On Combinatorial Identities: Symbolic Summation and Umbral Calculus

Dissertation

zur Erlangung des akademischen Grades
"Doktor der technischen Wissenschaften"

Eingereicht von

Roberto Pirastu

Juli 1996

Erster Begutachter: Univ.-Doz. Jochen Pfalzgraf
Zweiter Begutachter: A.Univ.-Prof. Günter Pilz

Angefertigt am Forschungsinstitut für Symbolisches Rechnen
Technisch-Naturwissenschaftliche Fakultät
Johannes Kepler Universität Linz

Eidesstattliche Erklärung

Ich versichere, daß ich die Dissertation selbständing verfaßt habe, andere als die angegebenen Quellen und Hilfsmittel nicht verwendet und mich auch sonst keiner unerlaubten Hilfe bedient habe.

Roberto Pirastu
Hagenberg, 10. Juli 1996

Acknowledgments

This thesis would not have been possible without the help of several persons – to which I owe my gratitude. I apologize that I can not mention each name here.

My heartful thanks go to my thesis adviser, Peter Paule, for his selfless support and guidance during my graduate education. I will always admire his ability to infect me with his enthusiasm for research work, especially when I could not imagine that he would succeed.

I thank Jochen Pfalzgraf and Günter Pilz for serving on my thesis committee, and Bruno Buchberger for making my thesis possible through his great work as the chairman of the Research Institute for Symbolic Computation.

I wish to thank the co-authors of the research work reported in this thesis. I thank Volker Strehl for his encouraging attention to my PhD work after I left Erlangen. I thank also Kurt Siegl and Carla Limongelli for sharing with me their knowledge about parallel systems. My special thanks go to Giorgio Nicoletti for inviting me to the University of Bologna, where I stayed twice as a guest. Besides opening my horizon to the exciting world of umbral calculus and coalgebras, the time I spent in Bologna was a wonderful and intense experience for me.

Many members of RISC contributed to this thesis in one way or another. They all helped me by making my stay in Austria a great pleasure, and I thank all of them. I would like to thank in particular the members of the combinatorics seminar, who provided a stimulating atmosphere for my studies, and also all my office-mates during the years at RISC.

This research was partially supported by the *Commission of the European Communities* in the framework of the program "Human Capital and Mobility", contract Nr. ERBCHBICT930501. I thank Renzo Caddeo and Luigi Cerlienco for making me aware of this possibility of founding and for their constant support in my activities.

Zusammenfassung

Die Dissertation besteht aus drei Kapiteln.

Im ersten Kapitel wird das Problem der indefiniten Summation rationaler Funktionen behandelt, in Anlehnung an meine Arbeit [Pir95] und an [PSb], die in Zusammenarbeit mit Volker Strehl entstanden ist. Im Rahmen einer geeigneten algebraischen Spezifikation wird die Struktur der Lösungen beschrieben. Mit Hilfe eines kombinatorischen Analogons zur Gosper-Petkovšek Darstellung für rationale Funktionen stellen wir einen Algorithmus vor, der, unter bestimmten Randbedingungen, eine optimale Lösung liefert. Ferner vergleichen wir diesen *optimalen* Algorithmus mit den anderen, in der Literatur bekannten Methoden.

Im zweiten Kapitel werden parallele Implementierungen beschrieben. Im ersten Abschnitt berichten wir über eine Zusammenarbeit mit Kurt Siegl [PSa], die sich mit der Implementierung zweier im Kapitel 1 angegebenen Algorithmen im parallelen Computer-Algebra System ||MAPLE|| ("parallel Maple") befaßt. Im Vergleich zu den sequentiellen Implementierungen erreichen wir einen Speedup-Faktor von bis zu 7. Im zweiten Abschnitt wird ein paralleler Algorithmus beschrieben, der lineare Gleichungssysteme über rationale Zahlen mit Hilfe der p-adischen Arithmetik löst. Die Implementierung entstand in Zusammenarbeit mit Carla Limongelli [LP96]. Aus systematisch durchgeführten Experimenten hat sich herausgestellt, daß die Implementierung mit p-adischer Darstellung rationaler Zahlen zu einem Speedup-Faktor führt, der vergleichbar zu den üblichen modularen Methoden ist.

Das dritte Kapitel ist dem umbralen Kalkül gewidmet und ist aus einer Zusammenarbeit mit Giorgio Nicoletti entstanden. Durch eine möglichst weitgehende Reduktion auf lineare Algebra erhält man eine neue Beschreibung des Kalküls, durch welche die *einfache*, *kombinatorische* Natur der algebraischen Zusammenhänge hervorgehoben wird. Die durch Reduktion bzw. Abstraktion erzielte Verallgemeinerung schließt auch nicht-polynomiale Klassen mit ein.

Contents

Overview

This thesis is divided into three chapters: the first two chapters are closely related to each other, since they both treat algebraic and algorithmic aspects of the problem of *symbolic summation of rational functions*. The third one is independent of the first two, and is devoted to an algebraic and combinatorial study of the *umbral calculus*. Here we give a short summary of the results.

In the first chapter we consider the problem of *indefinite summation of rational functions*. This problem mainly consists in inverting the *forward difference operator* Δ over the field of rational functions $\mathbb{K}(x)$, i.e., in finding for any $\alpha \in \mathbb{K}(x)$ a $\beta \in \mathbb{K}(x)$ such that $\alpha(x) = \Delta\beta(x) := \beta(x+1) - \beta(x)$. In the case where no such rational β exists, one is interested in a "minimal correction" $\gamma \in \mathbb{K}(x)$ of α, for which there is a β with $\alpha - \gamma = \Delta\beta$.

We propose an algebraic framework for the problem and, since the solutions (β, γ) are not unique, we are particularly interested in describing the structure of all possible solutions in dependence of the input α. In this context the structure of the rational functions can be combinatorially modeled by sequences of integers, which makes the description more intuitive.

We propose an algorithm that computes a solution (β, γ), where also the denominator polynomial of β is *optimal*, in a sense that will be explained in Chapter 1. This method is based on a combinatorial analog of the Gosper-Petkovšek representation of rational functions for sequences. This part of the chapter reports on joint work with Volker Strehl [PSb].

In addition, we describe other known algorithms (of Abramov, Moenck and Paule, resp.) for computing a solution of the rational summation problem and their implementation in the computer algebra system Maple, as reported in [Pir95]. We compare the algorithms to our optimal one.

Finally, we give some examples of combinatorial identities due to Ramanujan which can be proven by rational summation methods.

In Chapter 2 we describe *parallel implementations* of symbolic computation algorithms. The first part is devoted to the parallel implementation of Abramov's and Paule's algorithm from Chapter 1, respectively. They are implemented in the parallel computer algebra system ||MAPLE|| (speak: "parallel Maple") on a workstation cluster and are also described in the joint work with Kurt Siegl [PSa].

The second part reports on joint work with Carla Limongelli [LP96]. We describe a parallel implementation of a solver for systems of linear equations over the rational numbers. The parallelization is based on the truncated p-adic representation of rational numbers. Our comparisons and experiments show that p-adic representation is a suitable tool for parallelizing the algorithm for rational numbers, compared to usual modular techniques.

The third chapter is devoted to *umbral calculus* and is based on joint work with

Giorgio Nicoletti.

Umbral calculus originally denoted some manipulation techniques for sequences and goes back to mathematicians as, for instance, Sylvester and Lucas. In short, they considered a sequence $(a_i)_{i\in\mathbb{N}}$ of numbers as a sequence of powers $(a^i)_{i\in\mathbb{N}}$. After doing the computations on the polynomials in the *symbol* a, they interpreted the exponents as indices again, carrying on the result to the sequence situation. One used to call a the *umbra* associated to the sequence.

Consider, for instance, the Bernoulli numbers $(B_i)_{i\in\mathbb{N}}$, which satisfy the recurrence

$$\sum_{i=0}^{n-1} \binom{n}{i} B_i = 0 \quad (n > 1) \qquad \text{and} \qquad B_0 = 1 \ .$$

Following Lucas in Chapter XIII of [Luc91], we write B^i for B_i and the recurrence becomes the more concise form $(B+1)^n - B^n = 0$. Similarly, to the Bernoulli polynomials $B_n(x) = \sum_{i=0}^{n} \binom{n}{i} B_{n-i} x^i$ we can associate the umbral expression $(B+x)^n$. See, among others, the survey on elementary and mnemonic uses of umbral calculus by Guinand [Gui79].

One of the most interesting aspects of umbral calculus is that it can be applied to many sequences of polynomials, which are fundamental or useful in several branches of mathematics, like, among others, finite differences calculus, probability theory, combinatorics and invariant theory.

Although such symbolic methods *apparently* worked, there was no solid theoretical justification of this fact. Several authors tried to give a formally correct, axiomatic treatment of the subject, but in our eyes the first successful attempt in this direction has been done by Rota [RKO73, RR78], where he makes use of the unifying concepts from linear algebra. In principle, one considers a sequence $(a_i)_{i\in\mathbb{N}}$ as the evaluation of a linear operator L on the basis $(x^i)_{i\in\mathbb{N}}$ of the vector space $\mathbb{K}[x]$, or on another suitable basis. This way $(a_i)_{i\in\mathbb{N}}$ can be seen as the coordinates of L with respect to a pseudobasis of the dual $(\mathbb{K}[x])^* \cong \mathbb{K}[[x]]$, equipped with an appropriate topology. In this model the so-called sequences of polynomials of binomial type and the shift-invariant operators play a particular role. For a fundamental treatment of this subject we refer to [RKO73], while a neat introduction is given in [CNP85].

So, umbral calculus reduces to the study of particular bases of the space $\mathbb{K}[x]$, invariant under certain operators.

As Rota himself suggested, this structure can be made more clear using the concept of coalgebra. Several authors have already followed this path.

In this thesis we present a description with less requirements to the structure of the underlying set, which does not need to be a polynomial ring. We consider a general vector space $\mathbb{V}$ of countably infinite dimension over a field of arbitrary characteristic, and certain endomorphisms. In principle, this is already enough to develop the whole structure, and there are examples of spaces which fit into this scheme, and are not isomorphic to the polynomial one, especially for nonzero characteristic.

Our aim is to show the *simplicity* of the underlying algebraic structure and to derive several well-known umbral facts in a more natural, but also more abstract way. In addition, this framework helps to clearify the connection between sequences from umbral calculus and *recursive matrices* [BBN82]. Such matrices are particularly important because they describe inverse relations over arbitrary sequences.

1

Summation of Rational Functions

1.1 Problem description

Let $\mathbb{K}$ be a field of characteristic 0. As usual, $\mathbb{K}(x)$ denotes the field of rational functions over $\mathbb{K}$. Elements $\alpha = \alpha(x) \in \mathbb{K}(x)$ are written as quotients $\alpha = f/g$, where f and $g \neq 0$ are polynomials in x over $\mathbb{K}$. This representation is called *reduced* if $\gcd(f, g) = 1$ and if g is monic. Usually, but not always, we will assume that elements of $\mathbb{K}(x)$ are represented in this way. A *proper* rational function is an element $\alpha = f/g \in \mathbb{K}(x)$ such that $\deg f < \deg g$, where the constant polynomial 0 has degree -1. The constant 0 is the only proper rational function which is constant. The proper rational functions form a $\mathbb{K}$-sub-algebra of $\mathbb{K}(x)$, denoted by $\mathcal{R}$. By polynomial division, $\mathbb{K}(x) \cong \mathbb{K}[x] \oplus \mathcal{R}$.

The shift operator E and the (forward) difference operator Δ on $\mathbb{K}(x)$ are defined as usual:

$$
\begin{aligned}
E &: \mathbb{K}(x) \to \mathbb{K}(x) &&: \alpha(x) \mapsto \alpha(x+1) \\
\Delta = E - I &: \mathbb{K}(x) \to \mathbb{K}(x) &&: \alpha(x) \mapsto \alpha(x+1) - \alpha(x)
\end{aligned}
$$

Note that E is a $\mathbb{K}$-algebra isomorphism, Δ is $\mathbb{K}$-linear and has the constant functions as its kernel. If restricted to the sub-algebra $\mathcal{R}$, E is still a $\mathbb{K}$-algebra isomorphism, and Δ then is injective, since 0 is the only element in its kernel. *Indefinite summation* of rational function essentially asks for inverting the linear operator Δ. Inverting Δ on polynomials is trivial, since, for instance, it is known that $\Delta^{-1} x^m = \frac{B_{m+1}(x)}{m+1}$, where $B_m(x)$ is the sequence of Bernoulli polynomials.

So we can restrict our attention to the algebra $\mathcal{R}$. This also has the advantage that $\Delta^{-1}\alpha$ is uniquely determined - if it exists! As is well known, the latter is not always the case: Δ is not surjective on $\mathcal{R}$. E.g., for any $j > 0$, there is no $\beta \in \mathbb{K}(x)$ such that $\Delta\beta = 1/x^j$. More generally: if $f/g^j \in \mathcal{R}$ is a rational function in reduced form, with g irreducible, then it does not belong to $\Delta\mathcal{R}$. In view of this phenomenon one has to make a choice between the following alternatives:

- Asking for a decision procedure for the existence of $\Delta^{-1}\alpha \in \mathcal{R}$, and giving an algorithm to construct such an element in the case of a positive answer. First approaches and algorithms in this direction were presented by Abramov in [Abr71] and by Gosper in [Gos78].

- Enlarging the domain of functions under consideration (e.g. by adding polygamma functions), so that at least every $\alpha \in \mathcal{R}$ has an inverse with respect to Δ. The approach presented by Moenck in [Moe77], for instance, goes in this direction. A general method in analogy to Risch's integration method is described by Karr in [Kar81, Kar85].

- Considering a "refined" rational summation problem: given $\alpha \in \mathcal{R}$, construct $\beta \in \mathcal{R}$ which is as "close" as possible to what we expect $\Delta^{-1}\alpha$ to be, this means, making $\gamma = \alpha - \Delta\beta \in \mathcal{R}$ as "small" as possible. In particular this requires: for $\alpha \in \Delta\mathcal{R}$ one should get the true inverse, i.e., $\gamma = 0$, and in general, if the same procedure is applied to the difference γ, this should not lead to any improvement. This problem has apparently been first stated by Abramov in [Abr75].

Here we will concentrate on the last alternative. As a reasonable measure of "smallness" we will take the degree of the denominator polynomial in the reduced presentation of γ.

Definition 1.1 *For $\alpha = f/g \in \mathcal{R}$ in reduced form, we define*

$$\| \alpha \| := \deg g$$

and in particular $\| 0 \| = 0$.

Note that $\| \, . \, \|$ induces a metric on $\mathcal{R}$.

The rational summation problem can the be stated as follows.

Rational Summation Problem. Given $\alpha \in \mathcal{R}$, determine β and γ in $\mathcal{R}$ such that $\alpha = \Delta\,\beta + \gamma$, where $\| \gamma \|$ is minimal.

Thus one asks for an element of $\Delta\mathcal{R}$ which is closest to α with respect to the $\| \, . \, \|$-metric. Naturally, we will say that α is *summable* if $\alpha \in \Delta\mathcal{R}$, i.e., if there exists a pair (β, γ) with $\gamma = 0$. In this case we say that the indefinite sum over α has a *rational closed form*, since we can express any finite sum as

$$\sum_{k=a}^{b} \alpha(k) = \beta(b+1) - \beta(a)$$

i.e., as a rational function in the boundaries a and b, when no pole of α falls into $[a, b]$.

In the following sections we introduce a suitable algebraic framework for the problem and develop an approach to the solution of the problem.

Our aim is to study first the structure of the denominator polynomials of all possible solutions (β, γ), in order to make clear how different solutions are related to each other (see Section 1.4).

Then these observations lead us in Subsection 1.4.5 to an "Ansatz" for the denominator polynomials. Substituting these candidates into the equation $\alpha = \Delta\beta + \gamma$, in analogy to Hermite-Ostrogradski's strategy for rational integration the numerators of β and γ can be determined by coefficient comparison.

In addition, the Ansatz we compute is *optimal*, in a sense that will be specified in Subsection 1.4.4.

In Subsection 1.5.1 we give a combinatorial analog of the Gosper-Petkovšek representation of rational functions. We will show from this combinatorial representation that such an optimal Ansatz can be computed from the Gosper-Petkovšek representation of $E\alpha/\alpha$. A more algebraic motivation of this fact will be given in Subsection 1.5.3, by means of the concept of *Greatest Factorial Factorization*, introduced by Paule in [Pau93, Pau95].

Section 1.6 is devoted to the description of several algorithms known in literature for the rational summation problem. We compare them with our approach, particularly with respect to the optimality of the solutions computed.

To conclude, in Section 1.8 we present some applications of identities involving rational functions, which can be proven by means of rational summation.

The content of this chapter is partly due to a joint work with Volker Strehl, see [PSb].

1.2 Localization

For any monic irreducible polynomial $g \in \mathbb{K}[x]$ let $\mathcal{R}_g$ denote the sub-algebra of rational functions $f/g^i \in \mathcal{R}$, where $i \geq 0$. By partial fraction representation

$$\mathcal{R} \cong \bigoplus_g \mathcal{R}_g$$

where $\bigoplus_g$ runs over all monic irreducible polynomials. Clearly

$$\text{if } \alpha = \sum_g \alpha_g \ , \text{ where } \ \alpha_g \in \mathcal{R}_g \ , \text{ then } \ \| \alpha \| = \sum_g \| \alpha_g \|$$

When dealing with the shift operator E and the difference operator Δ, one has to consider E-orbits, i.e., shift-invariant subspaces of $\mathcal{R}$. For any monic irreducible polynomial $g \in \mathcal{R}$ we put

$$\mathcal{R}_{[g]} := \bigoplus_{i \in \mathbb{Z}} \mathcal{R}_{E^i g}$$

Here $[g]$ denotes the class of monic irreducible polynomials *shift-equivalent* to g, i.e., the polynomials $E^i g(x) = g(x+i)$ for $i \in \mathbb{Z}$. Note that if the polynomial g is irreducible over $\mathbb{K}$, then so is $E^i g$ for any $i \in \mathbb{Z}$. In the following the notation $\bigoplus_{[g]}$ and $\sum_{[g]}$ will be used to indicate direct sums or elements of direct sums over a system of representatives for the shift-equivalence classes (or, for shortness, shift-classes) of monic irreducible polynomials. Thus

$$\mathcal{R} \cong \bigoplus_{[g]} \mathcal{R}_{[g]}$$

with

$$\alpha = \sum_{[g]} \alpha_{[g]} \quad \text{and} \quad \alpha_{[g]} = \sum_{i \in \mathbb{Z}} \alpha_{E^i g}$$

It is clear that any equation

$$\alpha = \Delta\beta + \gamma \quad (\alpha, \beta, \gamma \in \mathcal{R})$$

localizes to

$$\alpha_{[g]} = \Delta\beta_{[g]} + \gamma_{[g]}$$

for any monic irreducible polynomial g. And since

$$\| \alpha \| = \sum_{[g]} \| \alpha_{[g]} \|$$

we can state

Proposition 1.1 *If $(\beta, \gamma) \in \mathcal{R}^2$ is a solution for the rational summation problem for $\alpha \in \mathcal{R}$, then for each monic irreducible polynomial g the pair $(\beta_{[g]}, \gamma_{[g]}) \in \mathcal{R}^2_{[g]}$ is a solution of the rational summation problem for $\alpha_{[g]}$, and conversely.*

This shows that for solving the problem and for answering the uniqueness question it suffices to study the "local" situation in the components $\mathcal{R}_{[g]}$. And in particular:

Corollary 1.1 *$\alpha \in \mathcal{R}$ is summable if and only if $\alpha_{[g]} \in \mathcal{R}_{[g]}$ is summable for each (shift-equivalence class of) monic irreducible polynomial(s) g.*

1.3 The spectrum and basic operators on sequences

Before considering the structure of the β and γ part of the localized summation problem, let us introduce a bit of notation.

Definition 1.2 *Given $q \in \mathbb{K}[x]$ and a monic irreducible polynomial g, we define the* spectrum *of q with respect to g as the doubly infinite sequence*

$$\langle q, g \rangle = (a_i)_{i \in \mathbb{Z}} \quad , \textit{ where } \quad E^i g^{a_i} \, \| \, q$$

i.e., a_i is the maximal integer, such that $E^i g^{a_i} \mid q$. For $\alpha = p/q \in \mathcal{R}$ in reduced form we define the spectrum of α with respect to g by $\langle \alpha, g \rangle := \langle q, g \rangle$.

Note that $\langle \alpha, g\rangle = (a_i)_{i\in\mathbb{Z}}$ means

$$\alpha_{[g]} = \sum_{i\in\mathbb{Z}} \alpha_{E^i g} = \sum_{i\in\mathbb{Z}} E^i \frac{f_i}{g^{a_i}}$$

with respect to the canonical decomposition of $\alpha_{[g]}$, where the f_i/g^{a_i} are in reduced form. Denote by $g^{\mathbf{a}}$ the polynomial $\prod_{i\in\mathbb{Z}} E^i g^{a_i}$ for any sequence $\mathbf{a}$ with only nonnegative, only finitely many nonzero components.

The sequences $\langle \alpha, g\rangle$ are non-negative integer sequences with finite support, but for reasons that become evident later on, we have to consider more general classes of sequences as well.

The spectrum of a polynomial (or of a rational function, resp.) can be graphically represented in the following way. For a spectrum $\langle q, g\rangle = (a_i)_{i\in\mathbb{Z}}$, as for any non-negative integer sequence $(a_i)_{i\in\mathbb{Z}}$, we just draw a sequence of stacks with a_i boxes at the i-th position. We use this pictorial representation in order to make the concepts and proofs more understandable.

For example, consider the polynomial $q = (x-2)x(x+1)^3(x+3)^2(x+4)$. Then, the spectrum $\langle q, x\rangle$ is represented in Fig. 1.1. In the picture, the stack at position 3 has two boxes, as $(x+3)^2 \parallel q$.

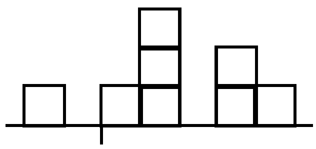

Figure 1.1: Spectrum $\langle (x-2)x(x+1)^3(x+3)^2(x+4), x\rangle$

In the following we will consider doubly infinite sequences $\mathbf{a} = (a_i)_{i\in\mathbb{Z}}$ and we need the following operators acting on them.

