qquad (n \geq 1),$$

from which it is easy to derive an algorithm that computes F_n in a repeated squaring fashion. Dijkstra (1978), while making fun of automated software optimization, asked how one could possibly transform automatically the naive evaluation based on the recurrence $F_{n+2} = F_{n+1} + F_n$ into the fast algorithm. Our algorithm can supply the underlying identities.

Example 7.27 For the *Tribonacci numbers* T_n, defined via

$$T_{n+3} = T_{n+2} + T_{n+1} + T_n \quad (n \geq 0), \qquad T_0 = 0, T_1 = T_2 = 1,$$

the identities

$$\begin{aligned} T_{2n} &= 2T_nT_{n+2} + T_{n+1}^2 - T_n^2 - 2T_{n+1}T_{n+2} - T_{n+1}^2 \\ T_{2n+1} &= T_n^2 - T_{n+1}^2 + 2T_{n+1}T_{n+2} \end{aligned}$$

are quickly delivered by Algorithm 7.18. More generally, we obtain the addition theorem

$$T_{n+m} = 2T_{m+1}T_n - T_{m+2}T_n + 2T_mT_{n+1} + T_{m+1}T_{n+1} - T_{m+2}T_{n+1} - T_mT_{n+2} - T_{m+1}T_{n+2} + T_{m+2}T_{n+2}$$

by Algorithm 7.22.

3 *Divisibility Properties and Modular Identities*

Another application of algebraic dependencies concerns congruence relations. For example, Graham *et al.* (1994) give the modular identity

$$F_{3n+1} \equiv F_{n+1}^3 \bmod F_n^2 \qquad (n \geq 0),$$

which we can treat as well. For, compute $\mathfrak{a} := \operatorname{ann}(F_{3n+1}, F_n) \cap \mathbb{K}[t_1, t_2, st_2]$ and observe that

$$\mathfrak{a} + \langle t_2 \rangle = \langle t_1 - st_2^2, 2t_2 + st_2 - st_2^5, 1 - 2st_2^4 + st_2^8 \rangle.$$

The first of the generators establishes the congruence above, and we even get the two additional identities

$$F_{n+1}^5 \equiv 2F_n + F_{n+1} \bmod F_n^2 \quad \text{and} \quad F_{n+1}^8 \equiv 2F_{n+1}^4 - 1 \bmod F_n^2$$

for free.

As an other example, the divisibility property

$$Q_n \mid Q_{n+2m}^4 - (Q_{2m}^2 - 4)^2 \qquad (n, m \geq 0)$$

for the Pell-Lucas numbers follows directly from the algebraic relation

$$Q_{2m+n}^4-(Q_{2m}^2-4)^2+Q_n(Q_n^3-2Q_n^2Q_{2m}Q_{2m+n}+2Q_nQ_{2m+n}^2+Q_nQ_{2m}^2Q_{2m+n}^2-2Q_{2m}Q_{2m+n}^3)=0$$

which was discovered using Algorithm 7.22. Curiously enough, the same divisibility property holds for the Lucas numbers L_n (though they satisfy a different recurrence than Q_n), but does not hold for the Pell numbers P_n (though they satisfy the same recurrence as Q_n).

We conclude this chapter by two more problems which have be posed in the literature and can be attacked by our algorithms.

Example 7.28

(1) (Furdui, 2002) *Prove that* $\gcd(L_n, F_{n+1}) = 1$ *for all* $n \geq 1$.

We have

$$\operatorname{ann}(L_n, F_n) \cap k[t_1, st_2] = \langle \underbrace{t_1^4 - 10t_1^3 st_2 + 35t_1^2 st_2 - 50t_1 st_2^3 + 25st_2^4 - 1}_{=:g} \rangle.$$

Using Gröbner basis (cf. Chapter 2), we can compute the Bézout factors (cofactors) of t_1, st_2, and g, obtaining the polynomial identity

$$1 = t_1^3 \cdot t_1 + (-10t_1^3 + 35t_1^2 st_2 - 50t_1 st_2^2 + 25st_2^3)st_2 + (-1)g.$$

Hence there are integer sequences $p(n), q(n)$ such that

$$1 = p(n)L_n + q(n)F_{n+1} + 0 \qquad (n \geq 1),$$

and the claim follows.

(2) (Friendman, 1995) *Let* $a(n)$ *be defined via*

$$a(n+2) = 5a(n+1) - a(n) \quad (n \geq 0), \qquad a(0) = a(1) = 1.$$

Prove that $a(n)a(n+1) \mid a(n+1)^2 + a(n)^2 + 3$ *for all* $n \in \mathbb{N}$.

The claim is obvious by $\operatorname{ann}(a(n)) = \langle\langle st_1^2 + t_1^2 + 3 - 5t_1 st_1, sst - 5st + t \rangle\rangle$.

We have to remark that not all divisibility properties are consequences of algebraic dependencies, and therefore it is possible that proving attempts like above fail. For instance, the congruence

$$P_{3n-1} \equiv F_{n+2} \bmod 13$$

(Seiffert, 1994a) is apparently not motivated by an algebraic dependency between the sequences P_{3n} and F_n.

8 Summation of Admissible Sequences

The summation problem is an important special case of the representation problem. Given a sequence $f(n)$, we seek an expression for $\sum_{k=1}^{n} f(k)$ without the outermost summation quantifier. We consider in this chapter the summation problem in the case where the summand sequence $f(n)$ is admissible. By Theorem 3.5.(2), then also $\sum_{k=1}^{n} f(k)$ is admissible. We want to know whether the sum can be written as a polynomial in the sequences $f_1(n), \dots, f_m(n)$ appearing in the admissible system which defines the summand sequence $f(n)$. Roughly speaking, we want to express the sum in terms of its subexpressions.

Of course, if we know a basis of $\operatorname{ann}(f_1(n), \dots, f_m(n), \sum_{k=1}^{n} f(k))$, then the summation problem can be answered easily. Just eliminate $st_{m+1}, s^2 t_{m+1}, \dots$ from the algebraic part (cf. Chapter 6). A closed form exists if and only if the elimination ideal contains a difference polynomial of the form $p + t_{m+1} q$ for some $p, q \in \mathbb{K}\{t_1, \dots, t_m\}$, $q \neq 0$.

The only problem is that in general we do not know a basis of the annihilating ideal. We only know how to compute bases of ideals that are, in a sense, close to the annihilating ideal. If an approximation is fine enough, it will contain the desired closed form(s) for the sum, and by computing finer and finer approximations, one can obtain a semidecision procedure for the summation problem (more generally, for the representation problem) that terminates if and only if a closed form exists (Section 6.5).

In this chapter, we introduce an alternative method for computing closed forms of indefinite sums over admissible sequences, which takes advantage of the special form of the summation problem in comparison to the general representation problem. We will also consider the problem of definite summation.

8.1 Solving Linear Difference Equations

Before we turn to the summation problem, let us consider the more general problem of solving linear difference equations in a difference ring.

Throughout this section, let $\mathfrak{a} \trianglelefteq \mathbb{K}\{t_1, \dots, t_m\}$ be a difference ideal which contains for each i a difference polynomial p_i of the form $p_i = s^j t_i - q$ or $p_i = q s^j t_i - 1$ for some q being of lower order than j. Note that the annihilating ideal of a tuple of admissible sequences as well as the approximations to it discussed in Chapter 6 are difference ideals of this form. If r denotes the maximum order j with the above property, then we may write $\mathfrak{a} = \langle\langle \mathfrak{a}_0 \rangle\rangle + \langle\langle p_1, \dots, p_m \rangle\rangle$ for some polynomial ideal $\mathfrak{a}_0 \trianglelefteq \mathbb{K}\{t_1, \dots, t_m\}_r$. We assume that $\mathfrak{a}$ is given by a basis of $\mathfrak{a}_0$ and $p_1, \dots, p_m$.

We consider linear difference equations in the difference ring $R := \mathbb{K}\{t_1, \dots, t_m\}/\mathfrak{a}$. That is, given $f, a_0, \dots, a_u \in R$, we want to find elements $x \in R$ that fulfill the equation

$$a_0 x + a_1 s(x) + \cdots + a_u s^u(x) = f. \tag{8.1}$$

In the homogeneous case, i.e., if $f = 0$, the set of solutions forms a vector space over the ground field $\mathbb{K}$. This solution space need not be finite dimensional, though. Every solution of the general equation ($f \neq 0$ allowed) can be written as the sum of some fixed particular solution and an element of the vector space of solutions to the homogeneous equation. By modifying equation (8.1) slightly, also this solution set can be turned into a vector space over $\mathbb{K}$. Introducing a new parameter ξ, the solutions $(x, \xi) \in R \times \mathbb{K}$ of the equation

$$a_0 x + a_1 s(x) + \cdots + a_u s^u(x) = \xi f \tag{8.2}$$

form a vector space over $\mathbb{K}$. Solutions with $\xi = 1$ are precisely the solutions of (8.1).
We have

$$(x, \xi) \text{ is a solution of (8.2)} \quad \Longleftrightarrow \quad (x, s(x), \ldots, s^u(x), \xi) \in \operatorname{Syz}(a_0, a_1, \ldots, a_u, -f),$$

by the definition of a syzygy. This gives rise to the following algorithm for computing solutions of (8.2)—and hence, solutions of (8.1). For convenience of notation, we identify an equivalence class $p \in \mathbb{K}\{t_1, \ldots, t_m\}/\mathfrak{a}$ with the unique normal form $\bar{p}$ of any element of this class modulo the Gröbner basis of $\langle\langle \mathfrak{a}_0 \rangle\rangle$ (with respect to some arbitrary but fixed term order).

Algorithm 8.1 (Solving linear difference equations)
Input: $\mathfrak{a}_0 \trianglelefteq \mathbb{K}\{t_1, \ldots, t_m\}_r$ and $p_1, \ldots, p_m$ like above; $f, a_0, \ldots, a_u \in R := \mathbb{K}\{t_1, \ldots, t_m\}/\mathfrak{a}$, where $\mathfrak{a} = \langle\langle \mathfrak{a}_0 \rangle\rangle + \langle\langle p_1, \ldots, p_m \rangle\rangle$; a number $D \in \mathbb{N}_0$
Output: Linearly independent pairs $(x_1, \xi_1), (x_2, \xi_2), (x_3, \xi_3), \ldots \in R \times \mathbb{K}$ with $a_0 x_n + a_1 s(x_n) + \cdots + a_u s^u(x_n) = \xi_n f$ $(n \in \mathbb{N})$.

1 $S = \text{Gröbnerbasis}(\operatorname{Syz}(a_0, \ldots, a_u, -f))$ // cf. Theorem 2.9
2 $B = \emptyset$
3 **for** $d = 0$ **to** D **do**
4 Let T be the set of terms τ in $\mathbb{K}\{t_1, \ldots, t_m\}_r$ with $\deg \tau \leq d$
5 $T := T \setminus \operatorname{LT}\langle \mathfrak{a}_0 \rangle$
6 $T = \{\tau_1, \ldots, \tau_\ell\} := T \setminus \operatorname{LT} B$
7 Let $x := \alpha_1 \tau_1 + \cdots + \alpha_\ell \tau_\ell$ where the α_i are indeterminates
8 Let $(q_0, \ldots, q_{u+1})$ be the normal form of $(x, sx, \ldots, s^u x, \xi)$ w.r.t. S (ξ an indeterminate)
9 Solve the linear system for the α_i and ξ
obtained by comparing the coefficients of the q_i to 0
10 Obtain $(x_1, \xi_1), \ldots, (x_s, \xi_s)$ from (x, ξ)
by spezializing the α_i and ξ to the solutions of this system
11 $B := B \cup \{(x_1, \xi_1), \ldots (x_s, \xi_s)\}$
12 **return** B

By $\operatorname{LT} B$ in line 6, we mean the set of all leading terms of the x_i, where $B = \{\, (x_i, \xi_i) : i \in I \,\}$.

Theorem 8.2 The output $B = \{\, (x_i, \xi_i) : i \in I \,\}$ of Algorithm 8.1 generates the $\mathbb{K}$-vector space of all solutions (x_i, ξ_i) of (8.2) with $\deg x_i \leq D$.

Proof "$\Rightarrow$" It suffices to show that every (x_i, ξ_i) is a solution. Indeed, every (x_i, ξ_i) which is returned by the algorithm is by construction such that $(x_i, sx_i, \ldots, s^u x_i, \xi_i)$ reduces to zero modulo S. Hence, $(x_i, sx_i, \ldots, s^u x_i, \xi_i) \in \operatorname{Syz}(a_0, \ldots, a_u, -f)$, and hence

$$a_0 x_i + a_1 s x_i + \cdots + a_u s^u x_i = \xi_i f,$$

as required.

"$\Leftarrow$" We have to argue that no solutions are lost in lines 5 and 6. For line 5, this is evident. For line 6, assume that in the dth iteration, the algorithm overlooks a solution p of degree d involving a monomial $\alpha\tau$ with $\tau \in \mathrm{LT}\, B$ and $\alpha \in \mathbb{K}$. Note that $\deg \tau < d$. Then some solution x with $\mathrm{LT}(x) = \tau$ was discovered earlier, and $\mathrm{LC}(x)p - \alpha x$ is another solution of degree d, which is free of τ. Repeating the argument if necessary leads to a solution x' of degree d which is not overlooked. The original solution x is a $\mathbb{K}$-linear combination of x' and solutions discovered earlier, so it need not be output. ■

Example 8.3 Consider the P-finite difference equation

$$\begin{aligned}(2n^3+13n^2+25n+14)f(n)+(5n^3+38n^2+93n+74)f(n+1)\\ +(6n^2+38n+58)f(n+2)-(5n^3+43n^2+120n+108)f(n+3)\\ -(6n^2+42n+72)f(n+4)+(n^3+12n^2+47n+60)f(n+5)=0.\end{aligned}$$

We are interested in solutions of this equation which can be expressed in terms of the Fibonacci numbers F_n and the Harmonic numbers H_n.

A suitable difference ring is $\mathbb{Q}\{t_1,t_2,t_3,t_4\}/\mathfrak{a}$ with

$$\mathfrak{a} = \langle\langle st_1 - t_1 - 1, t_2t_1 - 1, st_3 - t_3 - st_2, sst_4 - st_4 - t_4\rangle\rangle.$$

Let $a_0,\dots,a_5$ denote the difference polynomials corresponding to the respective coefficients in the difference equation (replace n by t_1). Then

$$S = \mathrm{Syz}(a_0,\dots,a_5) = [\begin{pmatrix}5\\-3\\2\\-1\\1\\0\end{pmatrix}, \begin{pmatrix}-3\\2\\-1\\1\\0\\1\end{pmatrix}, \begin{pmatrix}-57-20t_1\\21+8t_1\\0\\7\\0\\0\end{pmatrix}, \begin{pmatrix}-17\\13\\8+4t_1\\11\\0\\0\end{pmatrix}, \begin{pmatrix}-5\\17+4t_1\\0\\11+4t_1\\0\\0\end{pmatrix}]$$

(We can discard the inhomogeneous part 0 and leave out the components corresponding to the value ξ of equation (8.2) in the current description.)