Definition 1.3 *Naturally, the arithmetic operations "+" and "−" are defined coordinatewise. Analogously for the infimum "$\wedge$" and the partial order relation "$\leq$" we have:*

$$\begin{aligned} \wedge : ((a_i)_{i\in\mathbb{Z}}, (b_i)_{i\in\mathbb{Z}}) \mapsto \mathbf{a}\wedge\mathbf{b} \quad &:= \quad (\min\{a_i, b_i\})_{i\in\mathbb{Z}} \\ \mathbf{a} \leq \mathbf{b} \quad &\Longleftrightarrow \quad \forall i\in\mathbb{Z} \; : \; a_i \leq b_i \end{aligned}$$

We define a shift and a delta operator on sequences:

$$\begin{aligned} \epsilon : (a_i)_{i\in\mathbb{Z}} \mapsto \epsilon\,\mathbf{a} \quad &:= \quad (a_{i-1})_{i\in\mathbb{Z}} \\ \delta : (a_i)_{i\in\mathbb{Z}} \mapsto \delta\,\mathbf{a} \quad &:= \quad (a_i - a_{i-1})_{i\in\mathbb{Z}} = (1-\epsilon)\,\mathbf{a} \end{aligned}$$

The inverse of the δ operator is the summation operator σ given by

$$\sigma : (a_i)_{i\in\mathbb{Z}} \mapsto \sigma\,\mathbf{a} := \left(\sum_{j\leq i} a_j\right)_{i\in\mathbb{Z}}$$

Note that $\sigma\,\mathbf{a}$ is only defined for sequences $\mathbf{a}$ where $\sum_{j\leq i} a_j$ is finite for all $i \in \mathbb{Z}$. In particular, this holds if $\mathbf{a}$ has finite support.

Two further operators, which are also only defined on appropriate subspaces, are the left-to-right (right-to-left, respectively) maximum operators.

$$\begin{aligned}
\overrightarrow{\mu} : (a_i)_{i\in\mathbb{Z}} \mapsto \overrightarrow{\mu}\,\mathbf{a} &:= \left(\max_{j\leq i} a_j\right)_{i\in\mathbb{Z}} \\
\overleftarrow{\mu} : (a_i)_{i\in\mathbb{Z}} \mapsto \overleftarrow{\mu}\,\mathbf{a} &:= \left(\max_{j\geq i} a_j\right)_{i\in\mathbb{Z}}
\end{aligned}$$

Let us consider an example for the sequence with finite support $\mathbf{a} := (a_i)_{i\in\mathbb{Z}} = (\ldots, 0, 0, \underline{1}, 0, 1, 2, 0, \ldots)$. In Figure 1.2 and 1.3 we show the corresponding action on $\mathbf{a}$ of the operators defined above.

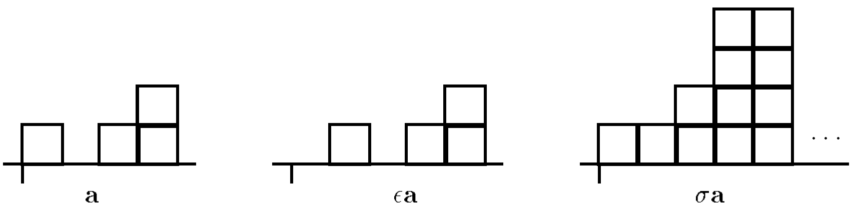

Figure 1.2: Examples for the operators ϵ and σ on sequences.

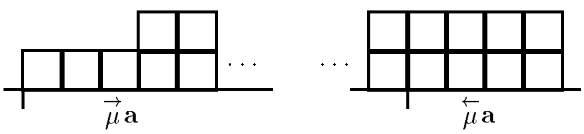

Figure 1.3: Examples for the left-to-right and right-to-left operators on sequences.

1.4 The structure of the local solutions

Following Proposition 1.1, we first concentrate on the the local solution of the rational summation problem with respect to a given monic irreducible polynomial g.

Any arbitrary element $\alpha \in \mathcal{R}_{[g]}$ can be written as

$$\alpha = \sum_{i \in \mathbb{Z}} E^i \frac{f_i}{g^{a_i}}$$

where the sum actually is finite. The strategy will be to introduce local transformations in order to *reduce* any pair of non-zero summands above to the form

$$E^i \frac{f_i}{g^{a_i}} + E^j \frac{f_j}{g^{a_j}} = \Delta \beta_{i,j} + E^j \frac{f_{i,j}}{g^b} .$$

where $\| E^i \frac{f_i}{g^{a_i}} + E^j \frac{f_j}{g^{a_j}} \| > \| E^j \frac{f_{i,j}}{g^b} \|$.

Applying such transformations stepwise, and summing up the partial β parts, we eventually get a decomposition of the form $\alpha = \Delta \beta + \gamma$ with $\gamma = E^j(f/g^c)$ for some j, and we will show that γ is minimal with respect to the metric $\| \cdot \|$. So, (β, γ) is a solution of the rational summation problem.

In addition, since the solutions are not unique (we can get one for each i), the structure of the β part is studied with respect to the corresponding γ part.

1.4.1 Local transformations

We start with a simple observation:

For each $d \in \mathbb{Z}$ the operator $\Delta_d = E^d - I$ is divisible by $\Delta_1 = \Delta$:

$$\Delta_d = E^d - I = \begin{cases} \Delta \cdot (I + E + E^2 + \cdots + E^{d-1}) & \text{if } d > 0 \\ -\Delta \cdot (E^{-1} + E^{-2} + \cdots + E^d) & \text{if } d < 0 \end{cases}$$

which means that

$$\Delta_d(\frac{f}{g^a}) = \Delta \left(\frac{\Delta_d}{\Delta} \frac{f}{g^a} \right) \in \Delta \mathcal{R}_{[g]}$$

for each $f/g^a \in \mathcal{R}_{[g]}$. Thus the sum of any two terms $E^i(f_1/g^a)$ and $E^j(f_2/g^b)$ with $i \neq j$, appearing in the canonical decomposition of some $\alpha \in \mathcal{R}_{[g]}$, may be transformed according to either of the two identities (assuming $d = j - i > 0$):

$$(T_{R \to L}) \qquad E^i \frac{f_1}{g^a} + E^j \frac{f_2}{g^b} = E^i \left(\frac{f_1}{g^a} + \frac{f_2}{g^b} \right) + \Delta (E^{d-1} + \cdots + I) E^i \frac{f_2}{g^b}$$

$$(T_{L \to R}) \qquad E^i \frac{f_1}{g^a} + E^j \frac{f_2}{g^b} = E^j \left(\frac{f_1}{g^a} + \frac{f_2}{g^b} \right) - \Delta (E^{-d} + \cdots + E^{-1}) E^j \frac{f_1}{g^a}$$

In Figure 1.4 we show, as an example, the spectrum of the rational functions involved in transformation $(T_{L\to R})$.

From the point of view of the spectrum of the γ part, transformation $(T_{R\to L})$ reduces two stacks to the leftmost one, while $(T_{L\to R})$ pushes the leftmost onto the rightmost. Notice that up to a shift E^{j-i} (or E^{i-j}) the first terms on the right are the same, namely a shift of

$$\frac{f_1}{g^a} + \frac{f_2}{g^b} = \frac{f}{g^c} \quad \text{where } c \leq \max\{a, b\}. \tag{1.1}$$

If these rational functions are written in reduced form, then $c = \max\{a, b\}$ if $a \neq b$. Note that for c as in (1.1) we have that the transformations reduces the metric $\|\cdot\|$ on the γ part, since

$$\| E^i \frac{f_1}{g^a} + E^j \frac{f_2}{g^b} \| = (a+b) \cdot \deg g > c \cdot \deg g = \| \frac{f_1}{g^a} + \frac{f_2}{g^b} \|$$

On the other hand, by Lemma 1.1 below, the denominators of the second terms on the right sides are precisely given by $(E^{i+d-1}g^b) \cdots E^i g^b$, or $(E^{j-d}g^a) \cdots E^{j-1}g^a$. This observation will be relevant in the study of the β part of a solution in Subsection 1.4.3.

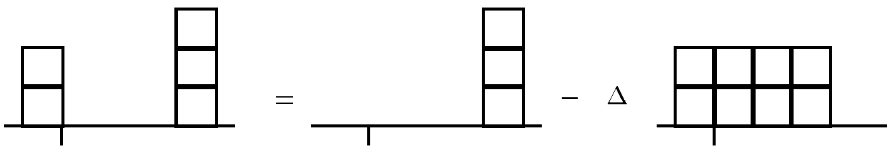

Figure 1.4: Local transformation $(T_{L\to R})$ for $\dfrac{f_1}{E^{-1}g^2} + \dfrac{f_2}{E^3 g^3}$

1.4.2 The structure of the γ part

We first need two technical lemmas, making precise what we said in Section 1.1 about rational functions of the form f/g^i.

Lemma 1.1 *Let $\alpha \in \mathcal{R}_{[g]}$ be given by*

$$\alpha = \sum_{i=l}^{l+m} E^i \frac{f_i}{g^{a_i}}$$

where all fractions are in reduced form. Then the reduced denominator of α is precisely

$$(E^l g^{a_l})(E^{l+1} g^{a_{l+1}}) \cdots (E^{l+m} g^{a_{l+m}})$$

Proof. Straightforward consequence of the fact that all $E^i g^{a_i}$ are pairwise relatively prime. ∎

Lemma 1.2 *Let $\alpha \in \mathcal{R}$ be of the form $\alpha = f/g^j$ for a monic irreducible g. Then α does not belong to $\Delta\mathcal{R}$.*

Proof. Assume that there is $\beta \in \mathcal{R}$ such that $\alpha = \Delta\beta$. If we look at the canonical decomposition of β with respect to g

$$\beta = \sum_{i=i_0}^{i_b} E^i \frac{f_i}{g^{b_i}}$$

with f_{i_0} and f_{i_b} nonzero, then we have

$$\Delta\beta = \sum_{i=i_0}^{i_b+1} E^i \frac{f_i'}{g^{b_i'}}$$

where $f_{i_0}' = -f_{i_0} \neq 0$ and $f_{i_b+1}' = f_{i_b} \neq 0$. By Lemma 1.1, this gives a contradiction to the assumption that the denominator of $\Delta\beta$ has the form $E^j g^j$. ∎

Consider now an arbitrary element $\alpha \in \mathcal{R}_{[g]}$, say

$$\alpha = \sum_{l \in \mathbb{Z}} E^l \frac{f_l}{g^{a_l}}$$

with all fractions in reduced form.

Let us start with $(\beta_0, \gamma_0) = (0, \alpha)$. Then we apply one of the transformations $(T_{L\to R})$ and $(T_{R\to L})$ to a pair of summands, say to those corresponding to the first indices i and j such that $i < j$ and $f_i \neq 0 \neq f_j$. We may apply any of $(T_{L\to R})$ and $(T_{R\to L})$, for instance $(T_{L\to R})$, and this gives us a decomposition

$$\alpha = \Delta \underbrace{(E^{-(j-i)} + \cdots + E^{-1}) E^j \frac{-f_i}{g^{a_i}}}_{\beta_1} + \underbrace{E^{i_j}\left(\frac{f_i}{g^{a_i}} + \frac{f_j}{g^{a_j}}\right) + \sum_{l > j+1} E^l \frac{f_l}{g^{a_l}}}_{\gamma_1}$$

Applying iteratively one of the transformations to the new γ part and summing up the β parts one eventually produces a pair (β, γ) such that $\alpha = \Delta\beta + \gamma$, where $\beta, \gamma \in \mathcal{R}_{[g]}$, and where in particular γ is a shift of

$$\sum_{l \in \mathbb{Z}} \frac{f_l}{g^{a_l}} = \frac{f}{g^a} \quad \text{with} \quad a \leq \max\{a_l \; ; \; l \in \mathbb{Z}\}$$

Note that computing such (β, γ) needs the application of at most $k-1$ local transformations, if k is the number of nonzero summands in the canonical decomposition of α. The pair (β, γ) is a solution of the local summation problem, as the following consideration shows.

If γ vanishes, then α is summable.

If γ does not vanish, then consider any two decompositions

$$\alpha = \Delta\beta^{(1)} + \gamma^{(1)} = \Delta\beta^{(2)} + \gamma^{(2)}$$

where $\alpha, \beta^{(i)}, \gamma^{(i)} \in \mathcal{R}_{[g]}$, $(i = 1, 2)$ and where the $\gamma^{(i)}$ cannot be reduced any further by local transformations. Recall that solutions of the rational summation problem are of that kind. Assume that

$$\gamma^{(1)} = E^i \frac{f_1}{g^a} \quad , \quad \gamma^{(2)} = E^j \frac{f_2}{g^b}$$

for some i, j, f_1, f_2, a, b. Hence

$$\Delta(\beta^{(1)} - \beta^{(2)}) = \gamma^{(2)} - \gamma^{(1)} = E^j \frac{f_2}{g^b} - E^i \frac{f_1}{g^a}$$

If $i = j$, then the r.h.s is of the form $E^i \left(f_2/g^b - f_1/g^a\right) = E^i(f/g^c)$, which — by Lemma 1.2 — is possible only if the r.h.s. vanishes, i.e., if $a = b$ and $f_1 = f_2$.
If $i \neq j$, then one local transformation step leads to a similar situation, and by the same argument one concludes that $a = b$ and $f_1 = f_2$.

We point out explicitly that, at each step, one may apply any of the two transformations $(T_{L\to R})$ and $(T_{R\to L})$. Different choices will produce, however, different solutions, i.e., the denominator polynomial of the γ part will have the form $E^j g^a$ for different j's, while the exponent a is independent of the shift m.

In order to make things clear, let us assume that we have a solution (β, γ) with $\gamma = f/g^a$. Then for any $j > 0$ in $\mathbb{Z}$ we have, by $(T_{L\to R})$, that

$$\frac{f}{g^a} = E^j \frac{f}{g^a} - \Delta(E^{-j} + \ldots + E^{-1}) E^j \frac{f}{g^a} \tag{1.2}$$

This means that another solution of the problem would be

$$\alpha = \Delta\left(\beta - (E^{-j} + \ldots + E^{-1}) E^j \frac{f}{g^a}\right) + E^j \frac{f}{g^a}$$

and analogously for $j < 0$ applying $(T_{R\to L})$.

In Figure 1.5 we give a pictorial example for such a transformation.

This corresponds to choosing a different representative $E^i g$ in the shift class $[g]$. As a consequence:

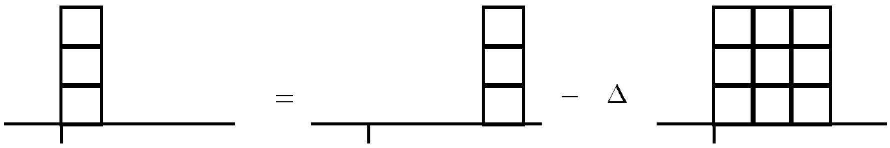

Figure 1.5: Transformation of the *gamma* part of a solution

Proposition 1.2 *Let $\alpha \in \mathcal{R}_{[g]}$, then either $\alpha \in \Delta\mathcal{R}_{[g]}$ (i.e., α is summable), or else there exist unique $\beta \in \mathcal{R}_{[g]}$, f and a such that*

$$\alpha = \Delta\beta + \frac{f}{g^a}$$

where $\gamma = f/g^a$ is reduced. Furthermore, a is bound by the highest exponent arising in the spectrum $\langle \alpha, g \rangle$.

The uniqueness of the β part follows from the injectivity of Δ on $\mathcal{R}$.

In the following section the structure of the β part in dependence of the chosen representative will become clear.

1.4.3 The structure of the β part

Again, consider a monic irreducible g and an arbitrary $\alpha \in \mathcal{R}_{[g]}$ such that

$$\alpha = \sum_{i \in \mathbb{Z}} \alpha_{E^i g} = \sum_{i \in \mathbb{Z}} E^i \frac{f_i}{g^{a_i}}$$

where all fractions are in reduced form. Applying local transformations in a similar way as in (1.2) one can bring each $E^i f_i / g^{a_i}$ into the form $\Delta\beta_i + f_i/g^{a_i}$. More precisely, one may apply $(T_{L\to R})$ if $i < 0$ and $(T_{R\to L})$ if $i > 0$.

Then the rational part β of the solution (β, γ) consists of the sum of all β_i. In terms of spectrum, each β_i has the shape of a *block*, as depicted in Figure 1.4, from the i-th position to the first, if $i < 0$, or from the origin to the $(i-1)$-th, if $i > 0$. Since the sum behaves on the spectrum like the coordinate-wise maximum (up to cancelation due to the numerators), the structure of the denominator of β is easy to describe.

More formally, we have:

$$\alpha \quad = \quad \sum_{i<0} E^i \frac{f_i}{g^{a_i}} + \frac{f_0}{g^{a_0}} + \sum_{j>0} E^j \frac{f_j}{g^{a_j}}$$

$$
\begin{aligned}
&= \sum_{i<0}\left(\frac{f_i}{g^{a_i}} - \Delta\sum_{s=i}^{-1} E^s\,\frac{f_i}{g^{a_i}}\right) + \frac{f_0}{g^{a_0}} + \sum_{j>0}\left(\frac{f_j}{g^{a_j}} + \Delta\sum_{t=0}^{j-1} E^t\,\frac{f_j}{g^{a_j}}\right)\\
&= \Delta\beta + \gamma
\end{aligned}
$$

where

$$
\beta \;=\; -\sum_{s<0} E^s \sum_{i\leq s}\frac{f_i}{g^{a_i}} + \sum_{t\geq 0} E^t \sum_{j>t}\frac{f_j}{g^{a_j}} \tag{1.3}
$$

$$
\gamma \;=\; \sum_{i\in\mathbb{Z}}\frac{f_i}{g^{a_i}} \tag{1.4}
$$

If we write $\mathbf{a} = (a_i)_{i\in\mathbb{Z}} = \langle\alpha, g\rangle$ and $\mathbf{b} = (b_i)_{i\in\mathbb{Z}} = \langle\beta, g\rangle$ then it follows that the s-th exponent b_s is bound by $(\overrightarrow{\mu}\mathbf{a})_s$ if $s < 0$ and by $(\epsilon^{-1}\overleftarrow{\mu}\mathbf{a})_s$ if $s \geq 0$. To determine where equality holds, consider the s-th component of β, i.e.,

$$
E^s \sum_{i\leq s}\frac{f_i}{g^{a_i}}
$$

where we assume $s < 0$, since the case $s \geq 0$ is completely analogous.

Then cancelation, this means $b_s < (\overrightarrow{\mu}\mathbf{a})_s$, may happen only if the maximal exponent $(\overrightarrow{\mu}\mathbf{a})_s$ arises more than once in the sum.

In particular, this can not happen if there is a *jump* in the spectrum at position s, i.e., $a_s = (\overrightarrow{\mu}\mathbf{a})_s$ and $(\delta\overrightarrow{\mu}\mathbf{a})_s > 0$, and in such cases we have $b_s = (\overrightarrow{\mu}\mathbf{a})_s$.

Summarizing, we have

$$
b_s \text{ is } \left\{
\begin{array}{ll}
\leq (\overrightarrow{\mu}\mathbf{a})_s & \text{for } s<0 \text{ in general}\\
= (\overrightarrow{\mu}\mathbf{a})_s & \text{for } s<0 \text{ with } |\{i\ ;\ i\leq s, a_i = (\overrightarrow{\mu}\mathbf{a})_s\}| = 1 \text{ in particular}\\
\leq (\epsilon^{-1}\overleftarrow{\mu}\mathbf{a})_s & \text{for } s\geq 0 \text{ in general}\\
= (\epsilon^{-1}\overleftarrow{\mu}\mathbf{a})_s & \text{for } s\geq 0 \text{ such that } |\{i\ ;\ i\geq s, a_i = (\epsilon^{-1}\overrightarrow{\mu}\mathbf{a})_s\}| = 1 \text{ in particular}
\end{array}\right.
$$

This reasoning can be extended without effort to the case where we take a different representative $E^j g$ of the class $[g]$. The spectrum of the denominator polynomials of β and γ for different representatives $E^j g$ is described in the following:

Proposition 1.3 *Given α as above and $j \in \mathbb{Z}$. Then in general there is a solution (β, γ) of the rational summation problem with*

$$
\gamma = E^j \sum_{i\in\mathbb{Z}}\frac{f_i}{g^{a_i}}
$$

and the denominator polynomial of β is a divisor of $g^{\hat{\mathbf{a}}} = \prod_{i\in\mathbb{Z}} E^i g^{\hat{\mathbf{a}}_i}$, where

$$\hat{a}_s = \begin{cases} (\overrightarrow{\mu}\,\mathbf{a})_s & \text{for } s < j \\ (\epsilon^{-1}\overleftarrow{\mu}\,\mathbf{a})_s & \text{for } s \geq j \end{cases}$$

Furthermore we have

$$\overrightarrow{\mu}\,\mathbf{a} \wedge \epsilon^{-1}\overleftarrow{\mu}\,\mathbf{a} \leq \hat{\mathbf{a}}$$

with equality holding if and only if

$$\min_s \{\ a_s = \max\langle \alpha, g\rangle\ \} \leq j \leq \max_s \{\ a_s = \max\langle \alpha, g\rangle\ \} \qquad (*)$$

Since $\overrightarrow{\mu}\,\mathbf{a} \wedge \epsilon^{-1}\overleftarrow{\mu}\,\mathbf{a}$ does not depend on j, it can be seen as a lower bound for the spectrum of β, up to cancelations due to the numerator of α. So, a "natural" Ansatz for the denominator of β is thus the polynomial

$$g^{\overrightarrow{\mu}\,\mathbf{a} \wedge \epsilon^{-1}\overleftarrow{\mu}\,\mathbf{a}}$$

We will show that this candidate is "optimal" in the sense that no smaller guess can be done without considering the possible cancelations due to the numerator. In other words, one can always find a different numerator polynomial such that no cancelation occurs and $g^{\overrightarrow{\mu}\,\mathbf{a} \wedge \epsilon^{-1}\overleftarrow{\mu}\,\mathbf{a}}$ is indeed the true denominator of β.