Next, we make an ansatz for the solution, $x = \alpha_0 + \alpha_1 t_3 + \alpha_2 t_4 + \alpha_3 st_4$. A normal form computation gives

$$\begin{pmatrix}x\\sx\\s^2x\\s^3x\\s^4x\\s^5x\end{pmatrix} = \begin{pmatrix}\alpha_1+\alpha_2t_3+\alpha_3t_4+\alpha_4st_4\\ \alpha_1+\alpha_2(t_3+st_2)+\alpha_3st_4+\alpha_4(t_4+st_4)\\ \alpha_1+\alpha_2(t_3+st_2+s^2t_2)+\alpha_3(t_4+st_4)+\alpha_4(t_4+2st_4)\\ \alpha_1+\alpha_2(t_3+st_2+s^2t_2+s^3t_2)+\alpha_3(t_4+2st_4)+\alpha_4(2t_4+3st_4)\\ \alpha_1+\alpha_2(t_3+st_2+s^2t_2+s^3t_2+s^4t_2)+\alpha_3(2t_4+3st_4)+\alpha_4(3t_4+5st_4)\\ \alpha_1+\alpha_2(t_3+st_2+s^2t_2+s^3t_2+s^4t_2+s^5t_2)+\alpha_3(3t_4+5st_4)+\alpha_4(5t_4+8st_4)\end{pmatrix}$$

$$\to_s \begin{pmatrix} 0 \\ 0 \\ (-2\alpha_1 - \alpha_2) - 2\alpha_2 t_3 \\ (-24\alpha_1 + \alpha_2) - 24\alpha_2 t_3 \\ (-48\alpha_1 + 7\alpha_2) - 48\alpha_2 t_3 \\ (-84\alpha_1 + 5\alpha_2) - 84\alpha_2 t_3 + \alpha_2 s t_2 \end{pmatrix}$$

(recall that all polynomials are understood modulo $\mathfrak{a}$). Comparing coefficients to zero leads to a linear system for $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ whose solution space is generated by $(0,0,1,0)$ and $(0,0,0,1)$. Hence, $x_1 := t_4$ and $x_2 := st_4$ are our first solutions.

We increase the degree and apply the same procedure. For the ansatz polynomial

$$x = \alpha_0 + \alpha_1 t_3 + \alpha_2 t_3^2 + \alpha_3 t_3 t_4 + \alpha_4 t_3 s t_4 + \alpha_5 t_4^2 + \alpha_6 t_4 s t_4 + \alpha_7 s t_4^2,$$

we obtain a linear system of size 31×8. The solutions of this system correspond to the solutions $t_3 t_4$ and $t_3 s t_4$ of the difference equation.

It follows that the difference equation above has the solution

$$f(n) = c_1 F_n + c_2 F_{n+1} + c_3 F_n \mathrm{H}_n + c_4 F_{n+1} \mathrm{H}_n$$

for $c_1, \dots, c_4 \in \mathbb{K}$.

We have set $\mathfrak{a}$ to the difference ideal generated by the recurrences in the example above, ignoring the nontrivial algebraic relations obeyed by the Fibonacci numbers. Nevertheless, we received a solution. In general, the solution set of a linear difference equation in a difference ring $\mathbb{K}\{t_1, \dots, t_m\}/\mathfrak{a}$ can have quite a different structure than a solution set in $\mathbb{K}^{\mathbb{N}}$.

Example 8.4

(1) According to Example 4.7, $f(n) = -F_n/F_{n+1}$ is a solution of the inhomogeneous difference equation

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

Consider the difference equation

$$s(x) - x = -t_1 s t_2 s^2 t_3$$

in the difference ring $R = \mathbb{K}\{t_1, t_2, t_3\}/\langle\langle st_1 + t_1, sst_2 - st_2 - t_2, t_3 t_2 - 1 \rangle\rangle$. By Algorithm 8.1, we find that there is no solution to this equation of total degree up to five (and probably neither of higher degree).

In order to receive the solution that we expect, we have to take into account the algebraic dependencies. Using Algorithm 7.18, we find

$$\operatorname{ann}((-1)^n, F_n) = \langle\langle t_1^2 - 1, t_1 + t_2^2 + t_2 s t_2 - s t_2^2, st_1 + t_1, sst_2 - st_2 - t_2 \rangle\rangle.$$

Indeed, in $\mathbb{K}\{t_1, t_2, t_3\}/\langle\langle t_1^2 - 1, t_1 + t_2^2 + t_2 s t_2 - s t_2^2, st_1 + t_1, sst_2 - st_2 - t_2, t_3 t_2 - 1 \rangle\rangle$, the difference equation has the solution $-t_2 s t_3$, and this solution is found by Algorithm 8.1.

(2) The solution space of the C-finite recurrence

$$f(n+2) = f(n+1) + f(n)$$

over $\mathbb{Q}$ is, of course, two-dimensional. A basis of the solution space is $\{F_n, F_{n+1}\}$. Another basis is $\{L_n, L_{n+1}\}$.

Consider the difference equation

$$x + s(x) - s^2(x) = 0 \tag{8.3}$$

in the difference ring $R = \mathbb{K}\{t_1,t_2\}/\langle\langle s^2t_1 - st_1 - t_1, s^2t_2 - st_2 - t_2\rangle\rangle$, where we interpret t_1 as a representatation of F_n and t_2 as a representation of L_n.

With Algorithm 8.1, we find the solutions t_1, st_1, t_2, st_2 of degree 1. Unlike the sequences $F_n, F_{n+1}, L_n, L_{n+1}$ that we have in mind, there is no linear dependence between their difference algebraic representations, i.e., t_1, st_1, t_2, st_2 are linearli independent over $\mathbb{K}$ as elements of R, even though the corresponding sequences are not.

Again, the situation changes if we use the complete annihilating ideal in the definition of R. Again, Algorithm 7.18 can be applied. Solving equation (8.3) in the ring $R = \mathbb{K}\{t_1,t_2\}/\mathfrak{a}$ with

$$\begin{aligned}\mathfrak{a} := \operatorname{ann}(F_n, L_n) = \langle\langle & 25 - t_2^4 - 2t_2^3 st_2 + t_2^2 st_2^2 + 2t_2 st_2^3 - st_2^4, 5t_1 + t_2 - 2st_2, 5st_1 - 2t_2 - st_2,\\ & s^2t_1 - st_1 - t_1, s^2t_2 - st_2 - t_2\rangle\rangle\end{aligned}$$

gives only the two solutions t_1, st_1 (or a different basis with two elements, depending on the choice of the term order).

(3) Even if we solve (8.3) in the ring $\mathbb{K}\{t\}/\langle\langle sst - st - t\rangle\rangle$, we get an unusual solution set. Besides t and st, there is an infinite series of additional solutions, some of which are

$$t(st^2 - st\,t - t^2)^2, st(st^2 - st\,t - t^2)^2, t(st^2 - st\,t - t^2)^4, st(st^2 - st\,t - t^2)^4, \dots$$

The special thing about the polynomial $(st^2 - st\,t - t^2)^2$ is that it denotes a nontrivial *constant* in the difference ring $k\{t\}/\langle\langle sst - st - t\rangle\rangle$:

$$\begin{aligned} s\left((st^2 - st\,t - t^2)^2\right) &= (sst^2 - sst\,st - st^2)^2 = ((st+t)^2 - (st+t)st - st^2)^2\\ &= (-st^2 + st\,t + t^2)^2 = (st^2 - st\,t - t^2).\end{aligned}$$

Constants and algebraic relations are closely related, as we will see in more detail in the following section.

8.2 Constants and Algebraic Dependencies

Let $f_1(n), \dots, f_m(n)$ be admissible sequences, and define

$$\varphi\colon \mathbb{K}\{t_1,\dots,t_m\} \to \mathbb{K}^{\mathbb{N}}, \qquad t_i \mapsto f_i(n)$$

as usual. For every subideal $\mathfrak{a} \subseteq \ker\varphi$, there is an induced difference homomorphism

$$\bar{\varphi}\colon \mathbb{K}\{t_1,\dots,t_m\}/\mathfrak{a} \to \mathbb{K}^{\mathbb{N}}$$

by Theorem 2.15.

For any ring $R = \mathbb{K}\{t_1,\dots,t_m\}/\mathfrak{a}$ defined in this way, we obviously have

$$\langle\langle x - \bar{\varphi}(x) : x \in \operatorname{const} R\rangle\rangle \subseteq \ker\bar{\varphi}.$$

We can therefore apply Algorithm 8.1 for finding algebraic dependencies, as follows. Let $\mathfrak{a}$ be the associated difference ideal of the admissible system by which the $f_1(n),\dots,f_m(n)$ are defined. Find a solution of

$$f(n+1) - f(n) = 0$$

as shown in the previous section. If a solution $x \in k\{t_1,\dots,t_m\}/\mathfrak{a}$ is found, replace $\mathfrak{a}$ by the ideal $\mathfrak{a} + \langle\langle x - \varphi(x)\rangle\rangle$, and proceed.

There is a temptation to believe that this procedure will eventually yield sufficiently many relations to generate the whole annihilating ideal $\ker\varphi$. Though this is often the case, it fails to hold in degenerate cases.

Example 8.5

(1) Let $\varphi\colon \mathbb{K}\{t_1,t_2\} \to \mathbb{K}^{\mathbb{N}}$ be such that $\varphi(t_1) = F_n$ and $\varphi(t_2) = (-1)^n$.
Taking $\mathfrak{a} = \langle\langle sst_1 - st_1 - t_1, st_2 + t_2\rangle\rangle$, we have

$$\operatorname{const}\mathbb{K}\{t_1,t_2\}/\mathfrak{a} \cong \mathbb{K}[(st_1^2 - st_1t_1 - t_1)t_2, (st_1^2 - st_1t_1 - t_1)^2, t_2^2]$$

and

$$\begin{aligned}\langle\langle x - \varphi(x) : x \in \operatorname{const}\mathbb{K}\{t_1,t_2\}/\mathfrak{a}\rangle\rangle &= \langle\langle (st_1^2 - st_1t_1 - t_1)t_2 - 1, (st_1^2 - st_1t_1 - t_1)^2 - 1, t_2^2 - 1\rangle\rangle\\ &= \langle\langle t_2^2 - 1, (st_1^2 - st_1t_1 - t_1) - t_2\rangle\rangle = \ker\varphi\end{aligned}$$

(Note that $\mathbb{K}\{t_1,t_2\}/\mathfrak{a} \cong \mathbb{K}[t_1, st_1, t_2]$).
In this example, all algebraic dependencies originate from the constants in the difference ring $\mathbb{K}\{t_1,t_2\}/\mathfrak{a}$.

(2) Let $f(n)$ be defined via

$$f(n+1) = 2f(n) \quad (n \geq 1), \qquad f(1) = 0.$$

Let $\varphi\colon \mathbb{K}\{t\} \to \mathbb{K}^{\mathbb{N}}$ be such that $\varphi(t) = f(n)$. It is easy to see that $\operatorname{const}\mathbb{K}\{t\}/\mathfrak{a} = \mathbb{K}$: Let $x \in \operatorname{const}\mathbb{K}\{t\}/\mathfrak{a}$, say $x = \alpha_0 + \alpha_1 t + \cdots + \alpha_d t^d$. Then $sx = \alpha_0 + 2\alpha_1 t + \cdots + 2^d\alpha_d t^d$. By $sx = x$, it follows that $\alpha_1 = \alpha_2 = \cdots = \alpha_d = 0$, hence $x = \alpha_0 \in \mathbb{K}$.
It follows that

$$\langle\langle x - \varphi(x) : x \in \operatorname{const}\mathbb{K}\{t\}/\mathfrak{a}\rangle\rangle = \{0\}.$$

On the other hand, we have $f(n) = 0$ for all $n \in \mathbb{N}$, hence $t \in \ker\varphi$, hence $\ker\varphi \neq \{0\}$.
This discrepancy is due to the "unlucky" choice of the initial value $f(1) = 0$. For any other initial value, we have $\ker\varphi = \{0\}$, as expected from the constants.

8.3 Indefinite Summation

We now turn to the summation problem. First we consider *indefinite* sums, i.e., sums of the form $\sum_{k=1}^{n} f(k)$ where $f(k)$ is independent of the summation bound n. Finding a closed form for the sum amounts to finding a solution of the first order inhomogeneous linear difference equation

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

which is also known as the *telescoping equation.* The solution of the telescoping equation is uniquely determined only up to an additive constant. If $F(n)$ is any solution, then the value of the sum is given by $F(n) - F(1)$.

There are known algorithms for solving the telescoping equation in various demains. Some classical algorithms are mentioned in the introduction (cf. p. 1). Most of these algorithms at some point transform the problem into the problem of finding a solution x of a difference equation

$$a_0 x + a_1 s(x) = f$$

in some difference ring of the form $\mathbb{K}[t]$, where $\mathbb{K}$ may be a nontrivial difference field. A major incredience is a *degree bound* for the solution x, i.e., a number d such that all solutions x of the equation have degree at most d. If such a degree bound can be effectively computed from the inhomogeneous part f, then the solution can be found by plugging an undetermined ansatz $x = \alpha_0 + \alpha_1 t + \cdots + \alpha_d t^d$ into the equation and comparing coefficients.

If $f_1(n), \dots, f_m(n)$ are admissible sequences, and $f(n)$ is a polynomial in the $f_i(n)$, then we can search for closed forms of the sum $\sum_{k=1}^{n} f(k)$ in terms of the $f_i(n)$ by applying Algorithm 8.1 to the telescoping equation in the difference ring $\mathbb{K}\{t_1, \dots, t_m\}/\mathfrak{a}$, for some known difference ideal $\mathfrak{a} \subseteq \operatorname{ann}(f_1(n), \dots, f_m(n))$ containing at least the associated difference ideal of the admissible system by which the $f_i(n)$ are defined. Bounding the total degree of the solutions of the telescoping equation causes difficulties in this setting. As we have seen in Example 8.4.(3), there might be an infinite set of solutions with unbounded total degree, due to nontrivial constants in the ring $\mathbb{K}\{t_1, \dots, t_m\}/\mathfrak{a}$. We would need to be able to compute a bound d with the property that, if the telescoping equation has any solution at all, then it already has a solution of total degree at most d.

Unfortunately, we have not been able to come up with such a degree bound. From a theoretic viewpoint, summation via Algorithm 8.1 is therefore not superior to successive approximation of the annihilating ideal by means of the algorithms described in Chapter 6. In both cases, we have a semidecision procedure that terminates with a closed form for the given sum, if there is one, and that does not terminate, if no closed form exists.

The two approaches differ with respect to their performance. Experiments that we have carried out with the software described in Chapter 9 suggest that indefinite summation using Algorithm 8.1 is usually faster using the methods of Chapter 6. This is especially true when the involved sequences contain parameters. A few examples with timings measured on our machine (2GHz, 1GB) are given next.

Example 8.6 Consider the following indefinite sums involving orthogonal polynomials

$$\sum_{k=0}^{n} L_k^{\alpha}(x) = \frac{1}{x}\big((n+\alpha+x)L_n^{\alpha}(x) - (n+1)L_{n+1}^{\alpha}(x)\big) \tag{8.4}$$

$$\sum_{k=1}^{n} (1+2k)P_k(x)^2 = \frac{(n+1)^2}{1-x^2}\Big(P_n(x)^2 - 2xP_n(x)P_{n+1}(x) + P_{n+1}(x)^2\Big) - 1 \tag{8.5}$$

$$\begin{aligned}\sum_{k=0}^{n} (k+2mk+2m)C_k^m(x)P_k(x) = \frac{n+1}{1-x^2}\Big(&(n+2m)C_n^m(x)\big(P_n(x) - xP_{n+1}(x)\big) \\ &- (n+1)C_{n+1}^m(x)\big(xP_n(x) - P_{n+1}(x)\big)\Big)\end{aligned} \tag{8.6}$$

$$\begin{aligned}\sum_{k=1}^{n} k\sum_{i=1}^{k} P_i(x) = \frac{1}{8(x-1)}\Big(&1 - 3x + (2n^2+n-1)P_n(x) - (2n^2-n-3)P_{n+1}(x) \\ &+ (1-3x+4n(x-1)(n+1))\sum_{k=1}^{n} P_k(x)\Big)\end{aligned} \tag{8.7}$$

$$\begin{aligned}\sum_{k=1}^{n}\sum_{i=1}^{k} iP_i(x) = \frac{1}{2(x-1)^2}\Big(&(n+1)(n(x-1)+x)P_n(x) - (n(x-1)+1)P_{n+1}(x) \\ &+ (x-1)(3x-1+2n(x-1))\sum_{k=1}^{n} kP_k(x)\Big)\end{aligned} \tag{8.8}$$

Here $P_n(x)$, $L_n^{\alpha}(x)$ and $C_n^m(x)$ denote the Legendre, Laguerre, and Gegenbauer polynomials, respectively (cf. p. 11). All of the closed forms above can be discovered both by Algorithm 8.1 (A) and successive approximation of the annihilating ideal (B). Below are the timings for these two methods.

eq.	(8.4)	(8.5)	(8.6)	(8.7)	(8.8)
A	0.42s	3.55s	105.0s	2.18s	2.47s
B	5.81s	15.01s	435.9s	13.94s	8.01s

The runtime bottleneck are in both cases the linear systems that have to be solved. The same solver—a little-optimized implementation of Bareiss elimination—was used for both A and B. While the systems in B are large and dense, we face sparse systems in A, so that further improvements might be possible.