The structure of the spectrum of β for a specified γ part is made clear by an example in Figure 1.6. There we describe the result for two choices of the representative polynomial in the shift equivalence class $[g]$, denoted by an arrow in the picture on the left part. Assume that we want a γ with a denominator polynomial of the form $E^j g^c$. Then we *fill up* with "multiplicity boxes" the gaps on the left side of the j-th position in order to get the left-to-right maximum. Analogously, on the right side boxes are added in order to fill up to the right-to-left maximum. Then the j-th stack, which gives the contribution to the denominator polynomial of γ, is deleted and the right part of the structure is shifted by one to the left.

The spectrum of β is then shown in the out-most right part of Figure 1.6.

1.4.4 Optimality

We now show that the "Ansatz" suggested above for the denominator polynomial of β is optimal in the following sense:

Proposition 1.4 *Given $\alpha = p/q \in \mathcal{R}$, the Ansatz for the denominator polynomial of β in a solution of the rational summation problem given by*

$$\prod_g g^{\overrightarrow{\mu}\langle \alpha, g\rangle \wedge \epsilon^{-1}\overleftarrow{\mu}\langle \alpha, g\rangle} \qquad (1.5)$$

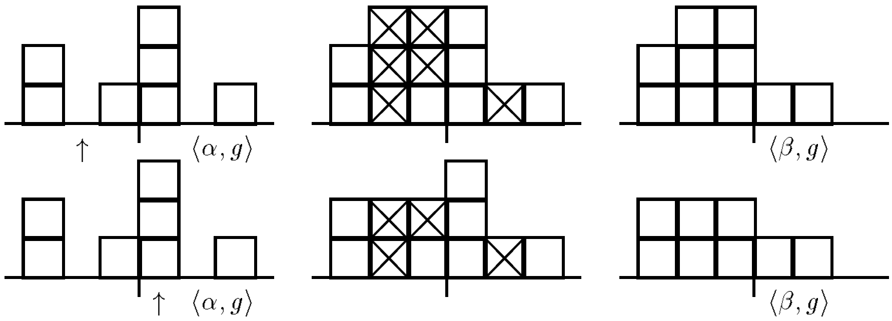

Figure 1.6: Examples of the structure of $\langle\beta, g\rangle$ by varying γ

is **optimal** *in the sense that it is the smallest possible candidate without taking into account the numerator polynomial of α. In other words, it is the smallest "Ansatz" which is suitable for all rational functions in $\mathcal{R}$ with denominator polynomial q.*

Proof. Since we know from Proposition 1.3 that the polynomial in (1.5) is a lower bound for the denominator polynomial of any β part of a solution, we only need to prove that it is a *sharp* bound. This means, it is sufficient to prove that there exists a polynomial $\tilde{p}$ such that the β part of a solution of the summation problem for $\tilde{\alpha} = \tilde{p}/q$ has precisely the denominator proposed above (and not a proper divisor of it).

By localization, we may restrict our attention to a single shift-equivalence class $[g]$. Let us construct such an $\tilde{\alpha} \in \mathcal{R}_{[g]}, \tilde{\alpha} \neq 0$, written as

$$\tilde{\alpha} = \sum_{i\in\mathbb{Z}} E^i \, \frac{f_i}{g^{a_i}}$$

where the sequence $\mathbf{a} = \langle\tilde{\alpha}, g\rangle$ satisfies property $(*)$ above for $j = 0$, i.e., a proper choice of the representative $g \in [g]$ has been made. Consider $\mathbf{a}$ as fixed, and the sequence $(f_i)_{i\in\mathbb{Z}}$ yet to be determined.

Let now $\mathbf{b} = \overrightarrow{\mu}\,\mathbf{a} \wedge \epsilon^{-1}\overleftarrow{\mu}\,\mathbf{a}$ (we assume $\mathbf{b} \neq \mathbf{0}$, otherwise the summation problem for α would have the trivial solution $\beta = 0$,$\gamma = \alpha$).

From property $(*)$ we know

$$b_s = \begin{cases} (\overrightarrow{\mu}\,\mathbf{a})_s & \text{if } s < 0 \\ (\epsilon^{-1}\,\overleftarrow{\mu}\,\mathbf{a})_s & \text{if } s \geq 0 \end{cases}$$

Consider now the sequence of polynomials

$$f_i = \begin{cases} 1 & \text{if } i = k \\ 1 - g^{(\delta\,\mathbf{b})_i} & \text{if } k < i < 0 \end{cases}$$

where $k = \min_s \{ a_s \neq 0 \} = \min_s \{ b_s \neq 0 \}$.

It is easy to check that these polynomials satisfy the system of equations

$$\sum_{k \leq i \leq s} f_i \cdot g^{b_s - a_i} = 1 \qquad (k \leq s < 0)$$

Similarly, the family of polynomials

$$f_j = \begin{cases} 1 & \text{if } j = l \\ 1 - g^{-(\delta\, \mathbf{b})_j} & \text{if } 0 \leq j < l \end{cases}$$

where $l = \max_t \{ a_t \neq 0 \} = \max_t \{ b_t \neq 0 \} + 1$, satisfies the system of equations

$$\sum_{t < j \leq l} f_j \cdot g^{b_t - a_j} = 1 \qquad (0 \leq t < l)$$

We put $f_i = 0$ if $i < k$ or $i > l$. Note that for all $i \in \mathbb{Z}$ f_i is prime to g^{a_i} and $\deg f_i < \deg g^{a_i}$, so that

$$\tilde{\alpha} = -\sum_{i<0} E^i \frac{f_i}{g^{a_i}} + \sum_{j \geq 0} E^j \frac{f_j}{g^{a_j}} \in \mathcal{R}$$

and is written in reduced form. Comparison with (1.3) shows that the β-part of $\tilde{\alpha} = \Delta\beta + \gamma$ satisfies

$$\beta = \sum_{k \leq s < l} E^s \frac{1}{g^{b_s}}$$

and thus $\langle \beta, g \rangle = \mathbf{b}$. In other words: $g^{\mathbf{b}}$ is the true denominator of β.

∎

1.4.5 Concluding remarks

Summarizing, the rational summation problem for α has a solution (β, γ) such that for each shift-equivalence class $[g]$ with $\alpha_{[g]} \neq 0$ we have, by proposition 1.3:

- The denominator of $\beta_{[g]}$ divides $g^{\overrightarrow{\mu}\langle \alpha, g \rangle \wedge \epsilon^{-1} \overleftarrow{\mu}\langle \alpha, g \rangle}$

- The denominator of $\gamma_{[g]}$ divides $g^{\max\langle \alpha, g \rangle}$

provided appropriate representatives $g \in [g]$ have been chosen (i.e., representatives satisfying $(*)$ for $j = 0$).

With this information an algorithm for computing (β, γ) may thus proceed as follows, following the Hermite-Ostrogradski strategy:

- Given $\alpha \in \mathcal{R}$, compute polynomials u, v:

$$u = \prod_g g^{\overrightarrow{\mu}\langle\alpha, g\rangle \wedge \epsilon^{-1}\overleftarrow{\mu}\langle\alpha, g\rangle}$$
$$v = \prod_g g^{\max\langle\alpha, g\rangle}$$

  where the products run over an appropriately chosen system of representatives for the shift-equivalence classes of monic irreducible polynomials (respecting $(*)$ for $j = 0$).

- Put $\beta = a/u$, $\gamma = b/v$, where a, b are polynomials with $\deg a < \deg u$, $\deg b < \deg v$ with indeterminate coefficients.

- Determine the solution numerators a and b by solving

$$\alpha = \frac{Ea}{Eu} - \frac{a}{u} + \frac{b}{v}$$

Note that the solution $(a/u, b/v)$ will not necessarily be reduced, but in general this "Ansatz" is the optimum one can do (in keeping u and v as "small" as possible) without taking properties of the numerator of α into account.

In Subsection 1.6.4 we present an algorithm, which computes a solution (β, γ) following this method. The computation of the polynomials u and v is based on the Gosper-Petkovšek representation of rational functions, as V. Strehl suggested. This relationship is described in the next section.

We finally remark that our results about the denominator polynomial v can be rephrased using the notion of dispersion, as introduced by Abramov in [Abr75]:

Definition 1.4 *The* dispersion $\mathrm{dis}(q)$ *of a polynomial q is defined by*

$$\mathrm{dis}(q) := \max\{h \in \mathbb{Z}; \deg(\gcd(q, E^h q)) > 0\}$$

Clearly: v (as above) and all the variants, obtained by shifting the contributions from the shift-equivalence classes freely (see Section 1.4.2) are polynomials with dispersion zero.

As a consequence, as has been remarked earlier in [Abr75], solutions of the rational summation problem can be characterized as follows:

Proposition 1.5 *Let $\alpha \in \mathcal{R}$, then $(\beta, \gamma) \in \mathcal{R}^2$ is a solution of the rational summation problem for α if and only if*

$$\alpha = \Delta\beta + \gamma \qquad \textit{and} \qquad \mathrm{dis}(\mathit{denom}(\gamma)) = 0$$

The existence of a solution (β, γ) follows from Section 1.4.2, where it was also shown how two distinct solutions are related. The latter question has first been answered by Paule in [Pau95].

1.5 The Gosper-Petkovšek representation of rational functions

The following representation of rational functions is at the basis of Gosper's classical decision method for indefinite hypergeometric summation:

Proposition 1.6 *For any rational function $\alpha \in \mathbb{K}(x)$ there are polynomials $p, q, r \in \mathbb{K}[x]$ such that*

$$\alpha = \frac{E\,p}{p} \cdot \frac{q}{E\,r} \quad \textit{with} \quad \gcd(q, E^i r) = 1 \quad \textit{for all } i \geq 1$$

Petkovšek showed in [Pet92] that a presentation of $\alpha \in \mathbb{K}(x)$ as

$$\alpha = c \cdot \frac{E\,p}{p} \cdot \frac{q}{E\,r} \quad \text{with} \quad \gcd(q, E^i r) = 1 \quad \text{for all } i \geq 1 \quad , \quad \gcd(p, r) = 1 = \gcd(p, q)$$

with monic polynomials $p, q, r \in \mathbb{K}[x]$ and $c \in k$ is *unique*. In the following we will refer to this as the Gosper-Petkovšek (in short, GP) representation of rational functions.

Note that the usual algorithms for computing p, q, r (and c), as outlined in [Gos78] and [Pet92], are based on resultant- and gcd-computations, together with a search for integer zeros of polynomials. In the following, we will look at this representation from two different point of views. The first one, which goes back to V. Strehl, is related to the decomposition of rational functions according to shift-equivalence classes of irreducible polynomials. This is not meant as an *algorithmic* approach, but it gives a *combinatorial* view of this classical result which turns out to be useful for the rational summation problem. The second one motivates the use of the Gosper-Petkovšek representation for the rational summation problem using the concept of GFF (*Greatest Factorial Factorization*), introduced by Paule in [Pau95].

1.5.1 Combinatorial Gosper-Petkovšek representation

Note first that the Gosper-Petkovšek representation localizes perfectly (because it is a purely multiplicative statement). We may split α into a product $\alpha^{[g_1]} \cdots \alpha^{[g_k]}$, where the g_i are irreducible polynomials belonging to distinct shift-equivalence classes, and where each factor $\alpha^{[g_i]}$ accounts for the contribution of factors from the class of g_i to α. If we have a (unique) local representation

$$\alpha^{[g_i]} = \frac{E\,p_i}{p_i} \cdot \frac{q_i}{E\,r_i}$$

with the appropriate gcd-conditions satisfied, then the (unique) Gosper-Petkovšek-triple (p, q, r) for α results from multiplication*:

$$p = p_1 \cdots p_k \quad , \quad q = q_1 \cdots q_k \;,\; r = r_1 \cdots r_k$$

*The role of the scalar factor c in Petkovšek's assertion is irrelevant for our purpose.

If we look now at the local situation for any irreducible polynomial g, then $\alpha^{[g]}$ may be represented by a doubly infinite sequence of integers

$$(a_i)_{i\in\mathbb{Z}} := \langle f, g\rangle - \langle h, g\rangle \quad \text{for } \alpha = f/h$$

where "$-$" is the component-wise difference of sequences.

Let us say that an integer sequence $(a_i)_{i\in\mathbb{Z}}$ is a *rational sequence* if there are only finitely many nonzero terms. If all terms are nonnegative, and again only finitely many different from 0, then it is a *polynomial sequence.* Adopting this terminology, the Gosper-Petkovšek representation boils down to the following combinatorial assertion, where δ and ϵ are operators on sequences as before and $\mathbf{0}$ is the all-zero sequence.

This combinatorial equivalent of the Gosper-Petkovšek representation is illustrated by an example below.

Proposition 1.7 (Combinatorial Gosper-Petkovšek representation) *Let* $\mathbf{a}$ *be any rational sequence. Then there are unique polynomial sequences* $\mathbf{p}$, $\mathbf{q}$, *and* $\mathbf{r}$ *such that*

$$\mathbf{a} = -\delta\,\mathbf{p} + \mathbf{q} - \epsilon\,\mathbf{r}$$

where

$$\mathbf{p}\wedge\mathbf{q} = \mathbf{0}\ ,\ \ \mathbf{p}\wedge\mathbf{r} = \mathbf{0}\ ,\ \ \mathbf{q}\wedge\epsilon^j\,\mathbf{r} = \mathbf{0} \textit{ for all } j \geq 1$$

In addition, if $\mathbf{f} = \sigma\,\mathbf{a}$ *then* $\mathbf{p}, \mathbf{q}$ *and* $\mathbf{r}$ *are determined by*

$$\mathbf{q} = \delta\,\overrightarrow{\mu}\,\mathbf{f}\ ,\ \ \epsilon\,\mathbf{r} = -\delta\,\overleftarrow{\mu}\,\mathbf{f}\ ,\ \ \mathbf{p} = (\overrightarrow{\mu}\,\mathbf{f}\wedge\overleftarrow{\mu}\,\mathbf{f}) - \mathbf{f}$$

Proof. We define $\mathbf{f} := \sigma\,\mathbf{a}$ and

$$\mathbf{q} := \delta\,\overrightarrow{\mu}\,\mathbf{f}\ ,\ \ \epsilon\,\mathbf{r} := -\delta\,\overleftarrow{\mu}\,\mathbf{f}$$

We then have $\mathbf{q} \geq \mathbf{0}$, since $\overrightarrow{\mu}\,\mathbf{f}$ is non-decreasing, and $\mathbf{q}$ is a polynomial sequence, since $(\delta\,\overrightarrow{\mu}\,\mathbf{f})_i > 0$ implies $(\delta\,\mathbf{f})_i = a_i > 0$.
Similarly, $\mathbf{r} \geq \mathbf{0}$, since $\overleftarrow{\mu}\,\mathbf{f}$ in non-increasing, and $\mathbf{r}$ is a polynomial sequence since $(\delta\,\overleftarrow{\mu}\,\mathbf{f})_i < 0$ implies $(\delta\,\mathbf{f})_i = a_i < 0$.
Now obviously

$$(\delta\,\overrightarrow{\mu}\,\mathbf{f})_i > 0 \text{ and } (\delta\,\overleftarrow{\mu}\,\mathbf{f})_j < 0 \text{ implies } i < j$$

so that $\mathbf{q}\wedge\epsilon^j\,\mathbf{r} = \mathbf{0}$ holds for all $j \geq 1$.

Consider now

$$\mathbf{p} := \overrightarrow{\mu}\,\mathbf{f} + \overleftarrow{\mu}\,\mathbf{f} - \mathbf{f} - \mathbf{m_f} = (\overrightarrow{\mu} - id)\,\mathbf{f}\wedge(\overleftarrow{\mu} - id)\,\mathbf{f} = (\overrightarrow{\mu}\,\mathbf{f}\wedge\overleftarrow{\mu}\,\mathbf{f}) - \mathbf{f}$$

where $\mathbf{m_f}$ denotes the sequence which has value $m_{\mathbf{f}} := \max_{i\in\mathbb{Z}} f_i$ everywhere (note that this value is well-defined since $\delta\,\mathbf{f} = \mathbf{a}$ is a rational sequence). Again, it is easy to check that $\mathbf{p}$ is a polynomial sequence, and we have in addition

$$\delta\,\mathbf{p} = \delta\,\overrightarrow{\mu}\,\mathbf{f} + \delta\,\overleftarrow{\mu}\,\mathbf{f} - \delta\,\mathbf{f} = \mathbf{q} - \epsilon\,\mathbf{r} - \mathbf{a}$$

as desired.

We now have to show that $\mathbf{p}\wedge\mathbf{q} = \mathbf{0}$ and $\mathbf{p}\wedge\mathbf{r} = \mathbf{0}$ hold. Note that

- if $q_i = (\delta \overrightarrow{\mu}\, \mathbf{f})_i \neq 0$, then necessarily $f_i = (\overrightarrow{\mu}\, \mathbf{f})_i$ and thus $p_i = (\overleftarrow{\mu}\, \mathbf{f})_i - m_{\mathbf{f}} \leq 0$, which means $p_i = 0$, since $\mathbf{p}$ is a polynomial sequence.
- if $r_j = -(\delta \overleftarrow{\mu}\, \mathbf{f})_{j+1} \neq 0$, then necessarily $f_j = (\overleftarrow{\mu}\, \mathbf{f})_j$ and thus $p_j = (\overrightarrow{\mu}\, \mathbf{f})_j - m_{\mathbf{f}} \leq 0$, which means $p_j = 0$, since $\mathbf{p}$ is a polynomial sequence.

So far the *existence* part of the proposition has been established. For the uniqueness part, let $\mathbf{p}, \mathbf{q}, \mathbf{r}$ be any polynomial sequences such that the assertion of the lemma holds. We will show that these are identical with the corresponding sequences defined above.

Let $i_0 \in \mathbb{Z}$ be an index such that both

$$(\sigma\, \mathbf{q})_{i_0} = m_{\sigma\, \mathbf{q}} \quad \text{and} \quad (\sigma\, \epsilon\, \mathbf{r})_{i_0} = 0$$

hold. Note that the condition: "$\mathbf{q} \wedge \epsilon^j \mathbf{r} = \mathbf{0}$ for all $j \geq 1$" guarantees the existence of such an index. We then have

$$(\sigma\, \epsilon\, \mathbf{r})_i = 0 \;\text{ for all } i \leq i_0 \quad , \quad (\sigma\, \mathbf{q})_j = m_{\sigma\, \mathbf{q}} \;\text{ for all } j \geq i_0$$

For $i \leq i_0$ we now have

$$f_i = (\sigma\, \mathbf{a})_i = -p_i + (\sigma\, \mathbf{q})_i$$

hence $f_i \leq (\sigma\, \mathbf{q})_i$, and the "orthogonality" of $\mathbf{p}$ and $\mathbf{q}$ implies that $f_i = (\sigma\, \mathbf{q})_i$ whenever $q_i \neq 0$ holds. Both facts together imply

$$(\overrightarrow{\mu}\, \mathbf{f})_i = (\sigma\, \mathbf{q})_i \;\text{ for all } i \leq i_0$$

Similarly, for $j \geq i_0$ we have

$$f_j = (\sigma\, \mathbf{a})_j = -p_j + m_{\sigma\, \mathbf{q}} - (\sigma\, \epsilon\, \mathbf{r})_j$$

hence $f_j \leq m_{\sigma\, \mathbf{q}} - (\sigma\, \epsilon\, \mathbf{r})_j$, and here the "orthogonality" of $\mathbf{p}$ and $\mathbf{r}$ implies $f_j = m_{\sigma\, \mathbf{q}} - (\sigma\, \epsilon\, \mathbf{r})_j$ whenever $r_j \neq 0$. Here these two facts imply

$$(\overleftarrow{\mu}\, \mathbf{f})_j = m_{\sigma\, \mathbf{q}} - (\sigma\, \epsilon\, \mathbf{r})_j \;\text{ for all } j \geq i_0$$

We conclude, in particular, that $f_{i_0} = m_{\sigma\, \mathbf{q}}$, hence $m_{\mathbf{f}} = m_{\sigma\, \mathbf{q}}$, and consequently

$$\overrightarrow{\mu}\, \mathbf{f} = \sigma\, \mathbf{q} \quad \text{and} \quad \overleftarrow{\mu}\, \mathbf{f} = \mathbf{m_f} - \sigma\, \epsilon\, \mathbf{r}$$

which implies

$$\mathbf{q} = \delta \overrightarrow{\mu}\, \mathbf{f} \quad \text{and} \quad \epsilon\, \mathbf{r} = -\delta \overleftarrow{\mu}\, \mathbf{f}$$

Finally

$$\delta\, \mathbf{p} = -\delta\, \mathbf{f} + \mathbf{q} - \epsilon\, \mathbf{r} = \delta\, (-\mathbf{f} + \overrightarrow{\mu}\, \mathbf{f} + \overleftarrow{\mu}\, \mathbf{f})$$

and

$$\mathbf{p} = -\mathbf{f} + \overrightarrow{\mu}\, \mathbf{f} + \overleftarrow{\mu}\, \mathbf{f} - \mathbf{m_f}$$

follows. ∎

An Example

As an example, we determine the Gosper-Petkovšek representation of the rational function α given by

$$\alpha = \frac{(x-3)(x-2)^2(x+2)(x+5)^2}{(x-4)(x+1)^3(x+3)^2}$$

The rational sequence $\mathbf{a}$ associated to α with respect to the irreducible polynomial $g = x$ is

$$\mathbf{a} = \langle (x-3)(x-2)^2(x+2)(x+5)^2, x\rangle - \langle (x-4)(x+1)^3(x+3)^2, x\rangle$$

According to the proposition, we compute the following sequences, where the position of the index 0 is indicated by underlining.

$$\begin{array}{rcccccccccccccc}
\mathbf{a} & = & \ldots & 0 & -1 & 1 & 2 & 0 & \underline{0} & -3 & 1 & -2 & 0 & 2 & 0 & \ldots \\
\mathbf{f} = \sigma\mathbf{a} & = & \ldots & 0 & -1 & 0 & 2 & 2 & \underline{2} & -1 & 0 & -2 & -2 & 0 & 0 & \ldots \\
\overrightarrow{\mu}\mathbf{f} & = & \ldots & 0 & 0 & 0 & 2 & 2 & \underline{2} & 2 & 2 & 2 & 2 & 2 & 2 & \ldots \\
\mathbf{q} = \delta\overrightarrow{\mu}\mathbf{f} & = & \ldots & 0 & 0 & 0 & 2 & 0 & \underline{0} & 0 & 0 & 0 & 0 & 0 & 0 & \ldots \\
\overleftarrow{\mu}\mathbf{f} & = & \ldots & 2 & 2 & 2 & 2 & 2 & \underline{2} & 0 & 0 & 0 & 0 & 0 & 0 & \ldots \\
\epsilon\mathbf{r} = -\delta\overleftarrow{\mu}\mathbf{f} & = & \ldots & 0 & 0 & 0 & 0 & 0 & \underline{0} & 2 & 0 & 0 & 0 & 0 & 0 & \ldots \\
(\overrightarrow{\mu} - id)\mathbf{f} & = & \ldots & 0 & 1 & 0 & 0 & 0 & \underline{0} & 3 & 2 & 4 & 4 & 2 & 2 & \ldots \\
(\overleftarrow{\mu} - id)\mathbf{f} & = & \ldots & 2 & 3 & 2 & 0 & 0 & \underline{0} & 1 & 0 & 2 & 2 & 0 & 0 & \ldots \\
\mathbf{p} & = & \ldots & 0 & 1 & 0 & 0 & 0 & \underline{0} & 1 & 0 & 2 & 2 & 0 & 0 & \ldots
\end{array}$$

From this it follows that the Gosper-Petkovšek representation of α is given by

$$p = (x-4)(x+1)(x+3)^2(x+4)^2, \quad q = (x-2)^2, \quad r = x^2$$

One easily verifies that

$$\frac{Ep}{p} \cdot \frac{q}{Er} = \frac{(x-3)(x+2)(x+4)^2(x+5)^2}{(x-4)(x+1)(x+3)^2(x+4)^2} \cdot \frac{(x-2)^2}{(x+1)^2} = \alpha$$

and that $\gcd(q, E^i r) = 1$ for all $i \geq 1$ and $\gcd(p,r) = 1 = \gcd(p,q)$.

1.5.2 Relevance for Rational Summation

The proof of the proposition shows that the Gosper-Petkovšek representation of rational functions provides an algorithmic way to compute the polynomials u and v from Section 1.4.5.

Let $s/t \in \mathcal{R}$ be in reduced form and g any irreducible polynomial. We apply Proposition 1.7 to the rational function $\alpha_{[g]} = t_{[g]}/Et_{[g]}$. Let $\mathbf{t} := \langle t, g\rangle$, then we have

$\mathbf{a} = \langle t_{[g]}, g\rangle - \langle Et_{[g]}, g\rangle = \mathbf{t} - \epsilon\,\mathbf{t}$ and this implies $\mathbf{f} = \sigma\,\mathbf{a} = \mathbf{t}$. Consider now the $\mathbf{p}$, $\mathbf{q}$ and $\mathbf{r}$ from the Gosper-Petkovšek representation of $\alpha_{[g]}$ as in Proposition 1.7 and the corresponding polynomials p, q, r. Since $\mathbf{f} = \mathbf{t}$ and $\mathbf{p} = -\mathbf{t} + \overrightarrow{\mu}\,\mathbf{t} + \overleftarrow{\mu}\,\mathbf{t} - \mathbf{m_t}$, we have $\langle p\cdot t, g\rangle = \overrightarrow{\mu}\,\mathbf{t} + \overleftarrow{\mu}\,\mathbf{t} - \mathbf{m_t}$ and this is $\overrightarrow{\mu}\,\mathbf{t} \wedge \overleftarrow{\mu}\,\mathbf{t}$, the common part of the right-to-left and the left-to-right maximum of $\mathbf{t}$. This is not yet what we need for the Ansatz, viz., $\overrightarrow{\mu}\,\mathbf{t} \wedge \epsilon^{-1}\,\overleftarrow{\mu}\,\mathbf{t}$ (cf. Proposition 1.4). In order to arrive there, note that the sequences are the same from $-\infty$ to the position of the first right-maximum of $\mathbf{t}$. To the right of that, say at position i, $\overleftarrow{\mu}\,\mathbf{t}$ contributes exactly $(\overleftarrow{\mu}\,\mathbf{t})_i - (\overleftarrow{\mu}\,\mathbf{t})_{i+1}$ multiplicity-boxes more than $\epsilon^{-1}\,\overleftarrow{\mu}\,\mathbf{t}$, but this is precisely r_i, since $\mathbf{r} = -\epsilon^{-1}\,\delta\,\overleftarrow{\mu}\,\mathbf{t}$. As a consequence, $\mathbf{r}$ is precisely what we have to delete and $\mathbf{p} + \mathbf{t} - \mathbf{r}$ is the spectrum we are looking for. So, the polynomial $p\cdot t/r$ is the requested optimal Ansatz for the denominator of β.

In Figure 1.7 we show an example of this decomposition. On the left part is the spectrum $\mathbf{t}$, while on the right part we draw the spectrum of pt, where the crossed boxes correspond to $\mathbf{r}$.

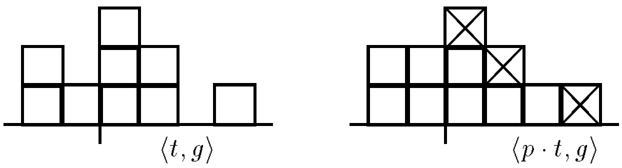

Figure 1.7: Spectrum of t and $p\cdot t$.

In other words,

$$\begin{aligned}
\langle p\cdot t, g\rangle - \langle r, g\rangle &= \mathbf{p} + \mathbf{t} - \mathbf{r} \\
&= -\mathbf{t} + \overrightarrow{\mu}\,\mathbf{t} + \overleftarrow{\mu}\,\mathbf{t} - \mathbf{m_t} + \mathbf{t} + \epsilon^{-1}\,\delta\,\overleftarrow{\mu}\,\mathbf{t} \\
&= \overrightarrow{\mu}\,\mathbf{t} + \overleftarrow{\mu}\,\mathbf{t} - \mathbf{m_t} + \epsilon^{-1}\,\overleftarrow{\mu}\,\mathbf{t} - \overleftarrow{\mu}\,\mathbf{t} \\
&= \overrightarrow{\mu}\,\mathbf{t} + \epsilon^{-1}\,\overleftarrow{\mu}\,\mathbf{t} - \mathbf{m_t} \qquad = \overrightarrow{\mu}\,\mathbf{t} \wedge \epsilon^{-1}\,\overleftarrow{\mu}\,\mathbf{t}
\end{aligned}$$

Since this holds for any irreducible g, for the Gosper-Petkovšek representation (p, q, r) of $\alpha = t/Et$ we have that $u = pt/r$ is *globally* optimal for all shift equivalence classes. On an algorithmic level, this means that the optimum denominator polynomial u can be obtained directly from an algorithm computing the Gosper-Petkovšek representation. In particular, no factorization with respect to shift-equivalence classes is necessary and no explicit choice of a representative for each shift-class is necessary.

Similarly, we show that also the denominator v can be obtained from the Gosper-Petkovšek representation of α. Consider again $\alpha_{[g]}$ as above and let j be the smallest index such that $t_j = m_{\mathbf{t}}$, then from $\mathbf{q} = \delta\,\overrightarrow{\mu}\,\mathbf{t}$ it follows that j is the largest index such that $q_j \neq 0$. Let us define a sequence $\mathbf{q}^+ = (q_i^+)_{i\in\mathbb{Z}}$ by

$$q_i^+ = \begin{cases} 0 & \text{if } i \neq j \\ \sum_{l\leq j} q_l & \text{if } i = j \end{cases}$$

and the polynomial $q_{[g]}^+$ by $q_{[g]}^+ := g^{\mathbf{q}^+}$. Since $\sum_{i\leq j} q_i = m_{\mathbf{t}}$, the denominator $v_{[g]} = q_{[g]}^+$ is optimal with respect to the shift class $[g]$.

From this it follows that the denominator $v = q^+ := \prod_{[g]} q_{[g]}^+$ is optimal with respect to s/t for solving the rational summation problem, in the sense of Subsection 1.4.4.

Summarizing, we have:

Proposition 1.8 *For any $\alpha = s/t \in \mathcal{R}$ an optimal choice of denominators u and v for the β and γ part of a solution of the rational summation problem for α is given by*

$$u = \frac{p\cdot t}{r} \quad \textit{and} \quad v = q^+$$

where (p, q, r) is the Gosper-Petkovšek representation of t/Et.

In Section 1.6.4 we describe the algorithm in more detail.

1.5.3 An algebraic description via GFF

The use of the Gosper-Petkovšek representation of rational functions for the rational summation problem can also be algebraically motivated in the framework presented by Paule in [Pau95] for hypergeometric telescoping.

The fundamental concept introduced by Paule is the *Greatest Factorial Factorization* (GFF) of polynomials, in analogy to the square-free factorization.

Definition 1.5 *We say that the tuple $(p_1, \ldots, p_k)$, $p_i \in \mathbb{K}[x]$, is a* GFF*-form of a monic polynomial $p \in \mathbb{K}[x]$ if the following conditions hold:*
(GFF1) $p = [p_1]^{\underline{1}} \cdots [p_k]^{\underline{k}}$,
(GFF2) each p_i is monic, and $k > 0$ implies $\deg(p_k) > 0$,
(GFF3) $i \leq j \Rightarrow \gcd([p_i]^{\underline{i}}, Ep_j) = 1 = \gcd([p_i]^{\underline{i}}, E^{-j}p_j)$.

Here $[p_i]^{\underline{i}}$ indicates the *i-th falling factorial* of p_i defined as

$$[p_i]^{\underline{i}} := \prod_{k=0}^{i-1} E^{-k} p_i$$

It can be shown that such a GFF-form is uniquely determined by p, and we write $\mathrm{GFF}(p) = (p_1, \ldots, p_k)$.

Intuitively, the GFF-form of a polynomial p gives information about the maximal chains in the shift structure of p. If p_k is the last component of $\mathrm{GFF}(p)$, then $[p_k]^{\underline{k}}$

collects the maximal chains of length k in p. Then $[p_{k-1}]^{\underline{k-1}}$ collects the chains of length $k-1$ in $p/[p_k]^{\underline{k}}$, if there are any, and so on.

Again, the GFF concept localizes. This means, w.l.o.g. we only have to consider factors of t of the form $E^j g^i$ for a monic irreducible g. In Figure 1.8 we show an example for GFF(t), where

$$t = (x-3)(x-2)^2(x+1)^4(x+2)^2(x+3)(x+5)^3(x+6)$$

In each box we indicate the index of the corresponding chain to which the box contributes.

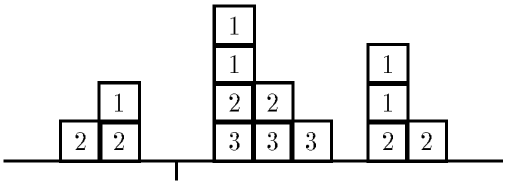

Figure 1.8: Example of GFF(t)

The GFF-form of t can be then read off the picture:

$$t := [(x-2)(x+1)^2(x+5)^2]^{\underline{1}}[(x-1)(x+2)(x+6)]^{\underline{2}}[x+3]^{\underline{3}}$$

Let now f_k be a hypergeometric term, i.e, a term such that the quotient f_{k+1}/f_k can be expressed as rational function in k (note that, in particular, rational functions themselves are hypergeometric). Then Paule, using the GFF concept, shows the well-known fact that a hypergeometric solution g_k of

$$f_k = g_{k+1} - g_k$$

is given by

$$g_k = \frac{r(k)u(k)}{p(k)} \cdot f_k$$

where u is a polynomial solution of $q \cdot Eu - r \cdot u = p$ and (p, q, r) is the Gosper-Petkovšek representation of the quotient f_{k+1}/f_k.

In our context, f_k is a rational function $f_k = \alpha(k)$ for $\alpha = s/t \in \mathcal{R}$ in reduced form. This means that a solution β of a summable $\alpha = \Delta\beta$ has the form

$$\beta = \frac{r \cdot u}{p} \cdot \alpha = \frac{r \cdot u}{p} \cdot \frac{s}{t}$$

Using results from [Pau95] we show that the denominator of this β is precisely the *optimal* denominator we describe in Subsection 1.4.4.

From Section 5.2 in [Pau95] (cf. especially Lemma 5.2 ibid.) we have:

Lemma 1.3 *Let $\langle p, q, r\rangle$ be the GP-representation of the reduced $a/b \in \mathcal{R}$. Then for $GFF(p) = [p_1]^{\underline{1}} \cdots [p_n]^{\underline{n}}$ we have:*

1. $a = (Ep_1) \cdots (Ep_n) \cdot q$
2. $b = (E^0 p_1) \cdots (E^{-n+1} p_n) \cdot Er$
3. $\forall i \in \{1, \ldots, n\} : \gcd([p_i]^{\underline{i}}, q) = 1$
4. $\forall i \in \{1, \ldots, n\} : \gcd([p_i]^{\underline{i}}, r) = 1$

First, we note that the denominator of β depends only on t. We have the following straightforward lemma.

Lemma 1.4 *Let $\alpha = s/t \in \mathcal{R}$ be in reduced form. If (p_t, q_t, r_t) is the GP-representation of t/Et, then $(s \cdot p_t, q_t, r_t)$ is the GP-representation of $E\alpha/\alpha$.*

Proof. Let (p_t, q_t, r_t) be the GP-representation of t/Et. The nontrivial part is to verify that

$$\gcd(sp_t, q_t) = 1 \quad \text{and} \quad \gcd(sp_t, r_t) = 1$$

but this is obvious since from properties 1 and 2 of the last lemma we know that $q_t | t$ and $Er_t | Et$ ∎

Therefore both, (p, q, r) and $(s \cdot p_t, q_t, r_t)$, are GP-representations of $E\alpha/\alpha$. Consequently, by uniqueness,

$$\beta = \frac{r \cdot u}{p} \cdot \frac{s}{t} = \frac{r_t \cdot u}{s \cdot p_t} \cdot \frac{s}{t} = \frac{r_t \cdot u}{p_t \cdot t}$$

and since $r_t | t$, the denominator polynomial of β is a divisor of $p_t t / r_t$, as proposed in Subsection 1.5.2.

Note that this reasoning is still valid for the refined rational summation problem, where we look for a solution (β, γ) of $\alpha = s/t = \Delta\beta + \gamma$. In fact, since we know that the denominator of the γ part corresponding to an optimal choice is a divisor of t (cf. Proposition 1.2), we have that, in general, $(\alpha - \gamma)$ has the same denominator as α. From the fact that the Ansatz for the denominator of β does not depend on the numerator of α, or $(\alpha - \gamma)$, it follows that the Ansatz for the denominator of β stays the same.

We now present an alternative proof of the fact that the spectrum of $p_t t / r_t$ has the form described as optimal in Subsection 1.4.4.

Let $GFF(t) = [t_1]^{\underline{1}} \cdots [t_m]^{\underline{m}}$, then, by the *fundamental Lemma* in [Pau95] the reduced form a/b of t/Et is given by

$$a = \frac{t}{\gcd(t, Et)} = t_1 (E^{-1} t_2) \cdots (E^{-m+1} t_m), \quad b = \frac{Et}{\gcd(t, Et)} = (Et_1)(Et_2) \cdots (Et_m),$$

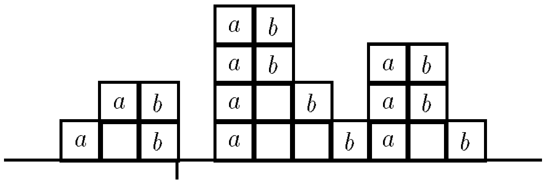

Figure 1.9: Example with respect to $a/b = t/Et$

In other words, a contains the left ends of each component of the GFF, while b gives the (shifted) right ends.

In Figure 1.9 we show the contributions to a and b in the previous example.

The structure of p_t follows then from Lemma 1.3. From now on we follow the notation of Lemma 1.3 for $a/b = t/Et$, so (p, q, r) is the GP-representation of a/b. In the following we describe how to read off p,q and r from the multiplicity-box diagram of t.

Namely from properties 1 and 2 it follows that the chains in $[p_i]^{\underline{i}}$ are, in some sense, *enclosed* between shift components of t. So, all factors in a which do not have a factor of b on the left must contribute to q, since no chain in p can come from that side. In our example $(x-2)(x-1)(x+1)^2|q$. Analogously, all b-boxes which have no a-box at the same level on the graph to the right must contribute to Er, so $(x+2)(x+6)^2(x+7)|Er$. In other words, r consists at least of those factors which give rise to a jump in the right-to-left maximum of $\langle t, g\rangle$. We will see that this indeed exhausts all divisors of q and Er, respectively, in other words equality actually holds.

Furthermore we know that $\gcd(q, E^j r) = 1$ for all $j \geq 1$, so all factors of q must be on the left of each factor of Er. Consider now a b-box which has an a-box on the right, like, for instance, $(x-1)$ in our example. This can not contribute to Er since, otherwise, the corresponding a-box on the right, that is $(x+1)$ in the example, would contribute to q and we would have a contradiction to $\gcd(q, E^j r) = 1$ for all $j \geq 1$. For this reason that b-box must be a left element of a chain in p, which then connects the b-box to the corresponding a-box.

Since all a and b-boxes have to be assigned following these rules, these considerations prove that p consists of all chains joining the shift components at same height of t, i.e. p adds to t what is missing to the gcd of the left-to-right and right-to-left maximum. In our example $q = (x-2)(x-1)(x+1)^2$, $Er = (x+2)(x+6)^2(x+7)$ and

$$p = (x-1)^2 x^2 (x+2)(x+3)^2(x+4)^3$$

This corresponds to filling up the gaps in the spectrum of t, i.e., $\langle t \cdot p, g\rangle = \overrightarrow{\mu}\langle t, g\rangle \wedge \overleftarrow{\mu}\langle t, g\rangle$.

In Figure 1.10 we show the spectrum of $t \cdot p$.

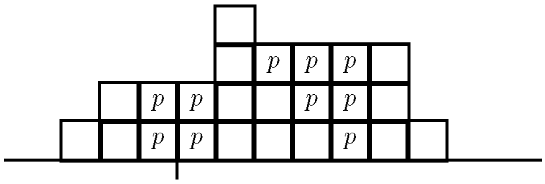

Figure 1.10: Spectrum of $p \cdot t$

From the known structure of r it then follows that

$$\langle \frac{p \cdot t}{r}, g \rangle = \overrightarrow{\mu}\langle t, g \rangle \wedge \varepsilon^{-1} \overleftarrow{\mu}\langle t, g \rangle$$

This gives a different proof of Proposition 1.8, since $p \cdot t/r$ is precisely the optimal denominator polynomial suggested in Subsection 1.4.4.

1.6 The Algorithms

The known algorithms for computing a solution (β, γ) of the rational summation problem are based, in principle, on one of the following methods:

- Local transformations in a similar form like in Subsection 1.4.1 are iteratively applied on the input function, producing a sequence (β_k, γ_k) which eventually yields a solution.
- Candidates u and v for the denominator polynomials of β and γ, resp., are computed. Then the problems reduces to solving a polynomial equation by coefficients comparison, i.e., to solving a system of linear equations.

The algorithms of Moenck and Abramov belong to the first kind, while the remaining ones are based on the second method.

The algorithms are described in more detail in other published works, see the articles cited in each subsection and [Pir95] for a description of an implementation in Maple. So, here we restrict ourselves to an informal presentation using the notation introduced above. Our goal is to work out clearly the structural differences of the various approaches. In particular, we want to stress the fact that only the algorithm based on the observations of Subsection 1.5.2 computes a solution choosing optimal candidates for the denominators of both β and γ.

In order to implement the methods, the field $\mathbb{K}$ should allow effective algorithms for the field operations, and additionally we assume algorithms for finding integer roots of polynomials over $\mathbb{K}$. The latter requirement is for the computation of the dispersion. We discuss in Subsection 1.6.6 a method for doing the computation if one is able to factorize polynomials over $\mathbb{K}$. On the other hand we notice that all algorithms described here work without factorization, using gcd and resultant computations only. All algorithms we discuss here have been implemented in the computer algebra system Maple.

In addition, since all the algorithms need the computation of the dispersion of a polynomial, we dedicate the Subsection 1.6.6 to this goal.

1.6.1 Moenck's algorithm

Moenck in 1977 published an algorithm in [Moe77]. Paule in 1993 noticed in [Pau93] that the Maple function `sum`, whose implementation is based on this algorithm, does not always lead to a correct answer. We describe the method by an example and point out in which way a gap in the description produced the error in the implementation.

Consider the rational function

$$\alpha = \frac{p}{q} = \frac{x^2+3}{x^2(x+2)^3(x+3)^2(x^2+2)(x^2+2x+3)^2} \in \mathbb{Q}(x)$$

The shift structure of the denominator q gives a decomposition into two classes, represented in the left part of Fig. 1.11.

Figure 1.11: Spectrum of $x^2(x+2)^3(x+3)^2(x^2+2)(x^2+2x+3)^2$ and of its saturation

The first step in Moenck's algorithm "fills up the gaps" in each class, i.e., we put as many boxes on each line as we need to get a rectangle on it. This means that we have to take all factors with the maximal multiplicity arising in that class. This way we obtain the new denominator as in the right part of Fig. 1.11. This brings along a new representation for the rational function as follows.

$$\alpha = \frac{x(x+1)^3(x+3)(x^2+2)(x^2+3)}{x^3(x+1)^3(x+2)^3(x+3)^3(x^2+2)^2(x^2+2x+3)^2}$$

Let us consider the two classes separately, i.e.,

$$q_1 = x^3(x+1)^3(x+2)^3(x+3)^3 \qquad \text{and} \qquad q_2 = (x^2+2)^2(x^2+2x+3)^2$$

As we know, it is sufficient to find solutions to the problem for each shift component separately. So, consider now the decomposition $\alpha = p_1/q_1 + p_2/q_2$ (for some p_1, p_2).