8.4 Definite Summation

Indefinite sums $F(n) = \sum_{k=1}^{n} f(k)$ satisfy the telescoping equation $F(n+1) = F(n) + f(n+1)$. This equation was used in the previous section for finding closed form representations of indefinite sums, and also earlier as defining relation of the sum in admissible systems. In contrast to indefinite sums, in a definite sum the summand sequence may depend not only on the summation index k but also on the summation bound n, i.e., $F(n) = \sum_{k=1}^{n} f(n,k)$. Such sums do not satisfy the telescoping equation in general, and there is no obvious alternative that may serve as a defining relation for a given definite sum $F(n)$.

The main task in definite summation is precisely to fill this gap: given a bivariate summand sequence $f(n,k)$, find a defining difference equation for the definite sum $F(n) = \sum_{k=1}^{n} f(n,k)$. Once such a relation is available, it can be used as a definition for $F(n)$ in the input of other algorithms, which, for example, compute a closed form representation of $F(n)$. The method of *creative telescoping,* propagated by Zeilberger (1991), turns an algorithm for indefinite summation into a method for finding a linear difference equation that a given definite sum fulfills. Zeilberger's original algorithm extended Gosper's algorithm (Gosper, 1978) for indefinite hypergeometric summation to an algorithm for definite hypergeometric summation. Later, creative telescoping was also applied by Chyzak (2000) for holonomic functions and by Schneider (2001, 2005) for extending Karr's indefinite summation algorithm (Karr, 1981, 1985) to definite sums.

The method of creative telescoping works as follows. In a first step, find an inhomogeneous linear difference equation of the form

$$g(n,k+1) - g(n,k) = c_0(n)f(n,k) + c_1(n)f(n+1,k) + \cdots + c_r(n)f(n+r,k), \tag{8.9}$$

where the $c_i(n)$ are independent of k. (Both the $c_i(n)$ and $g(n,k)$ have to be found, the $f(n+i,k)$ are given.) This is done by applying an indefinite summation algorithm to the right hand side, for indeterminates in place of the coefficients c_i and some fixed $r \in \mathbb{N}_0$. The method only applies if the summation algorithm can be extended in such a way that it also computes values c_i for which the sum of the right hand side over k actually has a closed form $g(n,k)$. If no such coefficients exist, the method is applied again with a greater value of r.

In a second step, the desired difference equation for the sum $F(n)$ is obtained by summing (8.9) for k from 1 to $n+r$:

$$\sum_{k=1}^{n+r}(g(n,k+1) - g(n,k)) = a_0(n)\sum_{k=1}^{n+r} f(n,k) + a_1(n)\sum_{k=1}^{n+r} f(n+1,k) + \cdots + a_r(n)\sum_{k=1}^{n+r} f(n+r,k)$$

gives the linear difference equation

$$\begin{aligned}
g(n,n+r+1) - g(n,1) = {} & a_0(n)(F(n) + f(n,n+1) + \cdots + f(n,n+r)) \\
& + a_1(n)(F(n+1) + f(n+1,n+2) + \cdots + f(n+1,n+r)) \\
& + \cdots \\
& + a_{r-1}(n)(F(n+r-1) + f(n+r-1,n+r)) \\
& + a_r(n)F(n+r)
\end{aligned}$$

for the sum $F(n)$.

Algorithm 8.1 can be extended in such a way that creative telescoping becomes possible for arbitrary admissible sequences. We consider bivariate sequences $f(n,k)$ which are admissible both as sequence in n (regarding k as a fixed parameter) and in k (regarding n as a fixed parameter). In addition, $f(n,k)$ must be such that the diagonal sequences $f(n,n+i)$ $(i \geq 0)$ are admissible, and defining admissible systems are available. For example, if $f(n)$ is an admissible sequence, then $f(an+bk)$ with $a,b \in \mathbb{N}_0$ fixed is a typical candidate for a summand sequence.

In order to find a recurrence of the type (8.9), the following straightforward extension of Algorithm 8.1 can be applied. The difference to the original procedure is only that, informally speaking, the inhomogeneous part is broken into several pieces which are homogenized independently. That is, we seek solutions $(x, \xi^{(1)}, \dots, \xi^{(v)}) \in R \times \mathbb{K}^v$ of difference equations of the form

$$a_0 x + a_1 s(x) + \cdots + a_u s^u(x) = \xi_1 f_1 + \xi_2 f_2 + \cdots + \xi_r f_v, \tag{8.10}$$

where R is as in Section 8.1 and $a_0, \dots, a_u, f_1, \dots, f_v \in R$ are given.

Algorithm 8.7 (Solving linear difference equations)
Input: $\mathfrak{a}_0, p_1, \dots, p_m$ as in Procedure 8.1; $f_1, \dots, f_v, a_0, \dots, a_u \in R := \mathbb{K}\{t_1, \dots, t_m\}/\mathfrak{a}$, where $\mathfrak{a} = \langle\langle \mathfrak{a}_0 \rangle\rangle + \langle\langle p_1, \dots, p_m \rangle\rangle$; a number $D \in \mathbb{N}_0$
Output: Linearly independent pairs $(x_n, \xi_n^{(1)}, \dots, \xi_n^{(v)}) \in R \times \mathbb{K}^v$ $(n = 1,2,3,\dots)$ with $a_0 x_n + a_1 s(x_n) + \cdots + a_u s^u(x_n) = \xi_n^{(1)} f_1 + \cdots + \xi_n^{(v)} f_v$ $(n = 1,2,3,\dots)$

$S = \text{Gröbnerbasis}(\text{Syz}(a_0, \dots, a_u, -f_1, \dots, -f_v))$
$B = \emptyset$
for $d = 1$ **to** D **do**
  Let T be the set of terms τ in $\mathbb{K}\{t_1, \dots, t_m\}_r$ with $\deg \tau \le d$
  $T := T \setminus \mathrm{LT}\langle \mathfrak{a}_0 \rangle$
  $T = \{\tau_1, \dots, \tau_\ell\} := T \setminus \mathrm{LT}\, B$
  Let $x := \alpha_1 \tau_1 + \cdots + \alpha_\ell \tau_\ell$ where the α_i are indeterminates
  Let $(q_0, \dots, q_{u+v})$ be the normal form of $(x, sx, \dots, s^u x, \xi^{(1)}, \dots, \xi^{(v)})$ w.r.t. S
    where $\xi = (\xi^{(1)}, \dots, \xi^{(v)})$ is a vector of indeterminates
  Solve the linear system for the α_i and $\xi^{(i)}$
    obtained by comparing the coefficients of the q_i to 0
  Obtain $(x_1, \xi_1), \dots, (x_s, \xi_s)$ from (x, ξ)
    by spezializing the α_i and $\xi^{(i)}$ to the solutions of this system
  $B := B \cup \{(x_1, \xi_1), \dots, (x_s, \xi_s)\}$
return B

Theorem 8.8 The output $B = \{\, (x_i, \xi_i) : i \in I \,\} \subseteq \mathbb{K}\{t_1, \dots, t_m\}_r/\mathfrak{a}_0 \times \mathbb{K}^v$ of Algorithm 8.7 generates the $\mathbb{K}$-vector space of all solutions (x_i, ξ_i) of (8.10) with $\deg x_i \le D$.

Proof Identical to the proof of Theorem 8.2. ∎

Example 8.9 Consider the definite hypergeometric sum

$$F(n) = \sum_{k=0}^{n} \binom{n}{k}.$$

(See Section 8.5 below for some more sophisticated examples.) We apply creative telescoping and Algorithm 8.7 for finding a difference equation satisfied by $F(n)$. First, we have to find $g(n,k)$ and $c_0(n), \dots, c_1(n)$ such that

$$\begin{aligned} g(n,k+1) - g(n) &= c_0(n) \binom{n}{k} + c_1(n) \binom{n+1}{k} \\ &= c_0(n) \binom{n}{k} + c_1(n) \frac{n+1}{n+1-k} \binom{n}{k}. \end{aligned}$$

A suitable difference ring is $R := \mathbb{Q}(n)\{t_1, t_2, t_3\}/\mathfrak{a}$ with

$$\mathfrak{a} = \langle\langle st_1 - t_1 - 1, st_2 - (n - t_1)t_3t_2, t_3t_1 - 1, t_4(n + 1 - t_1) - t_1\rangle\rangle.$$

Here t_1, t_2, t_3, and t_4 represent k, $\binom{n}{k}$, $1/k$ and $1/(n+1-k)$, respectively, and n is considered as a constant.

Applying Algorithm 8.7 to the difference equation

$$s(x) - x = \xi^{(1)}t_2 + \xi^{(2)}t_2t_4,$$

we obtain quickly two linearly independent solutions $(1,0,0)$ and $(t_1t_2t_4, 2, -1)$ for $(x, \xi^{(1)}, \xi^{(2)})$. From the latter solution, it follows that

$$\frac{(k+1)}{1-(k+1)+n}\binom{n}{k+1} - \frac{k}{1-k+n}\binom{n}{k} = 2\binom{n}{k} - \binom{n+1}{k},$$

and, by summation $k = 0, \ldots, n+1$,

$$\underbrace{\frac{n+2}{1-(n+2)+n}\binom{n}{n+2} - \frac{0}{n+1}\binom{n}{0}}_{=0} = 2(F(n) + \binom{n}{n+1}) - F(n+1).$$

This implies the recurrence $F(n+1) = 2F(n)$ for $F(n)$, in accordance with the output of Zeilberger's algorithm for this example.

Example 8.10 A simple non-hypergeometric example is the sum

$$F(n) := \sum_{k=0}^{n} \binom{n}{k} F_k,$$

where F_k is the kth Fibonacci number. Here, the summand sequence

$$f(n,k) := \binom{n}{k} F_k$$

does not admit a first order recurrence—at least not one with with coefficients of small degree. For order two, we find

$$\Delta_k \frac{k\,((2k-3-n)F_k - (k+n+2)F_{k+1})}{(k-n-1)(k-n-2)}\binom{n}{k} = -f(n,k) + 3f(n+1,k) - f(n+2,k)$$

using Algorithm 8.7 (Δ_k denotes the forward difference operator $\Delta_k f(n,k) = f(n,k+1) - f(n,k)$). This implies the recurrence

$$F(n+2) = 3F(n+1) - F(n)$$

for the original sum. Solutions to this recurrence can be found with Algorithm 8.1. For instance, in terms of F_n, we find the general solution

$$F(n) = c_1(2F_{n+1} - F_n)F_n + c_2(F_n^2 + F_{n+1}^2)$$

with indetermined constants c_1, c_2. By comparing the values of the general solution and the original sum for some small values of n, we obtain a linear system by which the values of c_1, c_2 can be determined. It turns out that $c_1 = 1, c_2 = 0$ is the right choice, i.e.,

$$\sum_{k=0}^{n} \binom{n}{k} F_k = (2F_{n+1} - F_n)F_n \qquad (n \geq 0).$$

8.5 Examples

Several problems appearing in the literature can be tackled with the algorithms described in this chapter. A small selection is given below. Some of these sums can be done by other algorithms as well, while others are done algorithmically for the first time.

Example 8.11 Indefinite sums.

(1) Diaz (2001) asks for the values of the infinite sums

$$\sum_{k=0}^{\infty} \frac{1+L_{k+1}}{L_k L_{k+2}} \qquad \text{and} \qquad \sum_{k=0}^{\infty} \frac{L_{k-1} L_{k+2}}{L_k^2 L_{k+1}^2}.$$

Using our indefinite summation algorithm, we find

$$\sum_{k=0}^{n} \frac{1+L_{k+1}}{L_k L_{k+2}} = 2 - \frac{1+L_n+L_{n+1}}{L_{n+1}(L_n+L_{n+1})} \xrightarrow{n\to\infty} 2,$$
$$\sum_{k=0}^{n} \frac{L_{k-1} L_{k+2}}{L_k^2 L_{k+1}^2} = \frac{1}{4} - \frac{1}{L_{n+1}^2} \xrightarrow{n\to\infty} \frac{1}{4}.$$

(2) The identities

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

(Graham *et al.*, 1994, Exercise 6.61) can be found as well.

(3) For the sequence of complex functions $Q_n(x)$ defined by

$$Q_n(x) := \frac{1}{2}\int_{-1}^{1} \frac{P_n(t)}{x-t}\,dt \quad (n \geq 0),$$

we have the indefinite sum identity

$$\sum_{k=0}^{n} (2k+1) Q_k(x) Q_k(y) = \frac{Q_0(y) - Q_0(x)}{x-y} + (n+1)\frac{Q_{n+1}(x)Q_n(y) - Q_n(x)Q_{n+1}(y)}{x-y}$$

(Andrews *et al.*, 1999, Exercise 5.29.(e)), which our algorithm can discover using the recurrence

$$Q_{n+2}(x) = \frac{1}{n+2}\big((2n+3)Q_{n+1}(x) - (n+1)Q_n(x)\big)$$

as definition for $Q_n(x)$.

(4) Consider the summation problem

$$\sum_{k=1}^{n} k^2 \mathrm{H}_{n+k}$$

posed in (Graham *et al.*, 1994, Exercise 6.69). Though this sum looks like a definite sum, it can be done via indefinite summation. For, we can first compute the more general identity

$$\begin{aligned}\sum_{k=1}^{n} k^2 \mathrm{H}_{a+k} = {} & \frac{-4(a+1)n^3 - (12a^3+24a^2+11a-1)n + (6a^2+9a+3)n^2 - 36}{36(a+1)} \\ & - \tfrac{1}{6}(6+a+3a^2+2a^3)\mathrm{H}_a + \mathrm{H}_{a+1} \\ & + \tfrac{1}{6}(a+3a^2+2a^3+n+3n^2+2n^3)\mathrm{H}_{a+n},\end{aligned}$$

and then specialize $a = n$. This gives the solution

$$\sum_{k=1}^{n} k^2 \mathrm{H}_{n+k} = \tfrac{1}{36} n(n+1)\big(6(2n+1)(2\mathrm{H}_{2n} - \mathrm{H}_n) - 10n + 1\big).$$

Example 8.12 Definite sums.

(1) (Seiffert, 1994b)

$$\sum_{k=0}^{n} F_{a(k+1)} P_{a(n-k+1)} = \frac{F_a P_{a(n+2)} - P_a F_{a(n+2)}}{F_a - 2F_{a+1} + 4P_a - 4P_{a+1}} \qquad (n \geq 0).$$

Here, we treat a as an element of the constant field.

(2) Table entries as in (Hoggatt, 1979; Vajda, 1989) can be (re)discovered automatically. The following Fibonacci sum identities were obtained by applying the summation algorithm to artificial left hand sides.

$$\begin{aligned}
\sum_{k=0}^{n} \binom{n}{k} F_k &= (2F_{n+1} - F_n)F_n, \\
\sum_{k=0}^{n} F_{n^2+k} &= F_{n^2+n+3} - F_{n^2+n+1} - F_{n^2+1}, \\
\sum_{k=0}^{n} F_k F_{n-k} &= \tfrac{1}{5}\big(n(2F_{n+1} - F_n) - F_n\big), \\
\sum_{k=0}^{n} F_k F_{n+k} &= \tfrac{1}{2} F_n \big(F_n^2 - 1 + 3F_{n+1}^2 - F_n F_{n+1}\big), \\
\sum_{k=0}^{n} F_{n+k} x^k &= -\frac{x^{n+1}((1-x)F_n^2 + F_{n+1}^2) - xF_{n+1} + (2x^{n+2}F_{n+1} + x - 1)F_n}{x^2 + x - 1}, \\
\sum_{k=0}^{n} (-1)^k F_{n+2k} &= (-1)^n F_{n+1}\big((-1)^n + 3F_n F_{n+1} - F_{n+1}^2\big), \\
\sum_{k=0}^{n} (-1)^k F_k^2 F_{n-k}^2 &= \tfrac{1}{6}\big(1 + (-1)^n\big)\big(F_{n+1}^2 - 3F_n F_{n+1} - 1\big), \qquad \text{etc.}
\end{aligned}$$

(3) For the definite sum $S(n) := \sum_{k=0}^{n} \binom{n+k}{k} P_k(x)$, we obtained the recurrence

$$\begin{aligned} S(n+2) = {} & \frac{2n+3}{2(n+2)} S(n) + \frac{n+1}{2(n+2)(x-1)} S(n+1) \\ & - \binom{2n+1}{n} \frac{(3n+5)P_{n+1}(x) + 2\big((7-8x)n - 12x + 10\big)P_{n+2}(x)}{2(n+2)(x-1)}. \end{aligned}$$

There does, however, not seem to be a closed form of $S(n)$ in terms of $n, n!, (2n)!$, and $P_n(x)$.