Next, we compute a partial fraction decomposition with respect to the first class

$$\frac{p_1}{q_1} = \frac{p_1(x)}{x^3(x+1)^3(x+2)^3(x+3)^3} = \sum_{j=0}^{3} \frac{p_{1,j}}{(x+j)^3(x+j+1)^3\cdots(x+3)^3} \tag{1.6}$$

where $\deg p_{1,j} < \deg(x+3)^3 = 3$. Then one iteratively decomposes each summand into the form $\Delta\,\beta + \gamma$ with a remainder having smaller dispersion after each step.

For that any summand of the right-hand side of equation (1.6) and compute a polynomial solution f, g to the equation

$$(x+j)^3 f + ((x+3)^3 - (x+j)^3)g = p_{1,j}(x)$$

This can be done by the Extended Euclidean Algorithm (the function `gcdex` in Maple), since $(x+j)^3$ and $((x+3)^3 - (x+j)^3)$ are relatively prime. Then one can verify that

$$\frac{p_{1,j}}{(x+j)^3\cdots(x+3)^3} = \Delta\,\frac{-g}{(x+j)^3\cdots(x+2)^3} + \frac{Eg-g+f}{(x+j+1)^3\cdots(x+3)^3}$$

and one obtains a decomposition for the j-th summand. We iterate the procedure till we obtain a remainder with dispersion zero (or zero itself), then we sum up all sub-results finding a decomposition for α.

This method can be seen as discrete analogue to Hermite's algorithm for the integration of rational functions (see [SM92]).

The problem in Moenck's method lies in the computation of the shift saturation of the denominator. In his work the definition of such a *shift saturation* does not make sure that the classes are relatively prime. Using the `printlevel` facility one sees that in Maple the denominator is decomposed into three classes $q_1 = x^2(x+2)^2(x+3)^2$, $q_2 = x+2$ and $q_3 = (x^2+2)^2(x^2+2x+3)^2$. From this it follows that, in this case, the partial fraction decomposition of α can not be computed. As a consequence, if one tries to get a solution by the function `sum` of Maple V.3 one obtains:

```
> sum ( (x^2+3)/(x^2*(x+2)^3*(x+3)^2*(x^2+2)*(x^2+2*x+3)^2),x);
Error, (in gcdex/diophant) wrong number (or type) of arguments
```

Furthermore, Moenck does not give any algorithm to fill up the gaps in the shift structure of the denominator. This has been done by Paule, who needs such a saturation in his algorithm too, see Section 1.6.3.

1.6.2 Abramov's algorithm

The method of Abramov is based on iterated application of a reduction transformation, eventually yielding a solution.

Consider now an arbitrary element $\alpha \in \mathcal{R}_{[g]}$, say

$$\alpha = E^i\,\frac{f_i}{g^{a_i}} + E^{i+1}\,\frac{f_{i+1}}{g^{a_{i+1}}} + \cdots + E^{i+k}\,\frac{f_{i+k}}{g^{a_{i+k}}}$$

for some $i \in \mathbb{Z}$, $k > 0$ and $f_i \neq 0 \neq f_{i+k}$. Then it is easily checked that

$$\alpha = \Delta\left(E^{i+k-1}\frac{f_{i+k}}{g^{a_{i+k}}}\right) + E^i\frac{f_i}{g^{a_i}} + \cdots + E^{i+k-1}\left(\frac{f_{i+k-1}}{g^{a_{i+k-1}}} + \frac{f_{i+k}}{g^{a_{i+k}}}\right)$$

this means that applying this transformation at most k times on the partial results we get a solution $\alpha = \Delta\beta + \gamma$, where the denominator polynomial of γ has the form $E^i g^a$ for some $a \leq \max\{a_i, \ldots, a_{i+k}\}$.

From an algorithmic point of view, we do not need any saturation of the shift structure of the denominator, but we now isolate the rightmost boxes from the rest. For instance,

$$\alpha = \frac{x^2+1}{x(x+1)^4(x+3)} = \frac{5x^4+5x^3+15x^2-x+8}{24x(x+1)^4} + \frac{-5}{24(x+3)}$$

From this we get a decomposition $\alpha = \Delta\beta + \gamma$

$$\alpha = \Delta\left(-\frac{5}{24(x+2)}\right) + \frac{-1}{24}\frac{5x^4+5x^3-9x^2-x-16}{x(x+1)^4(x+2)} \tag{1.7}$$

where the right summand of the right hand side has dispersion two, i.e. less than the dispersion of α.

We then iterate the procedure on the remainder γ, reducing the dispersion at each step. At the end we obtain either a trivial remainder, and α is summable, or a remainder with dispersion zero. We then only need to sum up the partial results β. This way we obtain the following decomposition

$$\begin{aligned}
\alpha &= \Delta\left(-\frac{5}{24(x+2)}\right) + \frac{-1}{24}\frac{5x^4+5x^3-9x^2-x-16}{x(x+1)^4(x+2)} \\
&= \Delta\left(-\frac{5}{24(x+2)} - \frac{5}{24(x+1)}\right) - \frac{5x^3+3x-4}{12x(x+1)^4} \\
&= \Delta\left(-\frac{5}{24(x+2)} - \frac{5}{24(x+1)} - \frac{4x^3+9x^2-6x+12}{12x^4}\right) - \frac{3x^2-2x+4}{4x^4}
\end{aligned}$$

If we look more closely at the spectrum of γ after each step, as in Fig. 1.12, then we see that we shifted the rightmost stack of boxes one place to the left at each iteration, while the erased stack is included in the rational part.

Observe that we obtain a final result for γ with the leftmost boxes in the shift structure of the denominator. On the other hand, this corresponds to choosing a fixed representative in $[g]$ for the γ part.

The modification we propose is based on the simple observation that in a similar way the dispersion can be reduced erasing the leftmost, instead of the rightmost, boxes of each component. This way we do not need to fix the stack corresponding to the

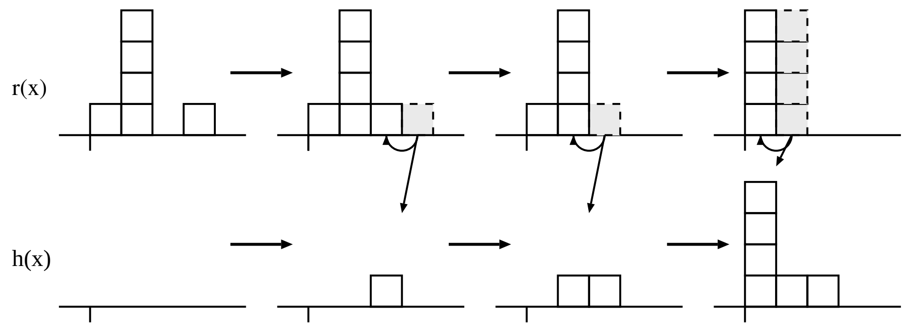

Figure 1.12: Spectrum of the sub-results in Abramov's algorithm

remainder as being the leftmost from the beginning on, but we can choose at each step at which endpoints we want to erase/shift a stack of boxes.

As a strategy we propose to shift at each step the stack (right or left) of smallest height, this means the stack which produces the smallest contribution to the degree of the denominator of the rational part. The algorithm can be implemented in Maple by the simple procedure given in Fig. 1.13.

The procedure call `part(p,h,v,w)` computes the h-part of p, i.e., the maximal factor v of p such that only factors of h arise in v. A more detailed description of this function in given in Subsection 1.6.4. The procedure **abrsum** takes three parameters, the input function α, the dispersion d of α, and finally the symbol x when $\alpha \in \mathbb{K}[x]$.

This modified approach has two main advantages in practice: (i) the decomposition of the rational function is in general easier if the degree of one part is lower and (ii) the degree in the denominator of the rational part h at the end is in general smaller than by fixed choice of the remainder.

In [Pir94] we show that, for a certain class of rational summands, the denominator of the rational part obtained in this way has minimal degree among all solutions. Already in our small example we obtain with the usual Abramov algorithm a rational part with denominator of degree 6, while the modification keeps the degree at 3, as we see in Fig. 1.14.

One can easily convince himself that, at least for single shift equivalence classes, this modification of Abramov's algorithm gives a minimal solution, in the sense that both denominators of β and γ have minimal possible degree. Note the difference with respect to the optimal candidate denominator polynomials defined in Subsection 1.4.4, where the numerator polynomials were not considered, while here the solution is minimal with respect to all possible solutions for the given α (and not, more generally, over all rational functions with a given denominator). In general, however, this minimality

```
abrsum := proc( f, dis, x)  local a,b,p,q,cp,vp,wp,vm,wm,u,newf;
   if f=0 then RETURN(0) fi;
   if dis=0 then RETURN ('Sum'( factor( f), x)) fi;
   p:=numer(f); q:=denom(f);
   cp:=gcd(q,subs(x=x+dis,q));
   if cp=1 then RETURN(abrsum(f,dis-1,x)) fi;
   part(q,cp,'vp','wp'); part(q,subs(x=x-dis,cp),'vm','wm');
   if degree(vm,x)>degree(vp,x) then
        gcdex(vp,wp,p,x,'b','a');
        u:=subs(x=x-1,a/vp); newf:=normal(b/wp+u);
   else gcdex(vm,wm,p,x,'b','a'); u:=-a/vm;
        newf:=normal(b/wm + subs(x=x+1,a/vm));
   fi;
   u + abrsum(newf, dis-1, x);
end:
```

Figure 1.13: Maple implementation of the (modified) Abramov's algorithm

property of the solution does not hold for a function $\alpha = p/q$ where q decomposes into several classes.

In our example the denominator q of α consists of only one shift component. In the case where q splits into more than one shift component the procedure can be applied in the same way. At each step, then, several factors from different shift components are isolated at once.

1.6.3 Paule's algorithm

Paule presents in [Pau93] (see also [Pau95]) an algorithm in analogy to Horowitz's algorithm, also called Hermite-Ostrogradski method, for integration of rational functions (see, for instance, Geddes et al. in [GCL92]). He reduces the problem to the solution of a system of linear equations. The idea consists in choosing an "Ansatz", i.e., a candidate for the denominators of the rational solutions (β, γ). The main difference to the proposed *optimal* solutions will be that Paule makes a fixed choice of the representative g in each class.

Let us explain the algorithm by an example: Consider again the rational function with shift structure as in the left part of Fig. 1.15.

$$\alpha = \frac{x^2+1}{x(x+1)^4(x+3)}$$

We need what Paule calls the *shift saturated extension* (SSE) of the denominator.

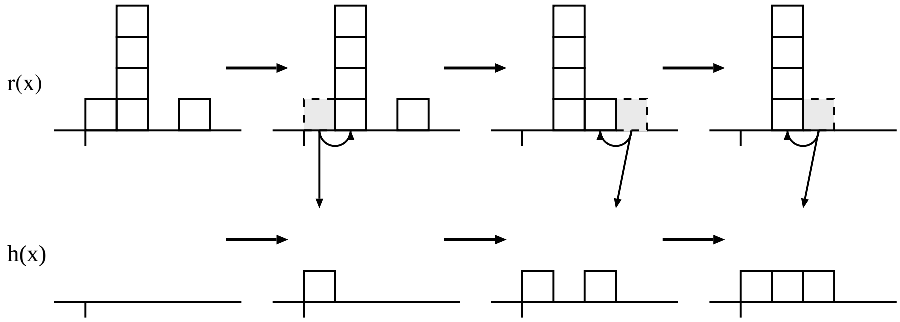

Figure 1.14: Shift-structures in the modified Abramov's algorithm

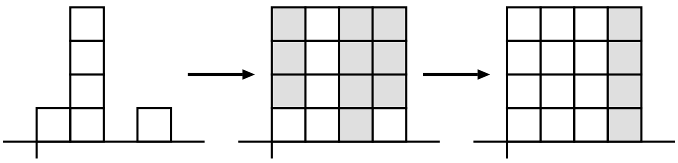

Figure 1.15: Paule's Ansatz for $\alpha = x(x+1)^4(x+3)$

This is an extension of the polynomial by some factors in order to let each stack of boxes in the class have the maximal arising height. In [Pau93] and [Pir92] algorithms are described, that compute such a saturation without knowing the complete factorization of the polynomials. In the example, the SSE of the denominator leads to the following representation of the rational function with shift structure like in the middle part of Fig. 1.15:

$$\alpha = \frac{s}{t} = \frac{x^3(x+2)^4(x+3)^3(x^2+1)}{x^4(x+1)^4(x+2)^4(x+3)^4}$$

We are looking for rational solutions $\alpha = \Delta\beta + \gamma = \Delta\tau/\delta + \varepsilon/\eta$, where Paule proved that a solution exists for $\delta = \gcd(t, E^{-1}t)$ and $\eta = t/\delta$. This means, one takes as denominator for the bound only the factors corresponding to the rightmost boxes in each component of the SSE and puts in the denominator of the summable part all other boxes. In the right part of Fig. 1.15 one can see the decomposition in *summable* and *non-summable* part of the SSE of the denominator.

Substituting the values for δ and η in $\alpha = \Delta\beta + \gamma$ we obtain a polynomial equation. In our example, $\delta = x^4(x+1)^4(x+2)^4$ and $\eta = (x+3)^4$, so one has to find polynomials

τ and ε s.t.

$$x^3(x+2)^4(x+3)^3(x^2+1) = x^4 E\tau - (x+3)^4\tau + x^4(x+1)^4(x+2)^4\varepsilon \qquad (1.8)$$

Since we know δ and η, we have bounds for the degree of γ and ε. Substituting these polynomials with indeterminate coefficients into (1.8) and equating coefficients of same powers of x we obtain a system of linear equations for the coefficients of τ and ε.

The computation can be done by Maple. The solution then is $\beta = \tau/\delta$ and $\gamma = \varepsilon/\eta$, i.e.,

$$\beta = -\frac{128 + 336x + 912x^2 + 1764x^3 + 2061x^4 + 1491x^5 + 661x^6 + 165x^7 + 18x^8}{24x(x+1)^4(x+2)^4}$$

and

$$\gamma = -\frac{25 + 16x + 3x^2}{4(x+3)^4}$$

It should be remarked at this point that in general the computed solutions τ/δ and ε/η are not in reduced form. In other words, the Ansatz for the denominators is not always minimal, as some factors may cancel. Although the denominator polynomial η computed by Paule's algorithm in the generic case is optimal, the degree of the polynomial δ is in general too big. This follows from the fact that always the right-most divisors of the denominator of α are chosen to build the denominator of γ.

From a computational point of view it turns out that the solution of the system of linear equations is the most time consuming task in the algorithm, taking over 80% of the total time.

1.6.4 The algorithm with optimal Ansatz

Here we describe the algorithm based on the observations made in subsection 1.5.2, as presented in [PSb].

For the computation of the Gosper-Petkovšek representation of a rational function we follow the algorithm proposed by Petkovšek in [Pet92], which we include for completeness as Alg. 1. Here $Res_x(p,q)$ denotes the resultant of $p(x)$ and $q(x)$ with respect to the indeterminate x.

The computation of q^+ described in Alg. 2 assumes an algorithm for computing the dispersion. Several algorithms are known for computing the dispersion of a polynomial, as we briefly discuss in Subsection 1.6.6.

Furthermore, we say that v is the *part* of p in q for $v, p, q \in \mathbb{K}[x]$ if $v \mid q$, $\gcd(p, q/v) = 1$ and only factors of p arise in v. In this case we write $v = \text{part}(p,q)$. To make things clear, let us consider an example where p is given by complete factorization as $p = p_1^{\alpha_1} \cdots p_n^{\alpha_n}$, where $\alpha_i > 0$ and q is, by appropriate ordering of the factors,

$$(p, q, r) \leftarrow \texttt{GP-Rep}\ (\alpha)$$

Inputs:
α : a rational function $\in \mathcal{R}$.

Outputs:
p, q, r : Gosper-Petkovšek representation of α, i.e.
$$\alpha = \frac{E\,p}{p} \cdot \frac{q}{E\,r} \text{ with } \gcd(q, E^i\,r) = 1 \text{ for all } i \geq 1 \text{ and } \gcd(p, r) = 1 = \gcd(p, q).$$

Begin

Step 1: *Initialization*

$p \leftarrow 1$, $q \leftarrow \texttt{numer}(\alpha)$, $r \leftarrow \texttt{denom}(\alpha)$

Step 2:

for $h \in \{h' \in \mathbb{N};\, Res_x(q, E^{h'}\,r) = 0\}$ **do**
$d \leftarrow \texttt{gcd}(q, E^h\,r)$
$q \leftarrow q/d$
$r \leftarrow r/E^{-h}\,d$
$p \leftarrow p \cdot \prod_{i=1}^{h} E^{-i}\,d$
endfor

return (p, q, r)
End

Algorithm 1: GP-Rep

given by $q = p_1^{\beta_1} \cdots p_n^{\beta_n} q_{n+1}^{\beta_{n+1}} \cdots q_{n+,}^{\beta_{n+m}}$ with $\beta_i \geq 0$. Then for $v = \text{part}(p, q)$ we have $v = p_1^{\max\{\alpha_1,\beta_1\}} \cdots p_n^{\max\{\alpha_n,\beta_n\}}$

The computation of $\texttt{part}(p, q)$ only needs repeated gcd-computations. Initialize $v_1 \leftarrow \texttt{gcd}(p, q)$ and $q_1 \leftarrow q/v_1$, then compute $v_i \leftarrow \texttt{gcd}(p, q_{i-1})$ and $q_i \leftarrow q_{i-1}/v_i$ for $i = 2, 3, \ldots$ until $v_n = 1$ for some n. Then all factors of p arising in q are isolated and we have $\texttt{part}(p, q) \leftarrow v_1 v_2 \cdots v_n$.

With this notation the algorithm for computing the polynomial q^+ is described in Alg. 2. We obtain q^+ from q by *collecting* all factors in a shift equivalence class at its right end. This is done by, first, isolating the left-most factors with the help of the function $\texttt{part}$, this means $h = \texttt{part}(E^{-d}q, q)$, where d is the dispersion of q. So, the first partial value of q^+ is set to $q^+ = E^d h$. The procedure then is iterated for $j = d, d-1, \ldots, 0$ and eventually we get in q^+ all factors of q shifted to the right of the corresponding class.

$$q^+ \leftarrow \texttt{plus}(q)$$

Inputs:
q : polynomial.

Outputs:
q^+ : polynomial with $\langle q^+, g\rangle = \langle q, g\rangle^+$ for all irreducible g.

Begin

Step 1: *Initialization*

$d \leftarrow \texttt{dis}(q)$, $g \leftarrow q$, $q^+ \leftarrow 1$

Step 2: *Collect all factors at the end of each class*

for $j = d$ **downto** 0 **do**
$h \leftarrow \texttt{part}(E^{-j}\, g, g)$
$g \leftarrow g/h$
$q^+ \leftarrow q^+ \cdot E^j\, h$
endfor

return (q^+)
End

Algorithm 2: `plus`

In Alg. 3 we show the algorithm for solving the rational summation problem with optimal bounds. Remark that Step 3 mainly reduces to the solution of a system of linear equations over the constant field $\mathbb{K}$.

1.6.5 Others

It has to be remarked that there are other algorithms which can be used for the summable case, i.e., for computing solutions of the rational summation problem of the form $(\beta, 0)$, if they exist. In this case one may, for instance, apply Gosper's method for hypergeometric summation. In addition, 1973 Abramov proposed in [Abr71] a different method for solving $\alpha = \Delta\beta$.

Malm and Subramaniam published in [MS95] a work on summation of rational functions. Indeed, also their approach makes use of information provided by the Gosper algorithm, i.e., the Gosper-Petkovšek representation.

$$(\beta, \gamma) \leftarrow \texttt{Opt-Rat-Sum}(\alpha)$$

Inputs:
α : a rational function in $\mathcal{R}$.

Outputs:
β, γ : a solution of the rational summation problem for α

Begin

Step 1: *Initialization*

$t \leftarrow \texttt{denom}(\alpha)$

Step 2: *(Optimal) bounds for the denominators of β and γ*

$(p, q, r) \leftarrow \texttt{GP-Rep}(t/E\,t)$
$u \leftarrow \frac{p\,t}{r}$
$v \leftarrow \texttt{plus}(q)$

Step 3: *Computation of the numerators of β and γ*

$a \leftarrow \sum_{i=0}^{\deg(u)-1} a_i x^i$, $b \leftarrow \sum_{j=0}^{\deg(v)-1} b_j x^j$ for indeterminates a_i's and b_j's

determine the a_i's and b_j's by coefficient comparison from $\quad \alpha = E\,\frac{a}{u} - \frac{a}{u} + \frac{b}{v}$

return $(a/u, b/v)$
End

Algorithm 3: `Opt-Rat-Sum`

We wish to point out, however, that the approach by Malm and Subramaniam usually does not lead to an *optimal* solution of the rational summation problem in the sense discussed in Subsection 1.4.4: the degree of the denominator polynomial u generally is much higher than necessary, so that in the final step a linear system of size bigger than necessary has to be solved. Let us illustrate our claim by an example:

In their article in 1995, Malm and Subramaniam compute:

$$\frac{1}{g} = \Delta\left(\frac{a}{u}\right) + \frac{b}{v}\ ,$$

where

$$g := x^3\,(\,x+2\,)^2\,(\,x+3\,)\,(\,x^2+1\,)\,(\,x^2+4\,x+5\,)^2\ .$$

They obtain as denominator polynomials

$$u = x^3\,(x+1)^3\,(x+2)^3\,(x^2+1)\,(x^2+2x+2)\ ,\ \ v = (x+3)^3\,(x^2+4x+5)^2$$

so that they have to solve the equation with indeterminate polynomial a (b resp.) of degree 12 (degree 6 resp.). They obtain actually polynomials of degree 12 (degree 5 resp.).

With our method from Subsection 1.6.4 we get a denominator polynomial u of degree 9 only:

$$u = x^2\,(x+1)^2(x+2)\,(x^2+1)(x^2+2\,x+2)\ ,\ \ v = x^3\,(x^2+4\,x+5)^2$$

The corresponding numerator polynomials for our solution are

$$\begin{aligned} a &= \frac{1}{43200}(4320 + 20896\,x + 56312\,x^2 + 86758\,x^3 + 97849\,x^4 \\ &\qquad + 82419\,x^5 + 48353\,x^6 + 17367\,x^7 + 2766\,x^8), \\ b &= \frac{-1}{7200}\,(\,-600 + 5300\,x + 13100\,x^2 + 10165\,x^3 + 3484\,x^4 + 461\,x^5\,) \end{aligned}$$

The reason for this non-optimality of the Malm-Subramaniam approach lies in their strategy to artificially shift the maximal exponent in each shift equivalence class to the rightmost position. Our refined algebraic analysis of the situation allows us to leave the maximal exponents where they are, while we choose the numerator polynomial v accordingly — this difference can be observed in the example above. In situations such as given at the end of their article (see [MS95]) as an example, when the maximal exponent in each shift equivalence class of the input denominator polynomial occurs in the rightmost position, both methods will produce solutions of the same degree.

1.6.6 The computation of the dispersion

As we saw in the last sections, all algorithms for computing a solution of the rational summation problem have the computation of the dispersion of a polynomial as a subtask.

From Definition 1.4 we know that for a polynomial $q \in \mathbb{K}[x]$ the dispersion $\mathrm{dis}(q)$ is the maximal value of k such that q and $E^k q$ have a nontrivial common factor.

One method for computing the value of $\mathrm{dis}(q)$ is based on well-known properties of resultants and gcd's. In fact, q and $E^k q$ have a non trivial common factor if and only if the resultant $Res_x(q, E^k q)$ is nonzero.

From this it follows that $\mathrm{dis}(q)$ is the maximal integer root of $Res_x(q, E^k\,q)$ as a polynomial in the indeterminate k, i.e., it is the maximal integer k such that q and $E^k\,q$ have a common factor.

Following this computation scheme, we just need the field $\mathbb{K}$ to allow algorithms for computing the resultant and finding integer roots of univariate polynomials.

On the other hand, if we work over a polynomial ring $\mathbb{K}[x]$ for which (effective) algorithms for irreducible factorization, i.e., into irreducible factors over $\mathbb{K}$, are available, then the task becomes almost trivial, as described in [Pir92]. Assume that the irreducible factorization of q is given, say, by $q = q_1^{e_1} \cdots q_d^{e_d}$. Since we know that the shift of an irreducible polynomial is again irreducible, the computation of the dispersion reduces to comparing all pairs (q_i, q_j) of irreducible factors of q and check for the maximal arising shift such that $E^k q_i = q_j$. This can be easily done looking at the coefficients of q_i and q_j.

Let namely

$$q_i = \sum_{l=0}^{n} a_l x^l \qquad \text{and} \qquad q_j = \sum_{l=0}^{m} b_l x^l$$

then we have

$$E^k q_i = \sum_{l=0}^{n} a_l (x+k)^l = \sum_{l=0}^{n} a_l \sum_{h=0}^{l} \binom{l}{h} x^h k^{l-h} = \sum_{h=0}^{n} \left(\sum_{l=h}^{n} \binom{l}{h} a_l k^{l-h} \right) x^h$$

If $E^k q_i = q_j$ this means, in particular, that

$$m = n, \qquad b_n = a_n \qquad \text{and} \qquad b_{n-1} = a_{n-1} + kna_n$$

must hold. In other words, the only possible value for a shift is $k = (b_{n-1} - a_{n-1})/na_n$, if this is an integer. If this is the case, then we can explicitly compute $E^k q_i$ and check if $E^k q_i = q_j$.

An algorithm would then proceed by a decision procedure like the following: First check if $\deg(q_i) = \deg(q_j)$, then, if this holds, compare the leading coefficients of both polynomials. If also this test succeeds, then a candidate for the shift k is computed from a_n, a_{n-1} and b_{n-1}, and finally $E^k q_i$ is compared to q_j.

This comparisons do not cost much time, so the most time consuming part is the computation of a factorization in irreducible factors.

On the other hand, for commonly used coefficient domains like the integers and the rationals the implemented algorithms in Maple turn out to be efficient enough to overcome the resultant method in speed. This is motivated also by the fact that the polynomial degree of the resultant involved is mainly quadratic with respect to the degree of the original polynomial. Similarly, also the coefficients of the resultant are expected to be significantly larger in magnitude than the coefficients of q.

Man and Wright discuss in some more detail in [MW94] the asymptotic behaviour of both approaches, confirming the practical evidence that, at least when working over rational coefficients, the implementations based on factorization are more efficient.

Nevertheless, it has to be noted that the computation based on resultants is more general, since it can be applied on polynomial rings where no unique factorization is possible too.

1.7 Representing the γ part by Polygamma functions

For the intentions of our study, a solution in the form $\alpha = \Delta\beta + \gamma$ is sufficient, where α, β and γ are all rational functions in $\mathcal{R}$. Introducing the so-called *polygamma functions* one can find a decomposition $\gamma = \Delta\psi$ for the γ part of the solution.

Consider the *Gamma function* $\Gamma(x)$, i.e., the generalization of the factorial to the complex numbers defined by

$$\Gamma(z) = \int_0^\infty e^{-t} t^{z-1}\, dt$$

for z having positive real part. Then the polygamma function Ψ_m of order m, for m positive integer, is defined as the m-th logarithmic derivative of Γ, viz.

$$\Psi_m(x) = \frac{d^m}{dx^m} \log \Gamma(x)$$

By $\Gamma(x+1) = x\Gamma(x)$ we have

$$\begin{aligned} \Delta\Psi_m(x) &= \frac{d^m}{dx^m}\Delta \log\Gamma(x) = \frac{d^m}{dx^m} \log \frac{\Gamma(x+1)}{\Gamma(x)} \\ &= \frac{d^m}{dx^m} \log(x) = \frac{d^{m-1}}{dx^{m-1}} \frac{1}{x} \\ &= \frac{(-1)^{m-1}(m-1)!}{x^m} \end{aligned}$$

This allows us to give the following solution for the rational summation problem for rational functions of the form $a/(x-b)^m$, where a and b are complex numbers:

$$\frac{a}{(x-b)^m} = \Delta\left(a\frac{(-1)^{m-1}}{(m-1)!}\Psi_m(x-b)\right) \tag{1.9}$$

in particular

$$\sum_{x=i}^{j} \frac{a}{(x-b)^m} = a\frac{(-1)^{m-1}}{(m-1)!}\left(\Psi_m(j-b+1) - \Psi_m(i-b)\right)$$

In view of the next section we explicitly remark that, considering now Δ acting on the symbol n, it follows that $\Delta\Psi_1(c\cdot n+d)$ is a rational function in n for positive integer c and complex d. Namely we have, for $\Psi = \Psi_1$:

$$\Psi(cn+d) = \sum_{k=1}^{cn-1} \frac{1}{k+d} + \Psi(d+1) \quad \text{and} \quad \Delta\Psi(cn+d) = \sum_{k=cn}^{cn+c-1} \frac{1}{k+d} \tag{1.10}$$

Assume now that for $\alpha \in \mathcal{R}$ the pair (β, γ) is solution of the rational summation problem, and the full partial fraction decomposition of $\gamma = \tau/\delta$ is

$$\gamma = \sum_{\delta(\eta)=0} \sum_{j=1}^{j_\eta} \frac{\tau_{\eta,j}}{(x-\eta)^j}$$

then from equation (1.9) it follows that

$$\tilde{\beta} = \beta + \sum_{\delta(\eta)=0} \sum_{j=1}^{j_\eta} \frac{\tau_{\eta,j}(-1)^{j-1}}{(j-1)!} \Psi_m(x-\eta)$$

is a solution such that $\alpha = \Delta\tilde{\beta}$.

From a computational point of view the main problem is to factorize the denominator of γ completely into linear factors. This is, in general, not possible.

However, M. Bronstein and B. Salvy presented in [BS93] an algorithm for computing the full partial fraction decomposition of rational functions, involving no factorization but only operations in $\mathbb{K}$. We do not describe the algorithm in detail, one may look at [BS93] for proofs, but we wish to point out *in which sense* their algorithm computes a full decomposition.

Consider now $\alpha \in \mathcal{R}$, with denominator δ and full partial fraction decomposition

$$\alpha = \sum_{\delta(\eta)=0} \sum_{j=1}^{j_\eta} \frac{\tau_{\eta,j}}{(x-\eta)^j}$$

Furthermore, let $\delta = \delta_1 \delta_2^2 \cdots \delta_m^m$ be the square-free factorization of δ. Then, for all $k = 1, \ldots, m$ the algorithm computes polynomials $\zeta_{k,1}, \ldots, \zeta_{k,k}$ and factors $\tilde{\delta}_{k,1}, \ldots, \tilde{\delta}_{k,k}$ of δ_k in $\mathbb{K}[x]$ such that the following decomposition holds:

$$\alpha = \sum_{k=1}^{m} \sum_{j=0}^{k} \sum_{\tilde{\delta}_{k,j}(\eta)=0} \frac{\zeta_{k,k-j}(\eta)}{(x-\eta)^{k-j}}$$

Combining the algorithm by Bronstein and Salvy with one of the algorithms for rational summation presented in this chapter, we obtain, in the sense specified above, a *complete* solution to the problem without using factorization in $\mathbb{K}[x]$. For any $\alpha \in \mathcal{R}$ such an algorithm computes a solution in the form

$$\alpha = \Delta\beta + \sum_{k=1}^{m} \sum_{j=0}^{k} \sum_{\tilde{\delta}_{k,j}(\eta)=0} (-1)^{k-j-1} \frac{\zeta_{k,k-j}(\eta)}{(k-j-1)!} \Psi_{k-j}(x-\eta)$$

where β, the $\tilde{\delta}_{k,j}$'s and the $\zeta_{k,j}$'s are explicitly computed.

1.8 Applications

In this section we present some applications for proving identities involving rational summations.

E. Clarke and X. Zhao describe in [CZ94] a theorem prover based on Mathematica. Among others, they give the identities listed below. as examples for problems that can not be solved by symbolic computation decision procedures. The identities are taken from Chapter 2 of Ramanujan's Notebooks†.

Using the considerations in Section 1.7 we show that almost all of those examples can be solved by rational summation methods in a deterministic way, without involving theorem proving techniques. The proof of the identities reduces to testing equality of rational functions.

Let us first define, following Ramanujan's abbreviations, $\Phi(x,n)$ and $\varphi(x,n)$ as

$$\Phi(x,n) = 1 + 2\sum_{k=1}^{n} \frac{1}{-kx + k^3x^3} \quad \text{and} \quad \varphi(x,n) = \sum_{k=1}^{n} \frac{1}{-kx + k^3x^3}$$

1.8.1 Identities with finite sums

Here we consider several identities involving $\Phi(x,n)$ and $\varphi(x,n)$ with given x and n is finite.

We exploit all details of a possible mechanical proof by rational summation methods for the first identity considered in Section 3.1 of Clarke and Zhao's work:

1. $\displaystyle\sum_{k=1}^{n} \frac{1}{n+k} = \frac{n}{2n+1} + \varphi(2,n)$

In order to prove this identity we first solve the rational summation problem for the Φ's and φ's involved, converting this expressions into polygamma functions. So we get an equivalent identity $lhs(n) = rhs(n)$, where lhs and rhs only involve rational functions in n and polygamma functions. Then it is sufficient to prove that $\Delta\, lhs(n) = \Delta\, rhs(n)$ and to check the identity for an initial value. From Equation (1.10) we know that this reduces to proving an identity of rational functions, which can be done mechanically.

Let us express $\varphi(x,n)$ in terms of polygamma functions. Since $-kx + k^3x^3 = kx(kx-1)(kx+1)$ we have

$$\begin{aligned} \varphi(x,n) &= \sum_{k=1}^{n} \frac{1}{-kx + k^3x^3} = \sum_{k=1}^{n} \left(\frac{1}{kx} + \frac{1}{2}\frac{1}{kx-1} + \frac{1}{2}\frac{1}{kx+1} \right) \\ &= \sum_{k=1}^{n} \frac{1}{x} \Delta \left(\Psi(k) + \frac{1}{2}\Psi(k - \frac{1}{x}) + \frac{1}{2}\Psi(k + \frac{1}{x}) \right) \end{aligned}$$

†B.C. Berndt, *Ramanujan's Notebooks, Part I*, Springer-Verlag, 1985, pp. 25-43

$$= \frac{1}{x}\left(\Psi(n+1)+\frac{1}{2}\left(\Psi(n-\frac{1}{x}+1)+\Psi(n+\frac{1}{x}+1)\right)\right)+C$$

where Δ acts on k and C is a constant with respect to n.

Similarly,

$$\sum_{k=1}^{n}\frac{1}{n+k}=\sum_{k=1}^{n}\Delta\Psi(n+k)=\Psi(2n+1)-\Psi(n+1)$$

We substitute now into identity 1. and apply the Δ operator with respect to n. With Equation (1.10) we get the equality of rational functions

$$\frac{1}{2n+1}+\frac{1}{2n+2}-\frac{1}{n+1}=\frac{n+1}{2n+3}-\frac{n}{2n+1}+\frac{1}{2}\left(\frac{1}{n+1}+\frac{1}{2}\left(\frac{1}{n+\frac{1}{2}}+\frac{1}{n+\frac{3}{2}}\right)\right)$$

which finds a straightforward verification. In Maple, for instance, this would reduce to asking if `simplify(lhs-rhs)=0` holds, where `lhs` is the left hand side of the identity, and `rhs` the right hand side, respectively.

The proof is then completed by checking an initial condition, say the values of both sides of the original identity for $n=1$.

This way one can automatically prove all identities in Section 3.1 of [CZ94] which involve only rational functions in k as summands, i.e., all of them but identity 5. and 10. in their list. These are:

2. $\displaystyle\sum_{k=1}^{n}\frac{n-k}{n+k}=2n\varphi(2,n)-\frac{n}{2n+1}$

3. $\displaystyle\sum_{k=1}^{2n+1}\frac{1}{n+k}=\Phi(3,n)$

4. $\displaystyle\left(\sum_{k=1}^{n}\frac{1}{n+k}\right)+\left(\sum_{k=0}^{n}\frac{1}{2n+2k+1}\right)=\Phi(4,n)$

6. $\displaystyle\Phi(6,n)=\frac{2}{3}\left(\sum_{k=1}^{n}\frac{1}{n+k}\right)+\left(\sum_{k=0}^{2n}\frac{1}{2n+2k+1}\right)$

7. $\displaystyle 2\Phi(4,n)=\Phi(2,2n)+\frac{\Phi(2,n)}{2}+\frac{1}{(4n+1)(4n+2)}$

8. $\displaystyle\Phi(4,n)=\frac{1}{2}\left(\sum_{k=n+1}^{2n}\frac{1}{k}\right)+\left(\sum_{k=2n+1}^{4n+1}\frac{1}{k}\right)$

9. $\displaystyle 2\Phi(6,n)+\frac{\Phi(2,n)}{3}=\Phi(3,n)+\Phi(2,3n)+\frac{2}{(6n+1)(6n+2)(6n+3)}$

A Maple session

As an example we show the Maple session for the automatic proof of identity 3. above.

Define the Delta operator applied on Psi(C*n+D)

```
> Delta:= proc(s)
> match(s=A*Psi(C*n+D),n,'l');
> eval(subs(l,'A*sum(1/(k+D),k=C*n..(C*n+C-1))'));
> end:
```

The functions Phi and phi defined by Ramanujan

```
> phi1:= (x,n) -> sum(1/(-k*x+k^3*x^3),k=1..n);
```

$$\phi 1 := (x,\, n) \to \sum_{k=1}^{n} \frac{1}{-k\,x + k^3\,x^3}$$

```
> Phi:= (x,n) -> 1+2*phi1(x,n);
```

$$\Phi := (x,\, n) \to 1 + 2\,\phi 1(x,\, n)$$

Proof of Identity 3.

```
> eq:= 'sum(1/(n+k),k=1..2*n+1)' = 'Phi(3,n)';
```

$$eq := \sum_{k=1}^{2\,n+1} \frac{1}{n+k} = \Phi(3,\, n)$$

Convert to polygamma notation

```
> eval(eq);
```

$$\Psi(3\,n+2) - \Psi(n+1) = -\frac{2}{3}\,\Psi(n+1) + \frac{1}{3}\,\Psi(n+\frac{2}{3}) + \frac{1}{3}\,\Psi(n+\frac{4}{3}) + \ln(3)$$

Apply the Delta operator on both sides of the identity

```
> map(Delta,lhs(eq));
```

$$\frac{1}{3\,n+2} + \frac{1}{3\,n+3} + \frac{1}{4+3\,n} - \frac{1}{n+1}$$

```
> map(Delta,rhs(eq));
```

$$-\frac{2}{3}\,\frac{1}{n+1} + \frac{1}{3}\,\frac{1}{n+\frac{2}{3}} + \frac{1}{3}\,\frac{1}{n+\frac{4}{3}}$$

Check for equality

```
> simplify("-"");
```

$$0$$

Check the initial condition

```
> subs(n=1,eq);
```

$$\Psi(5) - \Psi(2) = -\frac{2}{3}\,\Psi(2) + \frac{1}{3}\,\Psi(\frac{5}{3}) + \frac{1}{3}\,\Psi(\frac{7}{3}) + \ln(3)$$

```
> eval(");
```

$$\frac{13}{12} = \frac{13}{12}$$

1.8.2 Identities with infinite sums

Several identities involving infinite sums listed in Section 3.2 of [CZ94] can be proven by means of rational summation and limit computations. This can be done, for instance, using the functions **sum** and **limit** implemented in Maple.

As a matter of fact, Maple can even find the right hand side of the identities.

1. $\Phi(2,\infty) = 2\log(2)$
2. $\Phi(3,\infty) = \log(3)$
3. $\Phi(4,\infty) = \frac{3}{2}\log(2)$
4. $\Phi(6,\infty) = \frac{\log(3)}{2} + \frac{\log(4)}{3}$
5. $\sum_{k=1}^{\infty} \frac{1}{(2(2k-1))^3 - 2(2k-1)} = \frac{\log(2)}{4}$
7. $\sum_{k=1}^{\infty} \frac{1}{(3(2k-1))^3 - 3(2k-1)} = \frac{\log(3)}{4} - \frac{\log(2)}{3}$

Some infinite sums solved by Maple

```
> Phi(4,infinity);
```

$$\frac{3}{2}\ln(2)$$

```
>  Phi(3,n);
```

$$-\frac{2}{3}\,\Psi(n+1)+\frac{1}{3}\,\Psi(n+\frac{2}{3})+\frac{1}{3}\,\Psi(n+\frac{4}{3})+\ln(3)$$

```
>  limit(",n=infinity);
```

$$\ln(3)$$

```
>  sum(1/((3*(2*k-1))^3-3*(2*k-1)),k=1..infinity );
```

$$\frac{1}{4}\ln(3)-\frac{1}{3}\ln(2)$$

2

Parallel Implementations

2.1 Summation of rational functions

Both the algorithm of Paule and the algorithm of Abramov suggest the use of parallel computation methods. The first one because solving a system of linear equations is a classical parallel task. The latter for the possibility of solving the problem in parallel along the localization to the different shift equivalence classes of the denominator polynomial.

We implemented the parallel algorithm in ||MAPLE||, speak *parallel Maple*, a parallel computer algebra system developed at RISC-Linz. It is our goal to show that, in practice, ||MAPLE|| offers a suitable environment for developing parallel algorithms without technical knowledge about parallelism, and is based on the syntax of the sequential system Maple.

This section reports on joint work with Kurt Siegl, see [PSa].

2.1.1 The system ||MAPLE||

The algorithms have been implemented in ||MAPLE|| (for an introduction to the system see also [Sie93]) which is a portable system for parallel symbolic computation. The core of the system is built on top of an interface between the parallel declarative programming language Strand (see the description by Foster and Taylor in [FT89]) and the sequential computer algebra system Maple (see [CGGG83]), in the hope to keep both the elegance of Strand and the power of the existing sequential algorithms in Maple.

||MAPLE|| programs may run on different hardware, ranging from shared-memory machines over distributed memory architectures up to networks of workstations, without any modification or recompilation. All necessary communication is done automatically by the system without any additional programming effort. Since ||MAPLE|| uses implicit parallelism, it allows writing parallel programs without any expert knowledge in parallel programming.

The ||MAPLE|| system has two layers (Figure 2.1). The top layer is the parallel declarative programming language Strand which controls the parallel execution of an algorithm. For performing sequential tasks we may call arbitrary Maple functions or sequences of Maple statements in the underlying Maple system over some interface routines. The result is a parallel programming system with the full functionality of Maple and parallel power of Strand.

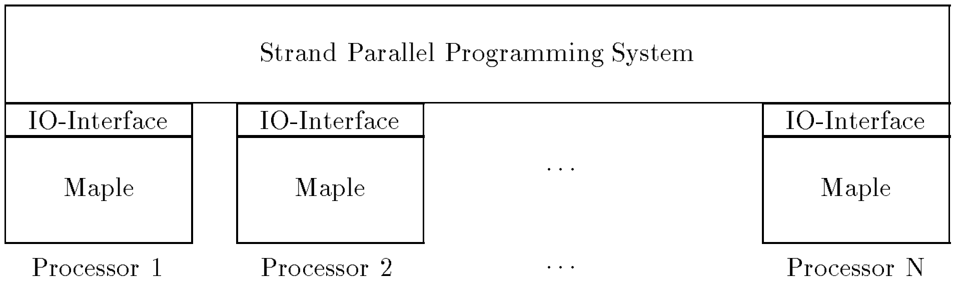

Figure 2.1: Structure of the ||MAPLE|| system

Usually, ||MAPLE|| programs reflect this structure and consist of two parts. The Strand code for the administration of the parallel tasks, and the Maple code for the functions to be executed sequentially. On the other hand, ||MAPLE|| has a set of pre-parallelized functions for some typical algorithmic structures, which allow us to write parallel programs directly as Maple code, without the use of Strand, as in the application presented here.

Here we use a function called `peval`, designed for algorithms following the divide and conquer principle. The function may be used for a wide range of algorithms in computer algebra, in particular for recursively defined algorithms.

The ||MAPLE|| call `peval( [ f1(x1), ...., fn(xn) ], recompose)` requires two arguments:

1. A list of unevaluated functions `f1(x1)...fn(xn)`. Each of these functions will be evaluated on different processes/nodes in parallel. Note that the functions `fi` might have more than one input parameter.