Closed form representations of many definite sums involve expressions that do not appear as subexpressions of the summand sequence. Also these can be found with our algorithms, if it is known which additional sequences appear on the right hand side. This, however, is often hard to guess. Blind experiments are of course possible, but they are barely sucessful on complicated sums. Even though, we can use creative telescoping for *proving* conjectured identitites for definite sums, as long as a recurrence for the sum can be found and the right hand side is admissible. The recurrence obtained by creative telescoping is used as defining relation for the definite sum, and the zero equivalence tester (Algorithm 4.2) is applied to the difference of left hand side and right hand side.

Example 8.13 In order to prove the (artificial) identity

$$\sum_{k=0}^{n} \binom{n}{k} F_k^2 = \frac{F_{n+1}(2F_n + F_{n+1}) + 1}{5F_{n+1}^2 - 2(-1)^n - 3} \sum_{k=0}^{n} \binom{n}{k} F_{2k} \qquad (n \geq 1) \tag{8.11}$$

we first compute recurrences for the sums

$$F(n) := \sum_{k=0}^{n} \binom{n}{k} F_k^2 \quad \text{and} \quad G(n) := \sum_{k=0}^{n} \binom{n}{k} F_{2k}.$$

The algorithm delivers

$$\begin{aligned} F(n+2) &= 5F(n+1) - 5F(n), \qquad & F(1) = 1, F(2) = 3, \\ G(n+2) &= 5G(n+1) - 5G(n), \qquad & G(1) = 1, G(2) = 5. \end{aligned}$$

With these recurrences at hand, proving (8.11) is trivial.

(Remark: An application of Algorithm 7.18 shows that neither $F(n)$ nor $G(n)$ can be expressed in terms of Fibonacci numbers only. This algorithm was also used to discover (8.11).)

9 Implementation: The SumCracker Package

For the purpose of being able to experiment with the algorithms presented in this thesis, we have undertaken an implementation in form of a Mathematica package. The package is named SumCracker and was developed for Mathematica 5.0. It is available upon request from the author.

The algorithm of Chapter 7, including our variant of Ge's algorithm, has been implemented in a separate package that will be described elsewhere (Kauers and Zimmermann, 2005b). All other algorithms are implemented in the SumCracker package, though not every algorithm is available through the user interface.

In this final chapter, we describe the usage of the package. Though being designed as a proto type for experimenting, we believe that the package might also be useful in practice.

The description below refers to the current version of the package. The input/output format of the various commands may, however, be subject to change in future versions.

9.1 Overview

In this section, we will show the usage of our package at some concrete examples. The examples given below were chosen such that virtually no computation time is needed to obtain the output. Detailed descriptions of the expressions, commands, and options will be given in subsequent sections.

One of the main features of the package is that it knows how to automatically transform a lot of expressions into defining admissible systems. Additional sequences can be explicitly defined by the user, and user defined sequences can be mixed with builtin sequences.

As an example, the ApproximateAnnihilator command computes an approximation of the annihilating difference ideal from below, as described in Chapter 6. In order to compute the algebraic dependencies of, say, the Fibonacci numbers with $(-1)^n$, we say

In[1]:= $\text{ApproximateAnnihilator}[\{(-1)^n, \text{Fibonacci}[n], \text{Fibonacci}[n+1]\}, t]$

Out[1]= $$\{t_1 + t_2^2 + t_2t_3 - t_3^2, t_1^2 - 1\}$$

This output means that $\langle t_1 + t_2^2 + t_2t_3 - t_3^2, t_1^2 - 1\rangle \subseteq \ker\varphi$ where $\varphi\colon \mathbb{K}[t_1,t_2,t_3] \to \mathbb{K}^{\mathbb{N}}$ is defined via $t_1 \mapsto (-1)^n, t_2 \mapsto F_n, t_3 \mapsto F_{n+1}$. If the identifier t is omitted, the original expressions are used in the output:

In[2]:= $\text{ApproximateAnnihilator}[\{(-1)^n, \text{Fibonacci}[n], \text{Fibonacci}[n+1]\}]$

Out[2]= $$\{(-1)^n + F_n^2 + F_nF_{n+1} - F_{n+1}^2, (-1)^{2n} - 1\}$$

This form is often convenient for further processing, but note that information might get lost, due to Mathematica automatic simplification.

In[3]:= $\text{ApproximateAnnihilator}[\{n,n\},t]$

Out[3]= $$\{t_1 - t_2\}$$

In[4]:= $\text{ApproximateAnnihilator}[\{n,n\}]$

Out[4]= $$\{0\}$$

The most frequent use of the annihilating ideal concerns answering the representation problem: express something in terms of something else. The Crack command provides an implementation which is fine tuned for this purpose. This command takes an expression and "breaks" it into smaller ones. As a trivial example, a closed form for the Popov sum can be obtained as follows.

In[5]:= $\text{Crack}[\text{SUM}[k^2,\{k,0,n\}]]$

Out[5]= $$\tfrac{1}{6}(2n^3+3n^2+n)$$

Observe that we use the symbol SUM for denoting sums, in order to avoid conflicts with Mathematica's summation command Sum. If no representation is found, the original expression is returned.

In[6]:= $\text{Crack}[\text{Fibonacci}[n]]$

Out[6]= $$F_n$$

Whether or not a different representation can be found depends on the sequences that are allowed to appear in the resulting closed form. By default, Crack allows all subexpressions of the given expression to appear on the right hand side. For instance, n is considered as a subexpression of $\sum_{k=0}^{n} k^2$, because it appears in the summand (though it is called k there). As F_n does not have subexpressions in this sense, Crack does not find a simpler representation. Subexpression for the right hand side can be specified via the option Into, whose default value is set to Automatic. For instance, in order to express the Fibonacci numbers in terms of Lucas numbers, we say

In[7]:= $\text{Crack}[\text{Fibonacci}[n],\text{Into}\to\{\text{Lucas}[n]\}]$

Out[7]= $$\tfrac{1}{5}(2L_{n+1}-L_n)$$

We have defined the Fibonacci sequence with the initial values $F_0=0, F_1=1$. Some authors prefer the definition $F_0=1, F_1=1$. The representation of these Fibonacci numbers in terms of Lucas numbers can be obtained by explicitly defining a new sequence by means of the Where option. Even symbolic initial values are possible.

In[8]:= $\text{Crack}[f[n],\text{Into}\to\{\text{Lucas}[n]\},\text{Where}\to\{f[n+2]==f[n+1]+f[n], f[0]==1, f[1]==1\}]$

Out[8]= $$\tfrac{1}{5}(2L_n+L_{n+1})$$

In[9]:= $\text{Crack}[f[n], \text{Into} \to \{\text{Lucas}[n]\}, \text{Where} \to \{f[n+2] == f[n+1] + f[n]\}]$

Out[9]= $$\frac{1}{5}(3f(0)L_n - f(1)L_n - f(0)L_{n+1} + 2f(1)L_{n+1})$$

In[10]:= $\text{Crack}[\text{Lucas}[n], \text{Into} \to \{f[n]\}, \text{Where} \to \{f[n+2] == f[n+1] + f[n]\}]$

Out[10]= $$\frac{2f(0)f(n) + f(1)f(n) + f(0)f(n+1) - 2f(1)f(n+1)}{f(0)^2 + f(0)f(1) - f(1)^2}$$

The Crack command can be used for solving nonlinear difference equations, as in Example 6.21 on page 72.

In[11]:= $\text{Crack}[u[n], \text{Into} \to \{\text{Fibonacci}[n]\}, \text{Where} \to \{u[n+1] == \frac{3u[n]+1}{5u[n]+3}, u[1] == 1\}]$

Out[11]= $$\frac{-2F_n^2 + 2F_nF_{n+1} - F_{n+1}^2}{4F_n^2 - 6F_nF_{n+1} + F_{n+1}^2}$$

In[12]:= $\text{Crack}[u[n], \text{Into} \to \{\text{Lucas}[n]\}, \text{Where} \to \{u[n+1] == \frac{3u[n]+1}{5u[n]+3}, u[1] == 1\}]$

Out[12]= $$\frac{-2L_n^2 + 2L_nL_{n+1} - L_{n+1}^2}{4L_n^2 - 6L_nL_{n+1} + L_{n+1}^2}$$

Builtin expressions and user definitions can be mixed. For instance

In[13]:= $\text{Crack}[f[n], \text{Where} \to \{f[n+2] == f[n] + f[n+1] + \text{Fibonacci}[n+2], f[0] == 0, f[1] == 1\},$
$\text{Into} \to \{n, \text{Fibonacci}[n]\}]$

Out[13]= $$\frac{1}{5}(2F_n + 2nF_n + nF_{n+1})$$

solves Exercise 7.26 of Graham *et al.* (1994) automatically.

Another special purpose command for searching for elements of the annihilating ideal, which are of a particular shape, is LinearRecurrence. This command searches for a (possibly inhomogeneous) linear recurrence whose solution is the sequence provided as input. For instance, the recurrence

In[14]:= $\text{LinearRecurrence}[n\,\text{Fibonacci}[n]^2]$

Out[14]= $$\text{SUM}(n) - 4\,\text{SUM}(n+1) + 10\,\text{SUM}(n+3) - 4\,\text{SUM}(n+5) + \text{SUM}(n+6) == 0$$

has the solution $\text{SUM}[n] = nF_n^2$. The symbol SUM is used by default as function symbol of the resulting recurrence. This can be changed to a different symbol using the option Head. The expressions that may appear in the coefficients are chosen automatically. For overriding the automatic selection, use the In option.

In[15]:= $\text{LinearRecurrence}[n\,\text{Fibonacci}[n]^2, \text{In} \to \{n\}]$

Out[15]=
$$(n^3+6n^2+11n+6)\,\text{SUM}(n) - (2n^3+10n^2+12n)\,\text{SUM}(n+1) - (2n^3+8n^2+6n)\,\text{SUM}(n+2) + (n^3+3n^2+2n)\,\text{SUM}(n+3) == 0$$

In[16]:= $\text{LinearRecurrence}[n\,\text{Fibonacci}[n]^2, \text{In} \to \{\text{Fibonacci}[n]\}]$

Out[16]=
$$F_n^2 F_{n+1}^2 + F_{n+1}^2\,\text{SUM}(n) - F_n^2\,\text{SUM}(n+1) == 0$$

In[17]:= $\text{LinearRecurrence}[n\,\text{Fibonacci}[n]^2, \text{In} \to \{n, \text{Fibonacci}[n]\}]$

Out[17]=
$$nF_n^2 - \text{SUM}(n) == 0$$

If no linear recurrence exists, the procedure does not terminate.

In[18]:= $\text{LinearRecurrence}[2^{2^n}]$

Out[18]= \$Aborted

While waiting for an output, one is often in doubt whether the procedure is really not terminating or it is just stuck solving a big linear system, or anything. It the latter case, it is desirable to get a clue as to whether it pays off to continue waiting for a result. In such situations, the option Infolevel is informative. By this option, the verbosity of a computation can be governed. Default value is 0 (= no information), higher integer values lead to more refined output.

In[19]:= $\text{LinearRecurrence}[2^{2^n}, \text{Infolevel} \to 1]$

Creating difference ring and homomorphism...
Translating expressions to difference polynomials...
Choosing a degree bound...
Considering order 0...
Considering order 1...
Considering order 2...
Considering order 3...
Considering order 4...
Considering order 5...
Considering order 6...
Considering order 7...
Considering order 8...
Considering order 9...
Considering order 10...
Considering order 11...
Considering order 12...
Considering order 13...
Considering order 14...

Out[19]= $Aborted

In[20]:= LinearRecurrence$[2^{2^n}, \text{Infolevel} \to \text{Infinity}]$

Creating difference ring and homomorphism...
  Extracting dependent variable...
    Taking n.
  Normalizing exponentials...
  Cracking expression into difference system...
    Determining bounded variables...
    Traversing expression tree...
      Function symbol encountered. Searching for definition...
        Built-in definition found.
    Creating expression transformators...
    Expression successfully traversed.
  Creating difference ring...

(... 10 pages of further output omitted...)

Considering order 14...
  Computing initial bases...
    Extracting relations from the difference ring...
    Adding definitions of coefficient expressions...
    Adding defining relations for multiplicative inverses...
  Entering completion loop...
    Searching for relation of degree up to 1...
      Creating ansatz polynomial...
        inhomogeneous part...
        homogeneous part...
      Constructing a structure set...
        Setting up linear system (12 x 13) ...
        Solving linear system...
      Structure set predicts existence of 0 nontrivial relations.

Out[20]= $Aborted

The Infolevel option can also be helpful for locating the origin of incomprehensible warnings that Mathematica prints out during some computation. All SumCracker commands support this option.

The command LinearRecurrence also contains the implementation of creative telescoping (Section 8.4). Creative telescoping is always executed when the expression is a definite sum, i.e., a sum which lexically contains the upper bound in the summand subexpression.

In[21]:= LinearRecurrence$[\text{SUM}[\text{Binomial}[n,k],\{k,0,n\}]]$

Out[21]= $$2\,\text{SUM}(n) - \text{SUM}(n+1) == 0$$

In[22]:= $\text{LinearRecurrence}[\text{SUM}[\text{Fibonacci}[k]\,\text{Lucas}[n-k],\{k,0,n\}]]$

Out[22]= $$\text{SUM}(n)+\text{SUM}(n+1)-\text{SUM}(n+2)==F_{n+1}-2F_{n+2}$$

While LinearRecurrence transforms an expression into a defining difference equation, the command SolveLinearRecurrence does the opposite: given a linear difference equation, this command computes linearly independent solutions of this equation.

In[23]:= $rec=(\text{SUM}[n]+\text{SUM}[n+1]-\text{SUM}[n+2]==\text{Fibonacci}[n+1]-2\,\text{Fibonacci}[n+2]);$
In[24]:= $\text{SolveLinearRecurrence}[rec,\text{SUM}[n]]$

Out[24]= $Failed

In[25]:= $\text{SolveLinearRecurrence}[rec,\text{SUM}[n],\text{In}\to\{n,\text{Fibonacci}[n]\}]$

Out[25]= $$nF_n+C_1F_n+C_2F_{n+1}$$

This is a general solution of the difference equation. The SolveLinearRecurrence command is based on an implementation of Algorithm 8.1 (page 94). Recall that this algorithm might overlook solutions if algebraic dependencies of the sequences at hand are not made explicit in the underlying difference ring. For efficiency reasons, the SolveLinearRecurrence does not invoke a completion algorithm for determining algebraic dependencies, but it offers an option Using, by which algebraic dependencies can be specified. In the example above, we were lucky enough to get a solution even though the algebraic dependency between F_n and F_{n+1} was not taken into account. An attempt to express the solution in terms of Lucas numbers, however, will fail if the relation between Fibonacci numbers and Lucas numbers is not supplied.