2. A composition function `recompose` which takes as input the results produced by the parallel evaluations and produces the final output of the `peval` call. The subresults are contained in `parg[1],...,parg[n]`.

In particular, the functions `fi(xi)` do not need to represent the same computations and may contain other `peval` statements, and so on.

As an example for the usage of the `peval` function we give a parallel version of the well known Karatsuba algorithm for multiplying long integers in Figure 2.2.

Here we execute the recursive sub-multiplications in parallel as long as both integers have more than 50 digits, otherwise we call the built-in sequential version.

```
imult:=proc(x,y) local lx,ly,n,x1,x2,y1,y2;
  lx:=length(x); ly:=length(y);
  if (lx > 50) and (ly > 50)
   then                # Parallel Karatsuba algorithm for large numbers
     n:=round(max(lx,ly)/2);
     x1:=iquo(x,10^n,'x2');
     y1:=iquo(y,10^n,'y2');

     peval(['imult'(x1,y1),'imult'(x2,y2),'imult'(x1+x2,y1+y2)],
          '<u*10^(2*n)+(w-u-v)*10^n+v | n,u,v,w>'
          (n,parg[1],parg[2],parg[3])
         );
   else                # Built in algorithm for small numbers
     x*y;
  fi;
end;
```

Figure 2.2: Parallel integer multiplication

2.1.2 Paule's Algorithm

As we already saw, the most important task in the implementation of Paule's algorithm is the solution of a system of linear equations, where the coefficients of the system come from the field $\mathbb{K}$ considered (usually rational numbers, but also symbolic rational functions). For this reason the main goal is to improve this part of the computation.

Solving a system of linear equations in general is a well-known task. In the sequential case, the Gaussian elimination algorithm will be used. But it turns out that the equation solver known as Gauss Jordan algorithm allows a better parallelization. Here we successively eliminate all elements of a column in parallel, which should give us a nearly optimal speedup for larger matrices up to a high number of processors. See for instance [GCL92] for these well-known elimination algorithms.

While with fixed-size coefficients the time required per element is constant, the elimination time with symbolic entries will vary and generate unbalanced execution times for individual rows, thus limiting the benefits of parallelism. Due to worse complexity, the Gauss Jordan algorithm is usually slower than the Gaussian algorithm by a factor of 3, so we need at least a speedup by 3 to compensate the algorithmic disadvantages. Additionally, an efficient execution with symbolic entries depends heavily on a few small details which may improve the execution time up to factor of 50 and more:

1. The smallest element in the row with minimal total size should be selected as the pivot element. In our implementation we used the amount of memory required

for an element to determine its size.

2. Coefficients should always be simplified and normalized to integers.

3. Representing rows of the matrix as polynomials over a set of new variables will give us access to the highly optimized polynomial operations available in Maple.

4. In a parallel environment, communication may be optimized by grouping several rows together to form a computation block.

The effect of all these optimizations is shown by the following table comparing several built-in Maple algorithms with our implementation on a relatively small matrix with dimension 78.

Algorithm	Time	Data Type	Pivot Search
Gauss Jordan	139 min	Matrix	none
Gauss	24 min	Matrix	row pivoting
Solve (Gauss)	4 min	Polynomials	min element in min row
Our Gauss Jordan	13 min	Polynomials	min element in min row
Parallel	2 min	Polynomials	min element in min row

The table shows that by parallelism and a few other improvements the Gauss Jordan method is able to beat a highly optimized Gauss elimination algorithm by a significant factor using a couple of workstations.

We tested our algorithm on several rational functions with rational coefficients. In Figure 2.4 we summarize the most important timings using a network version of ||MAPLE|| on a cluster of 12 Silicon Graphics (SGI) workstations. All computation times are given in the form min:sec.

The first column shows the shift structure of the denominator, viz. $2, 5, 9$ means that the denominator of the input has three shift classes, respectively of dispersion 2, 5 and 9, respectively. The dimension of the system is given in the ninth column, while the columns six and seven are the times needed by our parallel implementation running on 12 processors or on a single processor. It should be remarked that among the 12 processors only 11 are involved in the real computations, as the first is used as manager.

The implementation using the built-in solver in Maple is reported in the eighth column, while in the last we give the obtained speedup. The second column of the speedup entries represent the speedup between the parallel implementation of Paule's algorithm on 12 processors and the sequential reference implementation.

2.1.3 Abramov's Algorithm

The parallelization of the modified Abramov's algorithm lies mainly in the decomposition of the rational function with respect to the shift equivalence classes of the denominator. Let now $\alpha \in \mathcal{R}$ be given by the canonical form

$$\alpha = \alpha_{[g_1]} + \cdots + \alpha_{[g_s]}$$

and let $\alpha_{[g_i]} = p_i/\tilde{q}_i$ for $i = 1, \ldots, s$.

After this we apply the procedure on each of the summands $p_i/\tilde{q}_i$ in parallel. In the end we only need to sum up the results obtained by each parallel function call.

First we compute the decomposition of the denominator q of $\alpha = p/q$, say $q = \tilde{q}_1 \cdots \tilde{q}_s$ as above. Remark that this computation is also done by our implementation of Paule's algorithm.

Then we apply the divide-and-conquer principle in parallel using the ||MAPLE|| function `peval`, as in the following ||MAPLE||-code.

```
abrpar:= proc(f,x) local fs;
   fs:= shift_structure(f,x);
   decompose(fs,x);
end:

decompose := proc(f,x) local f1, f2;
   if nb_shift_comp(f,x)=1 then RETURN(abrsum(f,dis(f,x),x)) fi;
   shift_par_frac(f,x,'f1','f2');
   peval(['decompose'(f1,x),'decompose'(f2,x)],
        'sum_up'(parg[1],parg[2]));
end:
```

Figure 2.3: Parallel code for Abramov's algorithm

The function `shift_structure` computes the shift structure of $f = p/q$, i.e. the decomposition into shift classes of the denominator $q = \tilde{q}_1 \cdots \tilde{q}_s$. In `decompose` the number of shift classes is checked. If q has only one shift class, then the sequential procedure `abrsum` is applied, computing a solution of the rational summation problem for f.

Otherwise, if $s \geq 2$, then the function `shift_par_frac` determines a decomposition

$$f = f_1 + f_2 = \frac{p_1}{\tilde{q}_1 \cdots \tilde{q}_t} + \frac{p_2}{\tilde{q}_{t+1} \cdots \tilde{q}_s}$$

where $t = \lfloor \frac{s}{2} \rfloor$. This is done by the extended Euclidean algorithm, computing p_1, p_2 such that $p_1 \cdot \tilde{q}_{t+1} \cdots \tilde{q}_s + p_2 \cdot \tilde{q}_1 \cdots \tilde{q}_t = p$. Then `decompose` is applied in parallel to f_1 and f_2 by a `peval` call. The function `sum_up` just combines the results, summing up the summable and the nonsummable parts of the partial solutions.

In this kind of parallelization the number of really concurrent processes is directly related to the number s of shift components of q, in contrast to the solution of a linear system.

The timings for the same examples used for Paule's algorithm are given in columns two to five of Figure 2.4.

The entries are analogous to those corresponding to the algorithm of Paule. From the third and fourth columns follows the unexpected but interesting fact that the parallel implementation carried out on one processor is already faster than the sequential algorithm. Note that the sequential implementation considers all classes at once, so it does not need to decompose the rational function with respect to the shift classes of the denominator.

As a result of parallel considerations, this observations imply that also a sequential implementation should compute the partial fraction decomposition along the shift classes and apply the procedure on the single components (whenever efficient factorization algorithms are available, as at least over the coefficient field $\mathbb{Q}$ considered in our examples).

2.1.4 Comparison

Summerizing the data in Figure 2.4, for our examples the implementation of Abramov's algorithm is faster, in both the sequential and parallel case.

Shift cl.	Abr.				Paule					Speedup
	12 P	1 P	Seq	Deg	12 P	1 P	Seq	Dim	Deg	Abr./Paule
2,5,9	12	19	1:18	42	2:31	13:31	4:27	78	58	6.4 / 1.7
	53	1:16	3:38	53	7:58	56:42	16:06	92	67	4.0 / 2.0
	25	45	2:49	50	18:10	143:54	38:59	111	50	6.7 / 2.1
4,10,13	1:06	1:21	3:29	81	7:46	42:22	17:14	110	84	3.1 / 2.2
	30	45	3:31	69	15:12	106:50	43:21	123	100	6.9 / 2.8
2,2,7,12	15	21	59	28	1:59	10:36	2:48	65	44	3.9 / 1.4
	9	13	28	33	1:38	5:31	3:11	69	44	3.0 / 1.9
1,5,10,11	9	18	48	44	9:14	34:10	11:48	89	44	4.8 / 1.3
	5	9	16	52	1:09	3:21	1:50	66	53	2.8 / 1.5
3,3,7,12	8	21	1:17	33	10:03	73:16	26:11	99	57	8.9 / 2.6
2,2,2,3,3,3	1:30	2:01	3:41	31	3:37	25:33	16:50	66	31	2.4 / 4.6
	1:16	1:59	6:33	46	17:55	164:44	17:12	84	46	5.1 / 0.9

Figure 2.4: Timings

In the table one also finds the value of the degree in the denominator of the rational part computed by each algorithm. As to be expected from the consideration in Subsection 1.6.2 our modification of Abramov's algorithm often computes a result with significantly smaller degree.

We remark that the system solver used in Paule's algorithm would take considerable advantage of having more processors available. Since the number of parallel processors used by Abramov's algorithm is given by the input, the implementation does not take any advantage of more processors available. This means that by an appropriate number of processors, Paule's algorithm would be faster in certain cases, e.g., for rational functions with few shift classes, almost shift saturated structure, and corresponding to a system of high dimension.

2.2 Solving a linear system by p-adic arithmetics

Let us restrict the field of coefficients $\mathbb{K}$ to the field of rational numbers $\mathbb{Q}$. Then the p-adic representation of rationals gives a further possibility for parallelizing a linear system solver.

We report on joint work with Carla Limongelli (see [LP96]), where we describe the use of truncated p-adic expansion for handling rational numbers by parallel algorithms for symbolic computation. As a case study we propose a parallel implementation for solving linear systems over the rationals.

The parallelization is based on a multiple homomorphic image technique and the result is recovered by a parallel version of the Chinese remainder algorithm. Using a MIMD machine, we compare the proposed implementation with the classical modular arithmetic, showing that truncated p-adic arithmetic is a feasible tool for solving systems of linear equations working directly over rational numbers.

The implementation leads to a speedup factor up to seven by using ten processors in comparison to the sequential implementation.

p-adic arithmetic has been chosen for two main reasons:

1. p-adic arithmetic representation provides a unified form to treat numbers and functions by means of truncated power series and it constitutes the mathematical background for the definition of basic abstract data structures for a homogeneous computing environment. A unified representation can be obtained when numbers and functions are represented by power series and p-adic analysis offers an appropriate mathematical setting in this sense [Kri85]. In [LT92] it is shown how it is possible to treat numbers by truncated power series, together with the most general p-adic construction methods in an integrated computing environment.

2. p-adic arithmetic is an exact arithmetic and its algebraic framework overcome the problems of floating point arithmetic, essentially due to a lack of algebraic setting. This last characteristic belongs to modular arithmetic too [Knu81, GK84], but the difference is that while modular arithmetic works over the integers, p-adic arithmetic operates on rational numbers. In [Lim93b] the advantages of working directly over the rationals are shown. Moreover in [LT93, Lim93a] it is shown that also algebraic numbers are representable in this arithmetic.

For the case study we choose the classical linear algebra problem of solving linear systems, which is of relevance for rational summation too. For a positive integer n we want to solve a system of n linear equations for the n unknowns $x_1, \ldots, x_n$

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,n}x_n & = & b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \ldots + a_{2,n}x_n & = & b_2 \\ \ldots & & \ldots \\ a_{n,1}x_1 + a_{n,2}x_2 + \ldots + a_{n,n}x_n & = & b_n \end{cases} \tag{2.1}$$

where $a_{i,j}$ and b_i ($i = 1, \ldots, n$ and $j = 1, \ldots, n$) are rational numbers. We will denote the system (2.1) by $A\vec{x} = \vec{b}$.

The parallel implementation for solving linear systems is based on Gaussian elimination algorithm and the p-adic representation of rational numbers via truncated power series with respect to a prime basis p. Our parallelization consists of applying the well known Gaussian elimination method (see for instance [GCL92]) for several homomorphic images of the problem with respect to different prime bases, and recovering the result by the Chinese Remainder Algorithm (CRA). The order of truncation r, as well as the prime bases, is chosen in accordance with an a priori estimation of the magnitude of the solution of the problem. This allows us to do error-free computations directly with rational numbers. For a detailed treatment of p-adic arithmetic in the context of symbolic computation, we refer to [GK84, Kri85, Lim93a]. Krishnamurthy in [Kri93] proposes a similar method based on CRA, EEA (Extended Euclidean Algorithm) and HLA (Hensel Lifting Algorithm), for inverting matrices with rational entries. Dixon's approach [Dix82] for solving systems of linear equations has been studied in [Vil88b, Vil88a]. In this work we particularly want to stress the usefulness of p-adic representation of rational numbers via Hensel codes, therefore we compare our implementation with an equivalent parallel one which uses modular arithmetic.

In order to show that p-adic arithmetic provides an efficient tool for solving linear systems over the rational numbers, we compared our implementation with one using modular arithmetic and with a sequential implementation in the computer algebra system Maple [CFG$^+$86]. Aspects of our parallel implementation are also presented in [LP94].

The implementation was done in PACLIB, a C-language library for parallel symbolic computation [H$^+$92], on a Sequent parallel machine with a MIMD architecture.

2.2.1 Basic notions of p-adic arithmetic

A nonzero rational number $\alpha = a/b$ can always be uniquely expressed as

$$\alpha = \frac{c}{d} \cdot p^e$$

where e is an integer, p is a fixed prime number, and c, d, and p are pairwise relatively prime integers. This representation is called the *normalized form* of α. Moreover $\mathring{\mathbb{Q}}$ will denote the set of rational numbers c/d such that $\gcd(d, p) = 1$. The function

$$\|\cdot\|_p : \mathbb{Q} \to \mathbb{R}$$

from the rational numbers $\mathbb{Q}$ to the real numbers $\mathbb{R}$, defined as

$$\|\alpha\|_p = \begin{cases} p^{-e} & \text{if } \alpha \neq 0 \\ 0 & \text{if } \alpha = 0 \end{cases}$$

then is a norm on $\mathbb{Q}$ (see [Kob77]), called the p-adic norm. On the basis of this p-adic norm, it is possible to define a p-adic metric on $\mathbb{Q}$, such that, given two rational numbers α and β, their distance $d(\alpha, \beta)$ is expressed as:

$$d(\alpha, \beta) = \|\alpha - \beta\|_p.$$

Then $(\mathbb{Q}, d)$ is a metric space. Let $\mathbb{Q}_p$ be the set of equivalence classes of Cauchy sequences in $(\mathbb{Q}, d)$, then the system $(\mathbb{Q}_p, +, \cdot)$ forms a field called the field of p-adic numbers, and $(\mathbb{Q}_p, d)$ is a complete metric space.

The main characteristics of the field of p-adic numbers are:

1. the series
$$\sum_{i=0}^{\infty} p^i$$
converges to 1/(1-p) in $(\mathbb{Q}_p, d)$;

2. every non zero rational number α can be uniquely expressed in the form:
$$\alpha = \sum_{i=e}^{\infty} a_i p^i; \quad a_i \in \mathbb{Z}_p; \quad e \in \mathbb{Z}; \quad \|\alpha\|_p = p^{-e}; \quad a_e \neq 0, \tag{2.2}$$
where $\mathbb{Z}$ represents the set of integer numbers.

The p-adic representation of a rational number α is then an infinite sequence of digits (the p-adic digits) which are the coefficients of the series given in (2.2):

$$\alpha = (a_e a_{e-1} \ldots a_{-1} \quad . \quad a_0 a_1 a_2 \ldots).$$

Let us recall that the p-adic expansion of a rational number is periodic. Therefore the p-adic representation can also assume the following form:

$$\alpha = (a_e a_{e-1} \ldots a_{-1} \quad . \quad a_0 \ldots a_{k-m-1}{}' a_{k-m} \ldots a_{k-1} a_k)$$

where the $m+1$ rightmost digits are the period.

Let us now describe the procedure which computes the p-adic representation of a given rational number α:

p-adic Representation of a Rational Number

Input: p: prime number;
$\alpha \in \mathbb{Q}, \alpha \neq 0$, represented by its normalized form, $\alpha = c/d \cdot p^e$;
Output: the coefficients $a_e, a_{e+1}, a_{e+2}, \ldots$ of the p-adic expansion of α;
Begin

$c_1/d_1 := c/d;$

$i := 0;$

repeat

$a_{e+i} :=\mid c_{i+1}/d_{i+1} \mid_p;$

$c_{i+2}/d_{i+2} := \frac{1}{p}\left(c_{i+1}/d_{i+1} - a_{e+i}\right);$

$i := i + 1;$

until the period is detected;

End

Here $\mid c_i/d_i \mid_p = \mid c_i \mid d_i^{-1} \mid_p \mid_p$ is the least nonnegative remainder of c_i/d_i *mod* p, where $\mid d_i^{-1} \mid_p$ denotes the inverse of d_i in $\mathbb{Z}_p$. We note that the hypothesis of primality for p is necessary in order to ensure the existence and the uniqueness of $\mid d_i^{-1} \mid_p$. From now on we will consider p a prime number.

Example 2.1 We compute the p-adic expansion of the rational number $3/4$, with $p = 5$ (in this case $e = 0$):

$$\alpha = \frac{3}{4} \cdot 5^0, \quad \frac{c_1}{d_1} = \frac{3}{4};$$

$$\begin{aligned}
a_0 &= \mid \frac{c_1}{d_1} \mid_p = \mid \frac{3}{4} \mid_5 = \mid 3 \cdot \mid 4^{-1} \mid_5 \mid_5 = \mid 12 \mid_5 = 2; \\
\frac{c_2}{d_2} &= \frac{1}{5}\left(\frac{3}{4} - 2\right) = \frac{1}{5}\left(-\frac{5}{4}\right) = -\frac{1}{4}; \\
a_1 &= \mid \frac{c_2}{d_2} \mid_p = \mid -\frac{1}{4} \mid_5 = \mid \mid -1 \mid_5 \cdot \mid 4^{-1} \mid_5 \mid_5 = 1; \\
\frac{c_3}{d_3} &= \frac{1}{5}\left(-\frac{1}{4} - 1\right) = \frac{1}{5}\left(-\frac{5}{4}\right) = -\frac{1}{4}; \\
a_2 &= \mid \frac{c_3}{d_3} \mid_p = \mid -\frac{1}{4} \mid_5 = 1.
\end{aligned}$$

In general this process will not terminate, but, since we are assuming that α is a rational number, the p-adic expansion will be periodic. So, in this case, we just have to continue the computation of the p-adic coefficients until the period is detected. In our example the p-adic expansion of the number $3/4$ is $.211\ldots = .2\,'1$. □

Arithmetic operations on p-adic numbers are carried out, digit by digit, starting from the left-most digit a_e, as in usual base p arithmetic operations.

The division operation on p-adic numbers is performed in a different way with respect to usual integer arithmetic. Starting from the left-most digit of both the dividend and the divisor, we obtain the left-most digit of the quotient, and so on, in a way similar to the other three basic p-adic arithmetic operations.

For automatic p-adic arithmetic computations, the length of p-adic digit sequences might cause a problem. A natural solution is given by introducing a finite length p-adic arithmetic on the so-called Hensel codes as we will show below.

Definition 2.1 (HENSEL CODES) *Let p be a prime number. Then the* Hensel code *of length r of any number $\alpha = (c/d) \cdot p^e \in \mathbb{Q}$ is the pair*

$$(mant_\alpha, exp_\alpha) = (.\ a_0 a_1 \cdots a_{r-1}, e),$$

where the left-most r digits and the value e of the related p-adic expansion are called the mantissa and the exponent, respectively.

□

Note that in particular

$$\sum_{i=0}^{r-1} a_i \cdot p^i \in \mathbb{Z}_{p^r}.$$

Let $\mathbb{H}_{p,r}$ indicate the set of Hensel codes with respect to the prime p and the approximation r and let $H_{p,r}(\alpha)$ indicate the Hensel code representation of the rational number $\alpha = (a/b) \cdot p^e$ with respect to the prime p and the approximation r.

The forward mapping is essentially the application of EEA to d and p^r in order to find $\mid d^{-1} \mid_{p^r}$. Since we isolate the p-part, we can restrict our attention to $\hat{\mathbb{Q}}$ and assume that d and p are relatively prime. Then we can solve the Diophantine equation $p^r \cdot x + d \cdot y = 1$. Then $y = \mid d^{-1} \mid_{p^r}$ because

$$\mid p^r \cdot x + d \cdot y \mid_{p^r} = 1 \pmod{p^r} = \mid d \cdot y \mid_{p^r} .$$

Theorem 2.1 (FORWARD MAPPING) *Given a prime p, an integer r and a rational number $\alpha = (c/d) \cdot p^n$, such that* $\gcd(c,p) = \gcd(d,p) = 1$, *the mantissa $mant_\alpha$ of the code related to the rational number α, is computed by the Extended Euclidean Algorithm (EEA) applied to p^r and d as:*

$$mant_\alpha \equiv c \cdot y \pmod{p^r}$$

where y is the second output of the EEA **applied to** d **and** p^r.

Proof. See [Mio84]. ∎

Let us note that the correspondence between the commutative rings $(\hat{\mathbb{Q}}, +, \cdot)$ and $(\mathbb{H}_{p,r}, +, \cdot)$ is not bijective, since each Hensel code mantissa $.\, a_0 a_1 \cdots a_{r-1}$ $(= \sum_{i=0}^{r-1} a_i \cdot p^i \in \mathbb{Z}_{p^r})$ in $\mathbb{H}_{p,r}$, is the image of an infinite subset of the rational numbers. For this reason we need to define a suitable subset of $\hat{\mathbb{Q}}$, such that the correspondence between this subset and $\mathbb{H}_{p,r}$ is injective.