In[26]:= $\text{SolveLinearRecurrence}[rec,\text{SUM}[n],\text{In}\to\{n,\text{Lucas}[n]\}]$

Out[26]= $Failed

In[27]:= $rels=\text{ApproximateAnnihilator}[\{\text{Fibonacci}[n],\text{Fibonacci}[n+1],\text{Lucas}[n],\text{Lucas}[n+1]\}]$

Out[27]= $$\{5F_{n+1}-2L_n-L_{n+1},5F_n+L_n-2L_n,-25+L_n^4+L_n^3L_{n+1}-L_n^2L_{n+1}^2-2L_nL_{n+1}^3+L_{n+1}^4\}$$

In[28]:= $rels=\text{Thread}[rels==0];$ $(*$ convert expressions to equations $*)$
In[29]:= $\text{SolveLinearRecurrence}[rec,\text{SUM}[n],\text{In}\to\{n,\text{Lucas}[n]\},\text{Using}\to rels]$

Out[29]= $$\tfrac{1}{5}(2nL_{n+1}-nL_n)+C_1L_{n+1}+C_2L_n$$

The commands discussed so far are used for finding new identities. They might also be applied for proving conjectured identities, but often they are too expensive for this task. There is a special purpose command called ZeroSequenceQ which provides an implementation of Algorithm 4.2. This command can be used to prove identities for admissible sequences.

In[30]:= $\text{ZeroSequenceQ}[d[n]e[n+1]-d[n+1]e[n]-(-1)^nq^{\binom{n}{2}},$

$$\text{Where} \to \{d[n+2] == d[n+1] + q^n d[n], d[0] == 1, d[1] == 0,$$
$$e[n+2] == e[n+1] + q^n e[n], e[0] == 0, e[1] == 1\}]$$

Out[30]= True

In[31]:= $\text{ZeroSequenceQ}[d[n]e[n+1] - d[n+1]e[n] + (-1)^n q^{\binom{n}{2}},$
$$\text{Where} \to \{d[n+2] == d[n+1] + q^n d[n], d[0] == 1, d[1] == 0,$$
$$e[n+2] == e[n+1] + q^n e[n], e[0] == 0, e[1] == 1\}]$$

Out[31]= False

In[32]:= $\text{ZeroSequenceQ}[-5c[n]c[n+1] + c[n-1]c[n+2] + c[n-2]c[n+3],$
$$\text{Where} \to \{c[n+2] == (c[n]^2 + c[n-1]c[n+1])/c[n-2],$$
$$c[-2] == 1, c[-1] == 1, c[0] == 1, c[1] == 1\}]$$

Out[32]= True

This command always terminates, even if the expression does not represent the zero sequence. The ZeroSequenceQ command also supports free sequences (Section 4.6). The function symbols that represent free sequences have to be declared by means of the Free option.

In[33]:= ZeroSequenceQ[
$$\text{SUM}[\frac{\text{PRODUCT}[x[i]+a, \{i,1,k-1\}]}{\text{PRODUCT}[x[i], \{i,1,k\}]}, \{k,1,n\}] - \frac{1}{a}(\text{PRODUCT}[\frac{x[k]+a}{x[k]}, \{k,1,n\}] - 1)]$$

SumCracker::general: Undefined function x encountered.

Out[33]= \$Failed

In[34]:= ZeroSequenceQ[
$$\text{SUM}[\frac{\text{PRODUCT}[x[i]+a, \{i,1,k-1\}]}{\text{PRODUCT}[x[i], \{i,1,k\}]}, \{k,1,n\}] - \frac{1}{a}(\text{PRODUCT}[\frac{x[k]+a}{x[k]}, \{k,1,n\}] - 1),$$
$\text{Free} \to \{x\}]$

Out[34]= True

There is also a multivariate version of ZeroSequenceQ, which is sometimes useful. Here, the Variable option takes a list of variables. Zero equivalence is decided recursively in this case. For instance, in the call

In[35]:= ZeroSequenceQ[
$$\text{Fibonacci}[n+m] - \tfrac{1}{2}(\text{Fibonacci}[m]\,\text{Lucas}[n] + \text{Lucas}[m]\,\text{Fibonacci}[n]),$$
$\text{Variable} \to \{n, m\}]$

Out[35]= True

the zero equivalence is shown by a two-fold induction. First, induction on n is applied. For checking the induction base $n = 1, 2$, identities in m only are obtained, which are then independently proven by induction.

The procedure of Chapter 5 for proving inequalities for admissible sequences is implemented in the command ProveInequality. Additional knowledge, which is often needed in this case, can be specified with the Using option. Also restrictions about parameters can be issued in this way.

In[36]:= $\text{ProveInequality}[(x+1)^n \geq 1 + nx, \text{Using} \to \{x \geq -1\}]$

SumCracker::general: Unable to detect dependent variable. There are several equally reasonable possibilities.

Out[36]= $Failed

In[37]:= $\text{ProveInequality}[(x+1)^n \geq 1 + nx, \text{Using} \to \{x \geq -1\}, \text{Variable} \to n]$

Out[37]= True

In[38]:= $\text{ProveInequality}[\text{SUM}[x[k]y[k], \{k, 1, n\}]^2 \leq \text{SUM}[x[k]^2, \{k, 1, n\}]\,\text{SUM}[y[k]^2, \{k, 1, n\}],$
$\text{Using} \to \{\text{SUM}[x[k]^2, \{k, 1, n\}] \geq 0, \text{SUM}[y[k]^2, \{k, 1, n\}] \geq 0\},$
$\text{Free} \to \{x, y\}]$

Out[38]= True

9.2 Summary of Expressions

We now turn to a more systematic description of the behavior of the various parts of our package. The commands of SumCracker, which are described in the following section, operate on expressions that can be recognized as representations of admissible sequences. In the present section, we describe which expressions these are. We call an expression *recognizable* in n if the SumCracker accepts it as definition of an admissible sequence. In order to facilitate the usage as much as possible, many admissible sequences with practical relevance are builtin, and the SumCracker is able to execute the closure properties described in Section 3.2. Additional sequences can be declared by explicitly stating a defining admissible system.

Expressions of the following form are recognizable in n.

(1) $f[n]$ where f is declared either in the Where option or in the Free option (cf. Section 9.4)

(2) The expressions n, a^n (a free of n), $n!$, as well as the expressions in Table 9.1 are recognizable. Every expression which is free of n is recognizable.

(3) $\text{Binomial}[an + b, cn + d]$ for $a, c \in \mathbb{Z}$ and b, d free of n is recognizable.

(4) If $\langle expr \rangle_1$ and $\langle expr \rangle_2$ are recognizable in n, then so are

$$\langle expr \rangle_1 + \langle expr \rangle_2, \quad \langle expr \rangle_1 - \langle expr \rangle_2, \quad \langle expr \rangle_1 \langle expr \rangle_2, \quad \langle expr \rangle_1 / \langle expr \rangle_2.$$

In the last case, SumCracker assumes that $\langle expr \rangle_2$ corresponds to a sequence which vanishes nowhere on the domain of definition.

For $a \in \mathbb{Z}$, $\langle expr \rangle_1^a$ is recognizable.

Expression	**Meaning**
Fibonacci$[n]$, Fibonacci$[n,x]$	Fibonacci numbers F_n and polynomials $F_n(x)$
Lucas$[n]$, Lucas$[n,x]$	Lucas numbers L_n and polynomials $L_n(x)$
Pell$[n]$, Pell$[n,x]$	Pell numbers P_n and polynomials $P_n(x)$
PellLucas$[n]$, PellLucas$[n,x]$	Pell-Lucas numbers Q_n and polynomials $Q_n(x)$
FallingFactorial$[x,n]$, RaisingFactorial$[x,n]$	falling factorial $x^{\underline{n}}$ and raising factorial $x^{\overline{n}}$
ChebyshevT$[n,x]$, ChebyshevU$[n,x]$	Chebyshev polynomials
HarmonicNumber$[n]$, HarmonicNumber$[n,r]$	harmonic number H_n and Harmonic number of order r, $\mathrm{H}_n^{(r)}$
JacobiP$[n,a,b,x]$	Jacobi polynomials $P_n^{(a,b)}(x)$
HermiteH$[n,x]$	Hermite polynomials
GegenbauerC$[n,m,x]$	Gegenbauer polynomials $C_n^m(x)$
LaguerreL$[n,a,x]$	Laguerre polynomials $L_n^a(x)$
LegendreP$[n,x]$	Legendre polynomials $P_n(x)$

Table 9.1 List of SumCracker's builtin special functions. The dependent variable is n, d and r are positive integers, and a, b, m, x have to be free of n.

(5) If $\langle expr \rangle$ is a recognizable expression in k and free of n, representing the admissible sequence $f(k)$, and if $a \in \mathbb{Z}$, then the expressions

$$\mathrm{SUM}[\langle expr \rangle, \{k,a,n\}], \quad \mathrm{PRODUCT}[\langle expr \rangle, \{k,a,n\}] \quad \text{and} \quad \mathrm{CFRAC}[\langle expr \rangle, \{k,a,n\}]$$

are recognizable. They represent the sequences $\sum_{k=a}^{n} f(k)$, $\prod_{k=a}^{n} f(k)$, and $\mathrm{K}_{k=a}^{n}(1/f(k))$, respectively.

If $\langle expr \rangle_2$ is another recognizable expression then also

$$\mathrm{CFRAC}[\langle expr \rangle_2, \langle expr \rangle, \{k,a,n\}]$$

is recognizable and it represents $\mathrm{K}_{k=a}^{n}(g(k)/f(k))$, where $g(k)$ denotes the sequence represented by $\langle expr \rangle_2$.

(6) If $\langle expr \rangle$ is a recognizable expression in n and $\langle expr \rangle'$ is obtained from $\langle expr \rangle$ by replacing each occurrence of n by Floor$[an+b]$ for some fixed $a,b \in \mathbb{Q}$, $a > 0$, then $\langle expr \rangle'$ is recognizable in n. Floor constructions cannot be nested, i.e., the present rule only applies if $\langle expr \rangle$ is free of Floor.

If $\langle expr \rangle'$ is obtained form $\langle expr \rangle$ by replacing each occurrence of n by $an+b$ where $a \in \mathbb{N}$ and b is free of n, then $\langle expr \rangle'$ is recognizable in n. If $\langle expr \rangle$ is one of the expressions in Table 9.1, then a may also be negative.

(7) $f[\langle expr \rangle]$ is recognizable in n if f is specified by a C-finite recurrence (either via the Where option, or builtin), and $\langle expr \rangle$ belongs to the closure of constants, n, exponentials a^n (a free of n) and $g[an+b]$ (g specified by a C-finite recurrence, $a,b \in \mathbb{Z}$) under addition, multiplication, indefinite summation, and exponentiation with natural number exponent. For

instance,

$$\text{Fibonacci}\left[(2n^2+5)^9\,\text{SUM}[\text{Lucas}[2k-5],\{k,1,3n\}]\right]$$

is recognizable in n due to this rule.

When carrying out a computation for some admissible sequences $f_1(n),\ldots,f_m(n)$, the ground field $\mathbb{K}$ is implicitly assumed as being the smallest field containing all expressions possibly obtained by substituting a natural number for the dependent variable n in the given expressions.

9.3 Summary of Commands

In this section, we describe the commands that the SumCracker package provides. Several options are shared by all the commands of the package. These will be explained in Section 9.4 in detail. In the present section, we only comment on options that are specific to a particular command.

ApproximateAnnihilator[⟨*list of expressions*⟩, ⟨*identifier*⟩]

Input.	⟨*list of expressions*⟩	List of recognizable expressions for admissible sequences $f_1(n),\ldots,f_m(n)$ (cf. Section 9.2)
	⟨*identifier*⟩	Symbol for representing the polynomials in the output. If that identifier is x, then $x[i]$ will be used to refer to the variable that is mapped to $f_i(n)$. If no identifier is given, the expressions themselves will be used instead of $x[i]$.
Output.		A list of polynomials in $x[1],\ldots,x[m]$.

If $f_1(n),\ldots,f_m(n)$ are the sequences declared in the ⟨*list of expressions*⟩, then this command searches for elements of $\ker\varphi$, where the ring homomorphism $\varphi\colon \mathbb{K}[x_1,\ldots,x_m]\to\mathbb{K}^{\mathbb{N}}$ is such that $x_i\mapsto f_i(n)$ and $\mathbb{K}$ is invariant.

The implementation is close to Algorithm 6.19. It provides a combination of approximation from below and above. By the Points option it can be declared how many points should be taken into account in the approximation from above. The default is 0. The number of points chosen does not influence the output, but only the runtime. Usually, 0 is the fastest choice.

A degree bound is required for the approximation from below, which is specified using the Degree option. The default setting is 10, but whether this is a good choice depends heavily on the particular example.

The command returns a list of polynomials in $x[i]$ (assuming that x is the ⟨*identifier*⟩ supplied as second argument) that generate the smallest subideal of $\ker\varphi$ which contains all polynomials of $\ker\varphi$ whose total degree does not exceed the specified degree bound.

Example.

In[1]:= $\text{ApproximateAnnihilator}[\{(-1)^n,\text{Fibonacci}[n],\text{Fibonacci}[n+1]\},t]$

Out[1]= $$\{t_1+t_2^2+t_2t_3-t_3^2,\,t_1^2-1\}$$

In[2]:= $\text{ApproximateAnnihilator}[\{\text{SUM}[\text{Fibonacci}[k],\{k,0,n\}],\text{Fibonacci}[k]^2\},t]$

Out[2]= $\{-16t_1^2-48t_1^3-68t_1^4-56t_1^5-28t_1^6-8t_1^7-t_1^8+36t_2+128t_1t_2+232t_1^2t_2+280t_1^3t_2+210t_1^4t_2+84t_1^5t_2+14t_1^6t_2-49t_2^2-204t_1t_2^2-306t_1^2t_2^2-204t_1^3t_2^2-51t_1^4t_2^2+14t_2^3+28t_1t_2^3+14t_1^2t_2^3-t_2^4\}$

Crack[$\langle$*expression*$\rangle$]

Input.	$\langle$*expression*$\rangle$	A recognizable expression for an admissible sequence $f(n)$ (cf. Section 9.2)
Output.		An equivalent expression for $f(n)$

The Crack command computes solutions of the representation problem: for some other admissible sequences $f_1(n),\dots,f_m(n)$, it searches for a rational function rat such that

$$f(n)=\mathrm{rat}(f_1(n),\dots,f_m(n))$$

on the domain of definition. The underlying algorithm is a fine tuned version of Algorithm 6.17. A degree bound is specified by the Degree option. The default value is Automatic, and causes a heuristic degree bounding based on the degrees in the defining recurrences of the difference ring. If no representation is found, the original $\langle$*expression*$\rangle$ is returned.

Which sequences are taken for $f_1(n),\dots,f_m(n)$ depends on the value of the Into option. It takes a list of recognizable expressions. By default, Into is set to Automatic. In this case, the subexpressions of $\langle$*expression*$\rangle$ are taken for the $f_i(n)$. The notion subexpression is meant here with respect to field operations and the nesting of indefinite sum, product, and continued fraction quantifiers. For instance, $\sum_{k=1}^n k$ and n are subexpressions of $\sum_{k=1}^n\sum_{i=1}^k i$, but n is not considered as a subexpression of F_n.

By saying, for instance, Into $\to$ {Fibonacci[n]}, also the higher shift Fibonacci[$n+1$] is implicitly specified as a possible subexpression of right hand side. In order to avoid this, the option IncludingShifts has to be set to False.

Finally, it is also possible to restrict the search to polynomial solutions only. For this, set the option Denominator to False.

Examples.

In[1]:= $\mathrm{Crack}[\mathrm{SUM}[\frac{\mathrm{Fibonacci}[k]}{\mathrm{Fibonacci}[k+1]\,\mathrm{Fibonacci}[k+2]},\{k,1,n\}]]$

Out[1]= $$\frac{F_n+F_{n+1}-1}{F_n+F_{n+1}}$$

In[2]:= $\mathrm{Crack}[\mathrm{SUM}[\frac{\mathrm{Fibonacci}[k]}{\mathrm{Fibonacci}[k+1]\,\mathrm{Fibonacci}[k+2]},\{k,1,n\}],\mathrm{Into}\to\{\mathrm{Lucas}[n]\}]$

Out[2]= $$\frac{3L_{n+1}+L_n-5}{3L_{n+1}+L_n}$$

In[3]:= $\mathrm{Crack}[\mathrm{SUM}[\frac{\mathrm{Fibonacci}[k]}{\mathrm{Fibonacci}[k+1]\,\mathrm{Fibonacci}[k+2]},\{k,1,n\}],\mathrm{Into}\to\{\mathrm{Lucas}[n]\},$
$\mathrm{IncludingShifts}\to\mathrm{False}]$

Out[3]= $$\sum_{k=1}^{n} \frac{F_k}{F_{k+1}F_{k+2}}$$

In[4]:= $\text{Crack}[\text{SUM}[\frac{\text{Fibonacci}[k]}{\text{Fibonacci}[k+1]\,\text{Fibonacci}[k+2]}, \{k,1,n\}], \text{Into} \to \{\text{Lucas}[n]\},$
$\text{Denominator} \to \text{False}]$

Out[4]= $$\sum_{k=1}^{n} \frac{F_k}{F_{k+1}F_{k+2}}$$

LinearRecurrence[$\langle$*expression*$\rangle$]

Input.	$\langle$*expression*$\rangle$	A recognizable expression for an admissible sequence $f(n)$ (cf. Section 9.2), or an definite sum expression of the form $\text{SUM}[\langle summand\rangle, \{k,a,bn\}]$ with $a \in \mathbb{Z}$ and $b \in \mathbb{N}$, where the expression $\langle$*summand*$\rangle$ syntactically contains n. The $\langle$*summand*$\rangle$ expression must be recognizable in both n and k.
Output.		A linear difference equation which has $f(n)$ as a solution, or $Failed if no such equation is found.

In the resulting difference equation, the function is represented by SUM[n]. The symbol SUM can be replaced by any other symbol using the Head option. The order of the desired recurrence is specified by the option Order which accepts a nonnegative integer as value. The default value is Infinity, which causes the finder to loop over the orders $0,1,2,3,\ldots$ until a recurrence is found.

The details of command and option are slightly different, depending on whether $\langle$*expression*$\rangle$ is a definite sum or not.

First consider the case where $\langle$*expression*$\rangle$ is not a definite sum. This case algorithmically works similar to Crack, and it is guaranteed that no equation escapes from the search.

The Degree option can be used for specifying the total degree of the expressions that may appear as coefficients of the recurrence. The default setting is Automatic and treated as in the Crack command (see above). A more refined specification of degree bounds is possible as well. If a list $\{d_{-1}, d_0, \ldots, d_\ell\}$ is given as degree bound, where each d_i is either a nonnegative integer or the symbol Automatic, then d_{-1} will be used as degree bound for the inhomogeneous part and $d_{\max(i,\ell)}$ will be used as degree bound for the coefficient of order i.

Analogous remarks as for the degree bounding hold for the choice of expressions that may appear in the coefficients. The default choice (Automatic) can be overruled using the option In that points to a list of expressions, or to a list $\{in_{-1}, in_0, \ldots, in_\ell\}$ where each in_i is an expression list or the symbol Automatic, and the semantics is analogous to the degree bound specification.

The option IncludingShifts has the same meaning as in the Crack command (see above).

If the $\langle$*expression*$\rangle$ is a definite sum, then LinearRecurrence executes creative telescoping. In this case, the degree bound and coefficient expressions refer to the telescoping inhomogeneous part of the recurrence on the summand level (cf. Section 8.4). Degree and expressions appearing in the recurrence for the sum, as delivered by the command, cannot be influenced by options.

There is currently no way to specify bivariate admissible sequences $f[n,k]$ with the Where option. Understandable bivariate sequences are the binomial coefficient, expressions of the form $f[an+bk]$ with $a,b \in \mathbb{N}_0$ and f either a user defined are a built-in function, and rational functions thereof. For builtin functions such as F_n, the coefficients a and b may also be negative.

Examples.

In[1]:= $\text{LinearRecurrence}[\text{Fibonacci}[n]]$

Out[1]= $$\text{SUM}(n)+\text{SUM}(n+1)-\text{SUM}(n+2)==0$$

In[2]:= $\text{LinearRecurrence}[\text{Fibonacci}[n],\text{Head}\to f]$

Out[2]= $$f(n)+f(n+1)-f(n+2)==0$$

In[3]:= $\text{LinearRecurrence}[\text{SUM}[\text{Fibonacci}[n+k],\{k,0,n\}]]$

Out[3]= $$-\text{SUM}(n)+\text{SUM}(n+1)==-F_n+F_{2n+1}+F_{2n+2}$$

ProveInequality[⟨*formula*⟩]

Input.	⟨*formula*⟩	A boolean combination of inequalities on both sides of which there are recognizable expressions.
Output.		True or False

This command is an implementation of the procedure described in Chapter 5. Here, the knowledge specified via the Using option may also include inequalities that are known to be true.

ProveInequality also allows free sequences to appear in ⟨*formula*⟩, whose heads have to be declared via the Free option.

If the procedure encounters a point where the formula does not hold, it returns False. However, if the formula to be proven involves parameters or special functions, it might not be possible to distinguish a point violating the formula and a point at which the validity of the formula could not be determined. In such a situation, an exception is thrown and the command terminates with the return value $Failed.

Examples.

In[1]:= $\text{ProveInequality}[\text{SUM}[\frac{\text{Lucas}[k]^2}{\text{Fibonacci}[k]},\{k,1,n\}]\geq\frac{(\text{Lucas}[n+2]-3)^2}{\text{Fibonacci}[n+2]-1}]$

Out[1]= False

In[2]:= $\text{ProveInequality}[\text{SUM}[\frac{\text{Lucas}[k]^2}{\text{Fibonacci}[k]},\{k,1,n\}]\geq\frac{(\text{Lucas}[n+2]-3)^2}{\text{Fibonacci}[n+2]-1},\text{From}\to 2]$

Out[2]= $Aborted

In[3]:= $\text{ProveInequality}[\text{SUM}[\frac{\text{Lucas}[k]^2}{\text{Fibonacci}[k]}, \{k, 1, n\}] \geq \frac{(\text{Lucas}[n+2]-3)^2}{\text{Fibonacci}[n+2]-1}, \text{From} \to 2,$
$\text{Using} \to \{\text{Fibonacci}[n] \geq 1\}]$

Out[3]= True

In[4]:= $\text{ProveInequality}[\text{Fibonacci}[n] \geq 1, \text{From} \to 2]$

Out[4]= True

SolveLinearRecurrence[$\langle$*equation*$\rangle$, $f[n]$]

Input.	$\langle$*equation*$\rangle$	A linear difference equation, possibly inhomogeneous, in $f[n]$ whose coefficients are recognizable expressions in n
	$f[n]$	Function symbol and dependent variable of the equation
Output.		Sum of a particular solution (possibly zero) and linearly independent solutions of the associated homogeneous equation, using $C[1], C[2], \ldots$ as coefficients

This command is an implementation of Algorithm 8.1.

Via the options In and Degree, it can be specified which terms may appear in the solution and what the total degree of the solution might be. Both options by default have the value Automatic.

An early termination criterion is applied when a solution is found. If the equation is homogeneous, the command returns a $\mathbb{K}$-basis for all linearly independent solutions of the smallest total degree for which nontrivial solutions exist. In case of an inhomogeneous equation, the command loops up the degree until the first inhomogeneous solution is found. If this happens at degree d, it returns this inhomogeneous solution together with a basis of solutions of the homogeneous part whose degree does not exceed d. The early termination can be switched off by setting the option EarlyTermination to False.

Note that the difference ring in which the difference equation is solved might not be a faithful representation of the corresponding admissible sequences. As a consequence, solutions for the actual equation (with sequences in place of difference polynomials) might be missing, or there might be additional solutions given, which are linearly independent as difference ring elements but not as sequences. To exclude such phenomena, by the Using option one can specify a list of relations that generates the annihilating ideal of the involved quantities, or a subideal thereof. (See ApproximateAnnihilator above for computing such generators.) If the ideal is complete, the underlying difference ring is isomorphic to a subring of the ring of sequences over $\mathbb{K}$, and it is guaranteed that all solutions are found and that linear independence in the difference ring agrees with linear independence over the sequences.

Examples.

In[1]:= $\text{SolveLinearRecurrence}[f[n+1] == f[n], f[n]]$

Out[1]= C_1

In[2]:= SolveLinearRecurrence$[f[n+1] == f[n], f[n], \text{In} \to \{\text{Fibonacci}[n]\}]$

Out[2]= C_1

In[3]:= SolveLinearRecurrence$[f[n+1] == f[n], f[n], \text{In} \to \{\text{Fibonacci}[n]\},$
EarlyTermination $\to$ False$]$

Out[3]=
$$C_1 + C_2(F_n^4 + 2F_n^3F_{n+1} - F_n^2F_{n+1}^2 - 2F_nF_{n+1}^3 + F_{n+1}^4)$$

In[4]:= SolveLinearRecurrence$[f[n+1] == f[n], f[n], \text{In} \to \{\text{Fibonacci}[n]\},$
EarlyTermination $\to$ False, Degree $\to 8]$

Out[4]=
$$C_1 + C_2(F_n^4 + 2F_n^3F_{n+1} - F_n^2F_{n+1}^2 - 2F_nF_{n+1}^3 + F_{n+1}^4) + C_3(F_n^8 + 4F_n^7F_{n+1} + 2F_n^6F_{n+1}^2 - 8F_n^5F_{n+1}^3 - 5F_n^4F_{n+1}^4 + 8F_n^3F_{n+1}^5 + 2F_n^2F_{n+1}^6 - 4F_nF_{n+1}^7 + F_{n+1}^8)$$

In[5]:= SolveLinearRecurrence$[f[n+1] == f[n], f[n], \text{In} \to \{\text{Fibonacci}[n]\},$
EarlyTermination $\to$ False, Degree $\to 20$,
Using $\to \{\text{Fibonacci}[n]^4 + 2\,\text{Fibonacci}[n]^3\,\text{Fibonacci}[n+1]$
$-\,\text{Fibonacci}[n]^2\,\text{Fibonacci}[n+1]^2$
$-\,2\,\text{Fibonacci}[n]\,\text{Fibonacci}[n+1]^3 + \text{Fibonacci}[n+1]^4 == 1\}]$

Out[5]= C_1

ZeroSequenceQ[⟨*expression*⟩]

Input.	⟨*expression*⟩	A recognizable expression for an admissible sequence $f(n)$
Output.		True or False

This command contains an implementation of Algorithm 4.2. It returns True if $f(n) = 0$ for all n, and False otherwise. This command always terminates, though the computation time might be long on large examples.

ZeroSequenceQ also allows free sequences to appear in ⟨*expression*⟩, whose heads have to be declared by the Free option.

In Algorithm 4.2, deciding zero equivalence of the admissible sequence $f(n)$ is reduced to deciding zero equivalence for some initial values $f(1), f(2), \ldots$. By default, a ground field expression is considered to be zero if it is—after some simplification—lexicographically identical to 0. In nontrivial ground fields, it might happen that ZeroSequenceQ mistakenly returns False because zero equivalence of some initial value could not be established. In such situations, the user should specify which function should be used to determine zero equivalence in the ground field. Such a function can be supplied via the ZeroQ option.

There is also a multivariate extension. If the Variable option points to a list of variables, then Algorithm 4.2 is applied with respect to the first variable of that list. Each initial condition is an expression in the remaining variables, and zero equivalence of these expressions is decided by a recursive application of the algorithm.

Examples.

In[1]:= $\text{ZeroSequenceQ}[\text{SUM}[\frac{(-1)^k}{\text{Fibonacci}[k]\,\text{Fibonacci}[k+1]},\{k,1,n\}]+\frac{\text{Fibonacci}[n]}{\text{Fibonacci}[n+1]}]$

Out[1]= True

With function symbols for free sequences and the Using option, it is possible to prove identities like

$$\sum_{k=0}^{n}\frac{1}{\sqrt{k+1}+\sqrt{k}}=\sqrt{n+1}\quad(n\geq 0).$$

A special function has to be employed for testing zero equivalence in the ground field $\mathbb{Q}\langle\sqrt{n}\rangle$.

In[2]:= $\text{ZeroSequenceQ}[\text{SUM}[\frac{1}{x[k+1]+x[k]},\{k,0,n\}]-x[n+1],\text{Free}\to\{x\},$
$\text{Using}\to\{x[n]^2==n\}]$

Out[2]= False

In[3]:= $\text{ZeroSequenceQ}[\text{SUM}[\frac{1}{x[k+1]+x[k]},\{k,0,n\}]-x[n+1],\text{Free}\to\{x\},$
$\text{Using}\to\{x[n]^2==n\},\text{ZeroQ}\to(\text{SameQ}[0,\text{Expand}[\#\,/.\,(x[i\_]\to\text{Sqrt}[i])]]\&)]$

Out[3]= True

9.4 Summary of Options

Several options are shared by all commands of the SumCracker package. Some options like Degree and In/Into differ in syntax and/or semantics from one command to another. These options are explained in Section 9.3 above. In the present section, we describe the options which are identical for all commands. Most of these options concern the specification of additional knowledge for the expressions that are used to define admissible sequences.

Free → ⟨*list of symbols*⟩ (default: {})

List of function symbols that represent free sequences. In the current version of SumCracker, free sequences are only supported by ProveInequality and ZeroSequenceQ, but it seems possible to extend also the other commands such as to allow free sequences.

Function symbols without builtin meaning have to appear either in this list or in the admissible system specified with the Where option (see below), otherwise an error is generated.

From → ⟨*integer*⟩ or Automatic (default: Automatic)

First point from which on the sequences are defined. If set to automatic, SumCracker determines a startpoint from the lower indices of summation, product, and continued fraction quantifiers, and from initial value specifications appearing in the admissible system specified with the Where option (see below). If this does not lead to a result, the default start point 0 is taken.

Infolevel → ⟨*integer*⟩ or Infinity (default: 0)

Amount of information that should be printed during a computation. If set to 0, no information will be printed at all. If set to 1 (2, 3, ...), a one-line description will be printed for every major step (substep, subsubstep, ...) of the computation. A sample output is given on page 110. The finest level of information is obtained by setting Infolevel to Infinity.

Using → ⟨*list of equations*⟩ (default: {})

Knowledge base. SumCracker starts every computation by building a difference ring that represents the admissible sequences which are specified by the expressions of the input. The difference polynomials that correspond to the identities in the knowledge base are taken as generators of the annihilating ideal of these sequences. If the knowledge base contains an identity which does not hold on the domain of definition (see From, above), no command will produce a reliable result.

If the ideal generated by the identities in the knowledge base is not complete, ApproximateAnnihilator and Crack may run slowlier, but their output is not affected. Other commands may overlook solutions.

For the command ProveInequality, also inequalities can be given via the Using option.

Variable → ⟨*symbol*⟩ (default: Automatic)

Dependent variable. This is the variable in which the recognizable expressions are given, typically n or k. If set to Automatic, SumCracker tries to guess heuristically from the input expressions what the dependent variables might be. This is successful in most cases, but not always. When there are doubts about the choice, an error is generated and the computation is aborted. Only in rare cases, in particular if nested C-finite expressions occur, the computation is continued with a wrong guess. If Infolevel is set to 3 or higher, the chosen variable will be shown.

For the command ZeroSequenceQ, also a list of variables can be declared, see page 121.

Where → ⟨*list of equations*⟩ (default: {})

Admissible system for user-defined sequences. The ⟨*list of equations*⟩ contains recurrences and initial value specifications. The recurrences have to be as in Def. 3.1, though on the right hand side, also recognizable expressions and free sequences may be used. The ordering in which the recurrences are stated matters. Initial value specifications are of the form $f[i] == v$ for integers i. They can appear at any position in the list. Missing initial values are padded with symbolic values.

Function symbols without builtin meaning have to appear either in in the Free list (see above) or in the admissible system specified by the Where option, otherwise an error is generated.

9.5 Example Session

Consider once more Exercise 6.61 of Graham *et al.* (1994): *Prove the identity*

$$\sum_{k=0}^{n} \frac{1}{F_{2^k}} = 3 - \frac{F_{2^n-1}}{F_{2^n}}, \qquad \textit{integers } n \geq 1.$$

What is $\sum_{k=0}^{n} 1/F_{3\cdot 2^k}$*?*

Let us first solve the exercise as it stands.

In[1]:= $\text{ZeroSequenceQ}[\text{SUM}[\frac{1}{\text{Fibonacci}[2^k]}, \{k,0,n\}] - \Big(3 - \frac{\text{Fibonacci}[2^n-1]}{\text{Fibonacci}[2^n]}\Big), \text{From} \to 1]$

Out[1]= True

In[2]:= $\text{Crack}[\text{SUM}[\frac{1}{\text{Fibonacci}[3\cdot 2^k]}, \{k,0,n\}], \text{From} \to 1]$

Out[2]= $\frac{9}{4} - \frac{F_{3\cdot 2^n+1}}{F_{3\cdot 2^n}}$

This is similar to the right hand side of the original identity. For perfect consistency, we can force the negative shift in the numerator on the right hand side, for instance as follows.

In[3]:= $\text{Crack}[\text{SUM}[\frac{1}{\text{Fibonacci}[3\cdot 2^k]}, \{k,0,n\}] + \frac{\text{Fibonacci}[3\cdot 2^n-1]}{\text{Fibonacci}[3\cdot 2^n]}, \text{From} \to 1]$

Out[3]= $\frac{5}{4}$

Is there perhaps a more general identity of the form

$$\sum_{k=0}^{n} \frac{1}{F_{a2^k}} = c(a) - \frac{F_{a2^n-1}}{F_{a2^n}} \qquad (a, n \geq 1) \tag{9.1}$$

for some numbers $c(a)$ which are independent of n? Applying Crack as above for the particular values $a = 1, 2, 3, \ldots, 20$ indeed always gives a closed form that matches the right hand side of (9.1), with the following values of $c(a)$:

a	1	2	3	4	5	6	7	8	9	10
$c(a)$	3	2	$\frac{5}{4}$	1	$\frac{46}{55}$	$\frac{3}{4}$	$\frac{263}{377}$	$\frac{2}{3}$	$\frac{837}{1292}$	$\frac{7}{11}$

a	11	12	13	14	15	16	17	18	19	20
$c(a)$	$\frac{11146}{17711}$	$\frac{5}{8}$	$\frac{75547}{121393}$	$\frac{18}{29}$	$\frac{257797}{416020}$	$\frac{13}{21}$	$\frac{3528150}{5702887}$	$\frac{47}{76}$	$\frac{24167167}{39088169}$	$\frac{34}{55}$

Is there a closed form for these numbers? Let us apply Crack to the general sum.

In[4]:= $\text{Crack}[\text{SUM}[1/\text{Fibonacci}[a\cdot 2^k], \{k,0,n\}]]$

Out[4]=
$$\sum_{k=0}^{n} \frac{1}{F_{a\cdot 2^k}}$$

No result—we were too optimistic. We have to have a closer look at the numbers in the table. By looking close enough, we detect that $F_{10} = 55$ is the denominator of $c(5)$, $F_{14} = 377$ is the denominator of $c(7)$, and $F_{22} = 17711$ is the denominator of $c(11)$. In general, the denominator of $c(a)$ divides F_{2a}, so that $F_{2a}c(a)$ is an integer for small a. Let us apply the guessing function of Mallinger's package (Mallinger, 1996) to this sequence.

In[5]:= $c = \text{Table}[\text{Crack}[\text{SUM}[1/\text{Fibonacci}[a2^k], \{k,0,n\}] + \text{Fibonacci}[a2^n - 1]/\text{Fibonacci}[a2^n],$
$\text{From} \to 1], \{a,1,20\}];$

In[6]:= << GeneratingFunctions.m

GeneratingFunctions Package by Christian Mallinger – © RISC Linz – V 0.68 (07/17/03)

In[7]:= $c = c * \text{Table}[\text{Fibonacci}[2a], \{a,1,20\}];$ $\quad$ (* clear denominators *)

In[8]:= $\text{GuessRE}[c, num[a], 10, 0]$

Out[8]=
$$\{\{num(a) - 3num(a+1) - num(a+2) + 7num(a+3) - 5num(a+4) + num(a+5) == 0,$$
$$num(0) == 3, num(1) == 6, num(2) == 10, num(3) == 21, num(4) == 46\}, \text{ogf}\}$$

Next, we can use SumCracker for expressing the numerator sequence in terms of Fibonacci numbers. Note that Mallinger's package assumes that the sequence starts with $a = 0$ while $c(a)$ is defined for $a \geq 1$. We have to shift the initial values by 1.

In[9]:= $\text{Crack}[num[a],$

$\text{Where} \to \{num[a+5] == 5num[a+4] - 7num[a+3] + num[a+2] + 3num[a+1] - num[a],$

$num[1] == 3, num[2] == 6, num[3] == 10, num[4] == 21, num[5] == 46\}, \text{Into} \to \{\text{Fibonacci}[a]\}]$

Out[9]=
$$1 - F_a + 2F_a^2 + 2F_{a+1} - 2F_aF_{a+1} + F_{a+1}^2$$

Summarizing, we now have the conjecture

$$c(a) = \frac{F_{a+1}^2 - 2F_aF_{a+1} + 2F_a^2 + 2F_{a+1} - F_a + 1}{F_{2a}} \qquad (a \geq 1).$$

We will now prove this conjecture, also using facilities provided by our package. Before doing so, it is wise to cross-check the conjecture by evaluating it at small points.

In[10]:= $c[a\_] = (1 - \text{Fibonacci}[a] + 2\,\text{Fibonacci}[a]^2 + 2\,\text{Fibonacci}[a+1]$
$-2\,\text{Fibonacci}[a]\,\text{Fibonacci}[a+1] + \text{Fibonacci}[a+1]^2)/\text{Fibonacci}[2a];$

In[11]:= And @@ Flatten[Table[
$\text{SUM}[1/\text{Fibonacci}[a\cdot 2^k], \{k,0,n\}] == c[a] - \text{Fibonacci}[a\cdot 2^n - 1]/\text{Fibonacci}[a\cdot 2^n],$
$\{a,1,100\}, \{n,1,10\}]]$

Out[11]= True

This looks convincing. Now we turn to proving identity (9.1) for the $c(a)$ as above. We apply a nested induction along n (outer induction) and a (inner induction). For this computation, special purpose software was necessary to carry out the underlying Gröbner basis computations.

Theorem 9.2 For all $a, n \in \mathbb{N}$, we have

$$\sum_{k=0}^{n} \frac{1}{F_{a\cdot 2^k}} = \frac{F_{a+1}^2 - 2F_aF_{a+1} + 2F_a^2 + 2F_{a+1} - F_a + 1}{F_{2a}} - \frac{F_{a\cdot 2^n - 1}}{F_{a\cdot 2^n}}.$$

Proof We rely on the SumCracker.

In[12]:= ZeroSequenceQ[
SUM[1/Fibonacci[$a \cdot 2^k$], {k, 0, n}] + Fibonacci[$a \cdot 2^n - 1$]/Fibonacci[$a \cdot 2^n$] − c[a],
Variable → {n, a}, From → 1]

Out[12]= True ∎

In[13]:= Quit

References

Abramov, S. and Petkovšek, M. (2005). Gosper's algorithm, accurate summation, and the discrete Newton-Leibniz formula. In *Proceedings of ISSAC'05*, pages 5–12.

Abramov, S., Carette, J., Geddes, K., and Le, H. (2004). Telescoping in the context of symbolic summation in Maple. *Journal of Symbolic Computation*, **38**(4), 1303–1326.

Abramowitz, M. and Stegun, I. A. (1972). *Handbook of Mathematical Functions*. Dover Publications, Inc., 9 edition.

Aho, A. V. and Sloane, N. J. A. (1973). Some doubly exponential sequences. *The Fibonacci Quarterly*, **11**, 429–437.

Andrews, G. E., Askey, R., and Roy, R. (1999). *Special Functions*, volume 71 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press.

Andrews, G. E., Knopfmacher, A., and Paule, P. (2000). An infinite family of Engel expansions of Rogers-Ramanujan type. *Advances in Applied Mathematics*, **25**, 2–11.

Arnold, E. A. (2003). Modular algorithms for computing Gröbner bases. *Journal of Symbolic Computation*, **35**, 403–419.

Askey, R. and Gasper, G. (1976). Positive Jacobi polynomial sums II. *American Journal of Mathematics*, **98**, 709–737.

Becker, T., Weispfenning, V., and Kredel, H. (1993). *Gröbner Bases*. Springer.

Beesack, P. R. (1969). On certain discrete inequalities involving partial sums. *Canadian Journal of Mathematics*, **21**, 222–234.

Borwein, J. M. and Lisoněk, P. (2000). Applications of integer relation algorithms. *Discrete Mathematics*, **23**, 65–82.

Bourbaki, N. (1970). *Algebra I*. Springer.

Bousquet-Mélou, M. and Petkovšek, M. (2000). Linear recurrences with constant coefficients: the multivariate case. *Discrete Mathematics*, **225**, 51–75.

Bowman, D. and Laughlin, J. M. (2002). Polynomial continued fractions. *Acta Arithmetica*, **103**(4), 329–342.

Brent, R. P. (1976). Fast multiple-precision evaluation of elementary functions. *Journal of the ACM*, **23**, 242–251.

Bronstein, M. (1997). *Symbolic Integration I*. Number 1 in Algorithms and Computation in Mathematics. Springer.

Brown, C. W. (2003a). QEPCAD B – a program for computing with semi-algebraic sets. *Sigsam Bulletin*, **37**(4), 97–108.

Brown, S. H. (2003b). Problem B-952. *The Fibonacci Quarterly*, **41**(1), 85.

Bruckman, P. S. (2002a). Problem B-937. *The Fibonacci Quartely*, **40**(2), 181.

Bruckman, P. S. (2002b). Problem H-584. *The Fibonacci Quarterly*, **40**(2), 187.

Buchberger, B. (1965). *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal*. Ph.D. thesis, Universität Innsbruck.

Cai, J.-Y., Lipton, R. J., and Zalcstein, Y. (2000). The complexity of the *a b c* problem. *Siam Journal of Computation*, **29**(6), 1878–1888.

Calkin, N. J. (1994). A curious binomial identity. *Discrete Mathematics*, **131**(1–3), 335–337.

Caviness, B. and Johnson, J. R., editors (1998). *Quantifier Elimination and Cylindrical Algebraic Decomposition*, Texts and Monographs in Symbolic Computation. Springer.

Chihara, T. S. (1978). *An Introduction to Orthogonal Polynomials*, volume 13 of *Mathematics and its Applications*. Gordon and Breach Science Publishers.

Chrystal, G. (1922). *Algebra — an Elementary Textbook, Part II*. Cambridge University Press, 2nd edition.

Chyzak, F. (1998). *Fonctions holonomes en calcul formel*. Ph.D. thesis, INRIA Rocquencourt.

Chyzak, F. (2000). An extension of Zeilberger's fast algorithm to general holonomic functions. *Discrete Mathematics*, **217**, 115–134.

Chyzak, F. and Salvy, B. (1998). Non-commutative elimination in Ore algebras proves multivariate identities. *Journal of Symbolic Computation*, **26**, 187–227.

Cohen, H. (1993). *A Course in Computational Algebraic Number Theory*. Springer.

Cohn, R. M. (1965). *Difference Algebra*. Interscience Publishers, John Wiley & Sons.

Collins, G. E. (1975). Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition. *Lecture Notes in Computer Science*, **33**, 134–183.

Collins, G. E. and Hong, H. (1991). Partial cylindrical algebraic decomposition for quantifier elimination. *Journal of Symbolic Computation*, **12**(3), 299–328.

Cox, D., Little, J., and O'Shea, D. (1992). *Ideals, Varieties, and Algorithms*. Springer.

de Branges, L. (1985). A proof of the Bieberbach conjecture. *Acta Mathematica*, **154**, 137–152.

Derksen, H., Jeandel, E., and Koiran, P. (2005). Quantum automata and algebraic groups. *Journal of Symbolic Computation*, **39**(3–4), 357–371.

Diaz, J. L. (2001). Problem B-917. *The Fibonacci Quarterly*, **39**(2).

Diaz, J. L. and Egozcue, J. J. (2002a). Problem B-936. *The Fibonacci Quarterly*, **40**(2), 181.

Diaz, J. L. and Egozcue, J. J. (2002b). Problem B-943. *The Fibonacci Quarterly*, **40**(4), 372.

Dijkstra, E. W. (1978). In honour of Fibonacci. In *Program Construction*, volume 69 of *Lecture Notes in Computer Science*, pages 49–50. Springer.

Eisenbud, D. (1995). *Commutative Algebra*. Springer.

Ekhad, S. B. (1993). A short, elementary, and easy, WZ proof of the Askey-Gasper inequality that was used by de Branges in his proof of the Bieberbach conjecture. *Theoretical Computer Science*, **117**, 199–202.

Everest, G., van der Poorten, A., Shparlinski, I., and Ward, T. (2003). *Recurrence Sequences*, volume 104 of *Mathematical Surveys and Monographs*. American Mathematical Society.

Faugère, J.-C. (1999). A new efficient algorithm for computing Gröbner bases. *Journal of Pure and Applied Algebra*, **139**(1–3), 61–88.

Faugère, J.-C. (2002a). Gb tutorial. http://www-calfor.lip6.fr/˜jcf/.

Faugère, J.-C. (2002b). A new efficient algorithm for computing Gröbner bases without reduction to zero. In *Proceedings of ISSAC '02*.

Filipponi, P. (1996). On the Fibonacci numbers whose subscript is a power. *The Fibonacci Quarterly*, **34**(3), 271–276.

Friendman, J. (1995). Problem B-785. *The Fibonacci Quarterly*, **33**(2).

Furdui, O. (2002). Problem B-931. *The Fibonacci Quarterly*, **40**(1), 85.

Gale, D. (1991). The strange and surprising saga of the Somos sequences. *Mathematical Intelligencer*, **13**(1), 40–42.

Ge, G. (1993). *Algorithms related to multiplicative representations of algebraic numbers*. Ph.D. thesis, U.C. Berkeley.

Geddes, K. O., Czapor, S. R., and Labahn, G. (1992). *Algorithms for Computer Algebra*. Kluwer.

Gerhold, S. (2002). *Uncoupling Systems of Linear Ore Operator Equations*. Master's thesis, RISC-Linz.

Gerhold, S. (2004). On some non-holonomic sequences. *Electronic Journal of Combinatorics*, **11**(1), 1–8.

Gerhold, S. and Kauers, M. (2005). A procedure for proving special function inequalities involving a discrete parameter. In *Proceedings of ISSAC'05*, pages 156–162.

Golomb, S. W. (1963). On certain nonlinear recurring sequences. *The American Mathematical Monthly*, **70**, 403–405.

Gosper, W. (1978). Decision procedure for indefinite hypergeometric summation. *Proceedings of the National Academy of Sciences of the United States of America*, **75**, 40–42.

Graham, R. L., Knuth, D. E., and Patashnik, O. (1994). *Concrete Mathematics*. Addison-Wesley, second edition.

Hardy, G., Littlewood, J. E., and Pólya, G. (1952). *Inequalities*. Cambridge Mathematical Library. Cambridge University Press, second edition.

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

Hoggatt, V. E. (1979). *Fibonacci and Lucas numbers*. The Fibonacci Association.

Janous, W. (2002). Problem B-941. *The Fibonacci Quarterly*, **40**(4), 372.

Kaplansky, I. (1976). *An Introduction to Differential Algebra*. Hermann, 2nd edition.

Karr, M. (1981). Summation in finite terms. *Journal of the ACM*, **28**, 305–350.

Karr, M. (1985). Theory of summation in finite terms. *Journal of Symbolic Computation*, **1**(3), 303–315.

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

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

Kauers, M. and Schneider, C. (2004). Indefinite summation with unspecified sequences. Technical Report 2004-13, SFB F013, Johannes Kepler Universität. (submitted).

Kauers, M. and Zimmermann, B. (2005a). Computing the algebraic dependencies of recurrent sequences. Technical report, SFB F013, Johannes Kepler Universität. in preparation.

Kauers, M. and Zimmermann, B. (2005b). Dependencies — a Mathematica package for computing algebraic dependencies of C-finite sequences. Technical report, SFB F013, Johannes Kepler Universität. in preparation.

Khinchin, A. Y. (1964). *Continued Fractions*. University of Chicago Press.

Kreuzer, M. and Robbiano, L. (2000). *Computational Commutative Algebra I*. Springer.

Lenstra, A. K., Jr., H. W. L., and Lovász, L. (1982). Factoring polynomials with rational coefficients. *Annals of Mathematics*, **126**, 515–534.

Mallinger, C. (1996). *Algorithmic Manipulations and Transformations of Univariate Holonomic Functions and Sequences*. Master's thesis, J. Kepler University, Linz.

Masser, D. W. (1988). Linear relations on algebraic groups. In A. Baker, editor, *Proc. of New Advances in Transcendence Theory*, pages 248–262.

Mayr, E. W. (1997). Some complexity results for polynomial ideals. *Journal of Complexity*, **13**(3), 301–384.

Milne-Thomson, L. (1933). *The Calculus of Finite Differences*. Macmillan and Co., ltd.

Mitrinović, D. S. (1964). *Elementary Inequalities*. P. Noordhoff Ltd.

Mitrinović, D. S. (1970). *Analytic Inequalities*. Springer.

Nanjundiah, T. (1993). Problem 10347. *The American Mathematical Monthly*, **100**(10).

Nemes, I. and Petkovšek, M. (1995). Rcomp: A mathematica package for computing with recursive sequences. *Journal of Symbolic Computation*, **20**(5–6), 745–753.

Odlyzko, A. (1995). Asymptotic enumeration methods. *Handbook of Combinatorics*, **volume II**.

Paule, P. (2005). A computerized proof of $\zeta(2) = \pi^2/6$. *In preparation*.

Paule, P. and Schorn, M. (1995). A Mathematica version of Zeilberger's algorithm for proving binomial coefficient identities. *Journal of Symbolic Computation*, **20**(5–6), 673–698.

Pemantle, R. and Schneider, C. (2004). When is $0.999\ldots$ equal to 1? Technical Report 2004-30, SFB F013, Johannes Kepler Universität.

Perron, O. (1929). *Die Lehre von den Kettenbrüchen*. Chelsea Publishing Company, second edition.

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

Petkovšek, M., Wilf, H., and Zeilberger, D. (1997). $A = B$. AK Peters, Ltd.

Rabinowitz, S. (2003). Problem B-951. *The Fibonacci Quarterly*, **40**(1), 84.

Risch, R. (1969). The problem of integration in finite terms. *Transactions of the American Mathematical Society*, **139**, 167–189.

Risch, R. (1970). The solution of problem of integration in finite terms. *Bulletin of the American Mathematical Society*, **79**, 605–608.

Ritt, J. F. (1950). *Differential Algebra*. American Mathematical Society, Colloquium Publications.

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

Schneider, C. (2001). *Symbolic Summation in Difference Fields*. Ph.D. thesis, RISC-Linz, Johannes Kepler Universität Linz.

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

Schorn, M. (1995). *Contributions to Symbolic Summation*. Master's thesis, RISC-Linz, Johannes-Kepler-Universität.

Schur, I. and Knopp, K. (1918). Über die Herleitung der Gleichung $\sum_{n=1}^{\infty} 1/n^2 = \pi^2/6$. *Archiv der Mathematik und Physik*, **3**(27), 174–176.

Seiffert, H.-J. (1994a). Problem B-757. *The Fibonacci Quarterly*, **32**(1).

Seiffert, H.-J. (1994b). Problem B-759. *The Fibonacci Quarterly*, **32**(1).

Shackell, J. (1993). Zero-equivalence in function fields defined by algebraic differential equations. *Transactions of the American Mathematical Society*, **336**(1), 151–171.

Shackell, J. (2004). *Symbolic Asymptotics*. Number 12 in Algorithms and Computation in Mathematics. Springer.

Shafarevich, I. R. (1972). *Basic Algebraic Geometry*. Springer.

Somos, M. (1989). Problem 1470. *Crux Mathematicorum*, **15**, 208.

Stanley, R. P. (1997). *Enumerative Combinatorics, Volume 1*. Cambridge Studies in Advanced Mathematics 62. Cambridge University Press.

Stanley, R. P. (1999). *Enumerative Combinatorics, Volume 2*. Cambridge Studies in Advanced Mathematics 62. Cambridge University Press.

Stembridge, J. R. (1995). A Maple package for symmetric functions. *Journal of Symbolic Computation*, **20**, 755–768.

Storjohann, A. and Villard, G. (2005). Computing the rank and a small nullspace basis of a polynomial matrix. In *Proceedings of ISSAC'05*, pages 309–316.

Strzeboński, A. (2000). Solving systems of strict polynomial inequalities. *Journal of Symbolic Computation*, **29**, 471–480.

Tarski, A. (1951). *A Decision Method for Elementary Algebra and Geometry*. University of California Press.

Vajda, S. (1989). *Fibonacci & Lucas Numbers, and the Golden Section*. Ellis and Horwood Ltd.

van der Poorten, A. (1979). A proof that Euler missed... — Apéry's proof of the irrationality for $\zeta(3)$. *The Mathematical Intelligencer*, **1**, 195–203.

van der Poorten, A. (2004). Elliptic curves and continued fractions. preprint available online at http://www.arxiv.org/math.NT/0403225.

van der Put, M. and Singer, M. (1997). *Galois Theory of Difference Equations*, volume 1666 of *Lecture Notes in Mathematics*. Springer.

Watt, S. M. (2002). Solution to problem 10844. *The American Mathematical Monthly*, **109**(1), 78.

Wester, M. J. (1999). A critique of the mathematical abilities of CA systems. *Computer Algebra Systems: A Practical Guide*.

Wilf, H. S. and Zeilberger, D. (1990). Rational functions certify combinatorial identities. *Journal of the American Mathematical Society*, **3**, 147–158.

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

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

Zimmermann, B. (2002). On Fibonacci numbers. In *Seminar of the RISC Combinatorics Group*.

Zimmermann, B. (2005). *Definite Symbolic Summation and Integration*. Ph.D. thesis, RISC-Linz, Johannes Kepler Universität Linz. (in preparation).

Zürcher, B. (1994). *Rationale Normalformen von pseudo-linearen Abbildungen*. Master's thesis, ETH Zürich.

List of Symbols

Mathematical Notation

The notation which is used in this thesis follows as much as possible the established standards. Below are listed, for disambiguation, the most important symbolisms that have been used, as well as declarations of symbols whose meaning varies in the literature. The order follows roughly the order of first appearance in the text. Concerning notation, also see the remarks on page 4.

$\mathbb{N} = \{1,2,3,\dots\}$	The set of natural numbers
$\mathbb{N}_0 = \{0,1,2,3,\dots\}$	The set of nonnegative integers
$\mathbb{Z},\mathbb{Q},\mathbb{R},\mathbb{C}$	Sets of integers, rational, real, complex numbers
$\mathbb{Z}_p$	The residue class ring $\mathbb{Z}/\mathbb{Z}p$ ($p \in \mathbb{Z}$)
$f\colon A \hookrightarrow B$	f is an injective (difference) homomorphism from A to B
$f\colon A \twoheadrightarrow B$	f is a surjective (difference) homomorphism from A to B
$f\colon A \xrightarrow{\sim} B$	f is a bijective (difference) homomorphism from A to B
$\mathbb{K}$	A computable field of characteristic zero
$\bar{\mathbb{K}}$	The algebraic closure of the field $\mathbb{K}$
$\mathbb{K}[X] = \mathbb{K}[x_1,\dots,x_n]$	Ring of polynomials in $x_1,\dots,x_n$ with coefficients in $\mathbb{K}$
$\deg p$	Total degree of a polynomial p
$\deg_x p$	Degree of a polynomial p w.r.t. the variable x
$[X]$	The monoid of all terms in $x_1,\dots,x_n$
$\mathbb{K}(X) = \mathbb{K}(x_1,\dots,x_n)$	Field of rational functions in $x_1,\dots,x_n$ over $\mathbb{K}$
$Q(R)$	The quotient field of a ring R
$\mathfrak{a} \trianglelefteq R$	$\mathfrak{a}$ is an ideal of the ring R
$\mathfrak{a},\mathfrak{b},\mathfrak{c},\dots$	Symbols to denote ideals
$\mathfrak{a} = \langle p_1,\dots,p_s\rangle$	The ideal $\mathfrak{a}$ is generated by $p_1,\dots,p_s$
$V(\mathfrak{a}) \subseteq \mathbb{K}^n$	The affine variety of the ideal $\mathfrak{a} \trianglelefteq \mathbb{K}[X]$
$I(A) \trianglelefteq \mathbb{K}[X]$	The vanishing ideal of the set $A \subseteq \mathbb{K}^n$
$\mathfrak{a}+\mathfrak{b}$	Sum of ideals $\mathfrak{a},\mathfrak{b}$

$\mathfrak{a} \cdot \mathfrak{b}$	Product of ideals $\mathfrak{a}, \mathfrak{b}$
$\operatorname{Rad}(\mathfrak{a}) \trianglelefteq R$	The radical ideal of some ideal $\mathfrak{a} \trianglelefteq R$
$[b_1, \ldots, b_m] \subseteq R^d$	The smallest submodule of R^d containing $b_1, \ldots, b_m$ (R a ring)
$\prec, \succ, \preccurlyeq, \succcurlyeq$	(Strict) term order
$\mathrm{LM}(p), \mathrm{LT}(p), \mathrm{LC}(p)$	Leading monomial, leading term, and leading coefficient, respectively, of the polynomial $p \in \mathbb{K}[X]$
$\varepsilon_1, \ldots, \varepsilon_d$	Generators of the free $\mathbb{K}[X]$-module $\mathbb{K}[X]^d$
$\operatorname{Syz}(p_1, \ldots, p_n) \subseteq R^n$	Syzygy module of $p_1, \ldots, p_n \in R$ (R a ring)
$\mathbb{K}^{\mathbb{N}}$	The (difference) ring of all univariate sequences $f\colon \mathbb{K} \to \mathbb{N}$
$\mathbf{E}\colon \mathbb{K}^{\mathbb{N}} \to \mathbb{K}^{\mathbb{N}}$	Shift operator for sequences
F_n, L_n, P_n, Q_n	The sequences of Fibonacci numbers, Lucas numbers, Pell numbers, and Pell-Lucas numbers, respectively
$P_n(x), P_n^{(\alpha,\beta)}(x), C_n^m(x), L_n^\alpha(x)$	The families of Legendre, Jacobi, Gegenbauer, and Laguerre polynomials, respectively
$\sum_{k=a}^b f(k)$	The sum $f(a) + f(a+1) + \cdots + f(b)$
$\prod_{k=a}^b f(k)$	The product $f(a)f(a+1)\cdots f(b)$
$\mathbf{K}_{k=a}^b (f(k)/g(k))$	The continued fraction with numerators $f(a+1), f(a+2), \ldots, f(b)$ and denominators $g(a), g(a+1), \ldots, g(b)$
$n!, \binom{n}{k}$	Factorial $n! = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$ and binomial coefficient $\binom{n}{k} = n(n-1)\cdots(n-k+1)/k!$
$\mathrm{H}_n, \mathrm{H}_n^{(r)}$	Harmonic number, and harmonic number of rth order, $\mathrm{H}_n^{(r)} = \sum_{k=1}^n k^{-r}$, $\mathrm{H}_n = \mathrm{H}_n^{(1)}$
$\operatorname{sgn} x$	Sign function, $\operatorname{sgn}(x) := -1$ for $x < 0$, $\operatorname{sgn}(x) := 1$ for $x > 0$ and $\operatorname{sgn}(0) := 0$
$\mathbb{K}\{t_1, \ldots, t_m\}$	The free difference ring over $\mathbb{K}$ in m difference variables
$\mathbb{K}\langle t_1, \ldots, t_m\rangle$	The free difference field over $\mathbb{K}$ in m difference variables
$\mathbb{K}\{t_1, \ldots, t_m\}_r, \mathbb{K}\{t_1, \ldots, t_m\}_r^\ell$	Truncated difference rings (cf. pages 26 and 39)
$\operatorname{const} R$	The ring of all constants of the difference ring R
$\langle\langle p_1, \ldots, p_m\rangle\rangle \trianglelefteq R$	The difference ideal generated by $p_1, \ldots, p_m \in R$
$\lfloor q \rfloor$	The greatest integer $n \in \mathbb{Z}$ with $n \leq q$ (Floor function)
$a(n) \xrightarrow{n \to \infty} a$	The sequence $a(n)$ converges to a for $n \to \infty$

$\operatorname{ann}(f_1(n),\dots,f_m(n))$	Difference ideal of all polynomials $p \in \mathbb{K}\{t_1,\dots,t_m\}$ that vanish upon substitution $s^i t_j \mapsto f_j(n+i)$ (annihilating ideal)
$\operatorname{alg}(f_1(n),\dots,f_m(n))$	Algebraic part of the annihilating ideal (see Chapter 6)
$\lVert b\rVert, \lVert b\rVert_\infty$	Euclid norm and maximum norm of some vector b

We do not distinguish between column and row vectors.

Pseudo Code Constructs

Algorithms in the thesis are described using a pseudo code language. In the algorithm listings, we give readability preference over formal rigor. Whereever it seems appropriate, we also use natural language to describe steps of an algorithm. The "syntax" of the language should be self explanatory. The explanations below are for the sake of completeness.

```
while A do
   B
C
```

If A is true, execute B. If afterwards, A is still true, execute B again. Repeat, until A becomes false. Then proceed with C.

```
repeat
   B
while A
C
```

Execute B. If afterwards A is false, proceed with C. Otherwise, execute B again until A becomes true.

```
for i = a to b do
   A(i)
```

Execute $A(a)$, then $A(a+1)$ then $A(a+2)$, and so on, and finally $A(b)$.

```
if A then
   B
elseif C then
   D
else
   E
```

If A is true, execute B. If A is not true, but C is true, execute D. If neither A nor C is true, execute E. The "else if" clause can be omitted, or there might be several of them. Also the "else" part can be omitted, in which case nothing is executed if all conditions yield false.

```
return A
```

Declare A as output of the algorithm and stop.

Blocks are indicated by indentation only. Loop- and branch constructs only apply to those lines following the key word which are indented one level deeper than the key word itself. For example,

```
for i = 1 to 5
   Print i
Print 0
```

produces the output $1,2,3,4,5,0$, while

```
for i = 1 to 5
   Print i
   Print 0
```

produces the output $1,0,2,0,3,0,4,0,5,0$.

Algorithm 3.9 uses arrays. A one dimensional array a with n components $a_1,\ldots,a_n$ is written $a=[a_i : i=1,\ldots,n]$. The function append composes a new array from a given one by appending a given object to it. The notation for all other data structures should not be problematic.

Index

Curriculum Vitæ

Personal data

Name	Manuel Kauers
Nationality	German
Date and place of birth	Feb. 20, 1979, Lahnstein, Germany
Marital state	Single, no children

Contact

Email	manuel@kauers.de
WWW	http://www.kauers.de

Education

1998	Abitur (high school graduation) at staatl. Gymnasium Lahnstein
1998–2002	Studies in computer science at the University of Karlsruhe, Germany
2002	Diploma degree in computer science (Dipl.-Inform.) *Diploma thesis:* "Verstehen natürlicher Sprache durch statistische Übersetzung in eine termbasierte Interlingua" (Understanding natural language by statistical translation into a term based interlingua) *Thesis advisor:* Prof. Alex Waibel
2002–2005	Doctorate studies in symbolic computation at the Research Institute for Symbolic Computation (RISC), Johannes Kepler University of Linz, Austria