Definition 2.2 (Farey Fraction Set) *The Farey fraction set $\mathbb{F}_{p,r}$ is the subset of $\hat{\mathbb{Q}}$ such that:*

$$a/b \in \hat{\mathbb{Q}} : \gcd(a,b) = 1$$

and

$$0 \leq a \leq N, \quad 0 < b \leq N, \quad N = \left\lfloor \sqrt{\frac{p^r - 1}{2}} \right\rfloor.$$

$\mathbb{F}_{p,r}$ will also be called the Farey fraction set of order N, as $N = N(p,r)$.

Definition 2.3 *The generalized residue class $\mathbb{Q}_k^\circ$ is the subset of $\hat{\mathbb{Q}}$ defined as follows:*

$$\mathbb{Q}_k^\circ = \{a/b \in \hat{\mathbb{Q}} \ \text{ such that } \ |\, a/b \,|_{p^r} = k\}.$$

□

From the last definition it follows that

$$\hat{\mathbb{Q}} = \bigcup_{k=0}^{p^r-1} \mathbb{Q}_k^\circ.$$

Theorem 2.2 *Let N be the largest integer satisfying the inequality*

$$2N^2 + 1 \leq p^r$$

and let $\mathbb{Q}_k^\circ$ contain the order-N Farey fraction $x = a/b$. Then x is the only order-N Farey fraction in $\mathbb{Q}_k$.

Proof. See [GK84]. ∎

Also the backward mapping is carried out by EEA. In this case we have to solve the following Diophantine equation: $m \cdot x + p^r \cdot y = 1$ for x and y. This means that

$$\frac{m}{p^r} + \frac{y}{x} = \frac{1}{x \cdot p^r} < \frac{1}{x^2},$$

where by hypothesis $x < p^r$, so that we compute an approximation of $\frac{m}{p^r}$. In the sequence of pairs (x_i, y_i) produced by the EEA the result then is found looking for $y_i \in \mathbb{F}_{p^r}$.

Theorem 2.3 (BACKWARD MAPPING) *Given a prime p, an integer r, a positive integer $m \le p^r$ and a rational number $c/d \in \mathbb{F}_{p,r} \subset \hat{\mathbb{Q}}$, let m be the value in $\mathbb{Z}_{p^r}$ of the Hensel code mantissa related to c/d, then the EEA, applied to p^r and m, computes a finite sequence of pairs (x_i, y_i) such that there exists a subscript j for which $x_j/y_j = c/d$.*

Proof. See [Mio84]. ∎

From these considerations we can finally state the following theorem.

Theorem 2.4 *Given a prime p, an approximation r, an arithmetic operator Φ in $\mathbb{Q}$ and the related arithmetic operator Φ' over $\mathbb{H}_{p,r}$. Then for any $\alpha_1, \alpha_2 \in \mathbb{Q}$, if*

$$\alpha_1 \Phi \alpha_2 = \alpha_3, \quad \alpha_3 \in \mathbb{F}_{p,r},$$

there exists precisely one $\beta \in \mathbb{H}_{p,r}$ such that

$$H_{p,r}(\alpha_1)\Phi' H_{p,r}(\alpha_2) = \beta$$

and furthermore $\beta = H_{p,r}(\alpha_3)$.

On this basis, every computation over $\mathbb{H}_{p,r}$ gives a code which is exactly the image of the rational number given by the corresponding computation over $\hat{\mathbb{Q}}$.

A general scheme of computation may consist in mapping on $\mathbb{H}_{p,r}$ the rational numbers given as input to the computation and then performing the computation over $\mathbb{H}_{p,r}$. However, by Theorem 2.3, the inverse mapping can be performed only when the expected result belongs to $\mathbb{F}_{p,r}$.

We note that the choice of order of truncation, as well as the choice of the base p, are made in accordance with an a priori estimation of the magnitude of the solution of the problem. In fact we must identify a suitable set of Farey fractions that contains the rational solution; the choices of p and r are a consequence of this identification.

Such an estimate depends in general on the given algorithm/problem one is interested in. The computation of the estimate may turn out to be a difficult problem. Let us mention some examples.

1. *arithmetic over the rationals:* Let us consider the computation of a^b, where $a \in \mathbb{Q}$ and $b \in \mathbb{Z}$. The number of bits which are necessary to represent the rational result is: $b \cdot \log_2 a$.

2. *algebra of polynomials:* For example, it is easy to compute in advance the maximal coefficient which can be obtained by a polynomial multiplication. In fact: given the polynomials $\sum_{i=0}^{n} a_i \cdot x^i$ e $\sum_{j=0}^{m} b_j \cdot x^j$, if $a = \max\{\mid a_i \mid\}_{1 \le i \le n}$, $b = \max\{\mid b_j \mid\}_{1 \le j \le m}$ and $c = \max\{a, b\}$, then the greatest coefficient of the polynomial result is smaller than $\max\{n, m\} \cdot c^2$.

3. *linear algebra:* For example, it is well known that the determinant $\det(A)$ of an n-dimensional square matrix A, is bounded by $n! \cdot a^n$, where $a = \max\{\mid a_{i,j} \mid\}$, $1 \leq i, j \leq n$.

There is also a class of mathematical problems which are particularly well-suited for being solved by p-adic arithmetic: these are problems which are affected either by overflow during the computations or by ill-condition as we will see with the case study that we are going to analyze and implement.

Below we will discuss in more detail a bound for the solutions of linear equation systems over rational numbers.

For a detailed treatment of the algorithmic aspects of operations on Hensel codes, as well for the treatment of pseudo-Hensel codes, we refer to Limongelli and Pirastu [LP96].

2.2.2 Bounds for the Solutions

As we saw in the previous sections, the computation of a suitable bound for the size of the solutions is a fundamental step of any p-adic algorithm. In our case we consider systems of linear equations over rational numbers. This means that, for a given matrix $A \in \mathbb{Q}^{n \times n}$ and $\vec{b} \in \mathbb{Q}^n$, we need an integer m such that, if a solution vector $\vec{x} = (x_1, \ldots, x_n) \in \mathbb{Q}^n$ of $A\vec{x} = \vec{b}$, exists, then the denominator $\texttt{den}_i$ and the numerator $\texttt{num}_i$ of each entry x_i is bounded by m, viz.

$$|\texttt{den}_i| \leq m, \quad |\texttt{num}_i| \leq m \tag{2.3}$$

For the case $A \in \mathbb{Z}^{n \times n}$ and $\vec{b} \in \mathbb{Z}^n$ such a bound m can be easily computed, for instance, by Cramer's rule. We have in fact

$$x_i = \frac{\det(A_i)}{\det(A)} \tag{2.4}$$

where $\det(A)$ denotes the determinant of A, and A_i is the matrix obtained from A by substituting the ith column by $\vec{b}$. Now let a be a maximal entry in a matrix $M \in \mathbb{Z}^{n \times n}$, then by induction on n one has $\det(M) \leq n!a^n$. From this and (2.4) we obtain that both numerator and denominator of any x_i are bounded by $m := n!a^n$, where a is now a maximal entry in A and $\vec{b}$. From this bound we determine a value for r, such that the result is in $\mathbb{F}_{p,r}$ for a given prime p. From the definition it suffices that

$$n!a^n \leq \left\lfloor \sqrt{\frac{p^r - 1}{2}} \right\rfloor \tag{2.5}$$

Considering the square of both sides of the inequality we obtain $2(n!a^n)^2 + 1 \leq p^r$. This implies $\log_p(2(n!a^n)^2 + 1) \leq \log_p p^r$ or, equivalently,

$$r \geq \log_p(2(n!a^n)^2 + 1) \tag{2.6}$$

Hadamard's inequality (see for instance [Mig83]) gives another bound for the determinant

$$\det(A)^2 \leq \prod_{i=1}^{n} \left(\sum_{j=1}^{n} a_{i,j}^2 \right)^{1/2} \tag{2.7}$$

From this bound the following condition is derived in [GK84]

$$p^r \geq \sum_{i=1}^{n} |b_i| \prod_{i=1}^{n} \left(\sum_{j=1}^{n} a_{i,j}^2 \right) \tag{2.8}$$

In practice both bounds are still conservative, since a smaller choice of p and r is often sufficient. In the general case $A \in \mathbb{Q}^{n \times n}$ and $\vec{b} \in \mathbb{Q}^n$ the bound for the numerator and denominator of the x_i's becomes $n!a^{n(n+1)}$. This follows again from Cramer's rule by considering the equivalent system obtained from A by multiplying each row by the common denominator of all entries in that row and of the ith entry in $\vec{b}$, i.e., multiplying by a number of magnitude of at most a^{n+1}.

2.2.3 The Parallel Algorithm

The parallelization is based on the concurrent application of Gauss' algorithm on several homomorphic p-adic images of the problem.

The multiple homomorphic images technique [Kri85, Lip88] presents the following characteristics:

1. the image domains are simpler than the original domain so that the image problem can be solved more efficiently;

2. the forward mapping preserves the operations in every image domain as stated in Theorem 2.4;

3. the transformation leads to several independent homomorphic image problems each of which can be solved exactly, independently and in parallel as Figure 2.5 shows.

4. the correctness of the recovery step is assured by the Chinese Remainder Algorithm that has been parallelized in [Lim93b, LL93] on the basis of the following theorem taken from [Kri85] (the computation of each summand in (2.9) is done in parallel).

Theorem 2.5 (Chinese Remainder Theorem) *Let $p_1, \ldots, p_k$ be k relatively prime integers > 1. Then for any $s_1, \ldots, s_k (s_i < m_i)$ there is a unique integer s satisfying*

$$s < \prod_{i=1}^{k} p_i =: M$$

and $s_i \equiv s \mod p_i$ *; the integer* s *can be computed using*

$$s = \sum_{i=1}^{k} (\frac{M}{p_i}) s_i T_i \mod M, \tag{2.9}$$

where T_i *is the solution of*

$$(\frac{M}{p_i}) T_i \equiv 1 \mod p_i.$$

Many computer algebra algorithms use the modular approach. There the input is mapped into a homomorphic image, the computation is done in this image and the CRA is performed in one iteration of a big sequential loop. This step is repeated until enough image results have been computed to reconstruct the result in the original domain*.

Since in the sequential approach the CRA is interleaved with the remaining algorithm it is not possible to use a parallelized CRA for computing all necessary Chinese remainderings in one step.

In the following we describe the parallelization algorithm on, say, k concurrent processors, like in Fig. 2.5. Let us note that we have arbitrarily many virtual processors available, which will be automatically mapped and distributed on the real processors.

We first compute k prime numbers $p_1, \ldots, p_k$ at random and the corresponding code length r according to the computed bound, such that the entries of the solution $\vec{x}$ are expected to be in $\mathbb{F}_{g,r}$, where g is the smallest prime number such that $g \geq p_1 \cdots p_k$ (Step **1.1**, Fig. 2.6).

At this point k parallel tasks are started. Each of them computes the image of the problem with respect to one prime in p-adic representation of the rational entries, i.e., $H_{p_i,r}(A)$ and $H_{p_i,r}(\vec{b})$ (Step **1.2**, Fig. 2.6). By a certain abuse of notation we denote by $H_{p_i,r}(A)$ the matrix $(\tilde{a}_{i,j})$ with $\tilde{a}_{i,j} = H_{p_i,r}(a_{i,j})$, and analogously for $H_{p_i,r}(\vec{b})$.

Then for each processor a sequential implementation of Gaussian elimination is executed via p-adic arithmetic (Step **1.3**, Fig. 2.6).

Note that an homomorphic image of the problem may not allow a solution, since, for instance, the determinant of the matrix might be zero modulo the particular prime. In this case the program detects that a prime cannot be used and computes, at random, another prime. Then it applies the same algorithm on the new homomorphic image. Although such a situation implies a considerably longer execution time for the Gauss' algorithm, it turned out that this case did not often arise during our tests.

*One example for such an algorithm is the computation of the gcd over polynomials with the `IPGCDC` algorithm in SACLIB (see [C+93]).

Input		$A, \vec{b}$	
Prime Computation		$\downarrow$	
		$p_1, p_2, \ldots, p_k$	
Forward Mapping	$\swarrow$	$\downarrow$	$\searrow$
Hensel Codes	$H_{p_1,r}(A), H_{p_1,r}(\vec{b})$	$\ldots$	$H_{p_k,r}(A), H_{p_k,r}(\vec{b})$
	$\downarrow$	$\downarrow$	$\downarrow$
Gauss Algorithm	$\downarrow$	$\downarrow$	$\downarrow$
	$\downarrow$	$\downarrow$	$\downarrow$
Solutions	$\vec{x}^{(1)} = H_{p_1,r}(\vec{x})$	$\ldots$	$\vec{x}^{(k)} = H_{p_k,r}(\vec{x})$
Synchronization	$\searrow$	$\downarrow$	$\swarrow$
		$\downarrow$	
		$\vec{x}^{(1)}, \ldots, \vec{x}^{(k)}$	
	$\swarrow$	$\downarrow$	$\searrow$
	$\swarrow$	$\downarrow$	$\searrow$
	$\vec{x}_1^{(1)}, \ldots, \vec{x}_1^{(k)}$	$\ldots$	$\vec{x}_n^{(1)}, \ldots, \vec{x}_n^{(k)}$
CRA (Recovery Step)	$\downarrow$	$\downarrow$	$\downarrow$
Unique Solution	$\vec{x}_1 \in \mathbb{H}_{g,r}$	$\ldots$	$\vec{x}_n \in \mathbb{H}_{g,r}$
Inverse Mapping	$\searrow$	$\downarrow$	$\swarrow$
		$\downarrow$	
Output		$\vec{x}$	

Figure 2.5: Parallel Computation Scheme

Gaussian elimination computes solutions $\vec{x}^{(i)} \in \mathbb{H}^n_{p_i,r}$ for $i = 1, \ldots, k$. After collecting all of the $\vec{x}^{(i)}$ we execute k concurrent calls of CRA that have to be applied to codes with null exponent. In order to do it, we must multiply and shift the codes (Step **1.4**, Fig. 2.6). We apply to each sequence of components $x_j^{(1)}, \ldots, x_j^{(k)}$, obtaining the component $x_j \in \mathbb{H}_{g,r}$ of the solution vector $\vec{x}$ (Step **1.5**).

From the assumptions made on the bound and on r, the list $\{\vec{x}^{(1)}, \ldots, \vec{x}^{(k)}\}$ of results obtained this way can be mapped back to a vector over the Farey fraction set $\vec{x} \in \mathbb{F}^n_{g,r}$ by the EEA. From Theorem 2.4 we know that if the solution exists in $\mathbb{F}_{g,r}$, then it is unique (Step **1.6**).

After this, the result $\vec{x}$ only needs to be converted from the p-adic to the usual representation by the backward mapping, applied in parallel on each component.

In the case of dense matrices with large dimension and size with respect to the number of processors, also standard parallelization techniques for dense linear systems could be applied.

Input: n: degree of the linear system;
$A = (a_{i,j}) \in \mathbb{Q}^{n\times n}$: n-dimensional square matrix;
$\vec{b} = (a_{1,n+1}, \ldots, a_{n,n+1}) \in \mathbb{Q}^n$: column vector;
$p_1, \ldots, p_k$, k prime numbers;

Output: $\vec{x} = (x_1, \ldots, x_n) \in \mathbb{Q}^n$: solution of $A\vec{x} = \vec{b}$, if it exists;

Begin

1.1. (PRIME COMPUTATION)
Compute the maximal integer number a among the numerators and denominators of the rational entries in A and $\vec{b}$; compute the truncation order r, as shown in (2.6); compute the number k of necessary processors; compute g;

start k parallel tasks

1.2. (FORWARD MAPPING)
In each task apply the parallel mapping $H_{p_i,r}$ to all the entries of A and $\vec{b}$ in order to obtain the Hensel Codes;

1.3. (GAUSS' ALGORITHM)
Compute Gauss' algorithm in each domain $\mathbb{H}_{p_i,r}$ in parallel in order to obtain the k vectors solutions: $\vec{x}^{(1)} = H_{p_1,r}(\vec{x}), \ldots, \vec{x}^{(k)} = H_{p_k,r}(\vec{x})$

end k parallel tasks

1.4. (SYNCHRONIZATION)
For each of the k domains, the exponents of the related codes must become zero (in order to apply CRA);

start n parallel tasks

1.5. (CRA)
Apply parallel CRA to each k-tuple $\vec{x}_i^{(1)}, \ldots, \vec{x}_i^{(k)}$

start k parallel tasks

(UNIQUE SOLUTION $\vec{x}_i$ FOR CRA)

end k parallel tasks

and find n solutions $\vec{x}_1 \in \mathbb{H}_{p_1 \cdots p_k, r}, \ldots, \vec{x}_n \in \mathbb{H}_{p_1 \cdots p_k, r}$;

1.6. (INVERSE MAPPING)
Apply the backward mapping to each of the solutions, obtaining $\vec{x} = (x_1, \ldots, x_n)$;

end n parallel tasks

End

Figure 2.6: Parallel Algorithm

Table 2.1: Comparison of sequential and parallel algorithm

Dimension	Input Size	Sequential	Parallel	Speedup
10	10	3064	692	4.4
15	10	9388	1792	5.2
20	10	27707	4421	6.2
20	20	57014	7563	7.5
20	30	69481	11196	6.2
25	10	61528	8751	7.0
25	20	108695	15813	6.8
30	10	119890	16893	7.0

2.2.4 Implementation and Experimental Results

As a parallel environment for our implementation we used PACLIB (see [H^+92]), a system developed at RISC-Linz for parallel computer algebra. PACLIB is based on the SACLIB library (see [C^+93]), which provides several computer algebra algorithms written in C. On the other hand, several other symbolic computation systems provide a parallel implementation of a linear system solver. For instance, we discussed at the beginning of this chapter a solver based on the Gauss-Jordan algorithm implemented in the system ||MAPLE||, that can also handle symbolic entries, and in particular rational numbers.

We performed several tests of our implementation on randomly generated linear systems on a Sequent Symmetry machine with 20 processors, a MIMD computer with shared memory.

The parallel implementation is compared with the corresponding sequential implementation, where we apply sequentially the same mapping onto $\mathbb{H}_{p_i,r}$ for the same primes p_i as in the equivalent parallel execution.

In Table 2.1 the execution times of both the sequential and the parallel implementation are reported in milliseconds. The input size is the maximal bit length of the numbers. If the entries are rational numbers the numerator and the denominator have a bit length bounded by this input size. Parallelizing the algorithm over 10 processors we achieve a speedup up to 7.5, in comparison to the sequential algorithm.

We compared our implementation with an efficient modular parallel implementation for systems over integers which makes use of a mixed method (Gauss and Cramer), implemented in PACLIB. We consider two cases:

1. The input data are integers;
2. The input data are rationals. We present the case where only the vector $\vec{b}$ has rational entries.

Table 2.2: Comparison of modular, p-adic, and rational p-adic when length=10.

Dimension	Modular	p-adic	Rational
5	321	218	278
10	692	619	718
15	1422	1719	1315
20	3210	3315	2995
25	5972	5756	5579
30	14309	12299	11107

In the first case, p-adic arithmetic is essentially reduced to modular arithmetic and the backward mapping becomes almost trivial, so that no improvement is achieved. Since the denominator of any component x_i of the solution $\vec{x}$ divides $\det(A)$. So $H_{g,r}(x_i) \cdot H_{g,r}(\det(A))$ is an integer and no backward mapping is needed, since we have

$$x_i = \frac{/H_{g,r}(x_i)H_{g,r}(\det(A))/_g}{\det(A)} \quad , \tag{2.10}$$

where $/ \cdot /_g$ is the *signed* modulo mapping to $\{-\frac{g-1}{2}, \ldots, \frac{g-1}{2}\}$. Remark that the determinant $\det(A)$ is computed as a direct by-product of Gaussian elimination.

In the second, more interesting case, in order to do a fair comparison with the modular algorithm, which accepts only integers entries, we have to study the size of equivalent inputs for both algorithms. Let $A \in \mathbb{Q}^{n\times n}$ be again the matrix describing a system over the rational numbers and let s be the maximal size among all denominators of the entries in A and $\vec{b}$. We must transform $A\vec{x} = \vec{b}$ to an equivalent system $A'\vec{x} = \vec{b}'$ with integer entries. This means to multiply each row/equation by an appropriate integer. It is easy to see that the smallest integer m_j such that $m_j\vec{a}_j, m_jb_j$ are all integers, is equal to the lcm over all denominators arising in the jth row $\vec{a}_j$ (resp. b_j) of A (resp. $\vec{b}$). In the worst case, m_j will be the product of all denominators (i.e., $m \leq (n+1)s$). In the comparison, one must take into account this fact, although the average size of m will be usually smaller than this.

Here we are considering rational entries only in $\vec{b}$. So, if s is the size of the entries for the p-adic algorithm, the entries of the modular algorithm will be of size $2s$.

In Table 2.2 we show the behaviour of the algorithms for fixed input length. For the considerations stated above a length of 10 for integer entries means a length of 5 for the rational case.

In Table 2.3 the timings of some executions are shown, where the dimension of the system is 20.

We also compare our sequential algorithm with the implementation available in MapleV (see [CFG+86]). MapleV implements a fraction free Gaussian elimination, so

Table 2.3: Comparison of modular, p-adic, and rational p-adic when dim. =20.

Input length	Modular	p-adic	Rational
5	1908	2024	1722
10	3007	2966	2903
15	3519	3957	3416
20	5760	5441	4421
25	7502	8785	7067
30	9845	11151	9037
35	11222	11734	10023
40	22406	21372	11023

the equations are converted to have integer coefficients and after each elimination step, the greatest common divisor of the coefficients is divided out to minimize growth. The timings in Fig. 2.7 show the behaviour of the algorithms.

As expected, the p-adic representation is at least as efficient as the modular one for the case of integer coefficients and more efficient for the case of rational coefficients.

These experimental data confirm the expected behaviour of linear algebra algorithms implemented via p-adic arithmetic as regards the heavy computational complexity of CRA. In [Lim93b] it is shown that for problems with many large input data the asymptotic running time of the p-adic algorithm is never dominated by the cost of the recovering step.

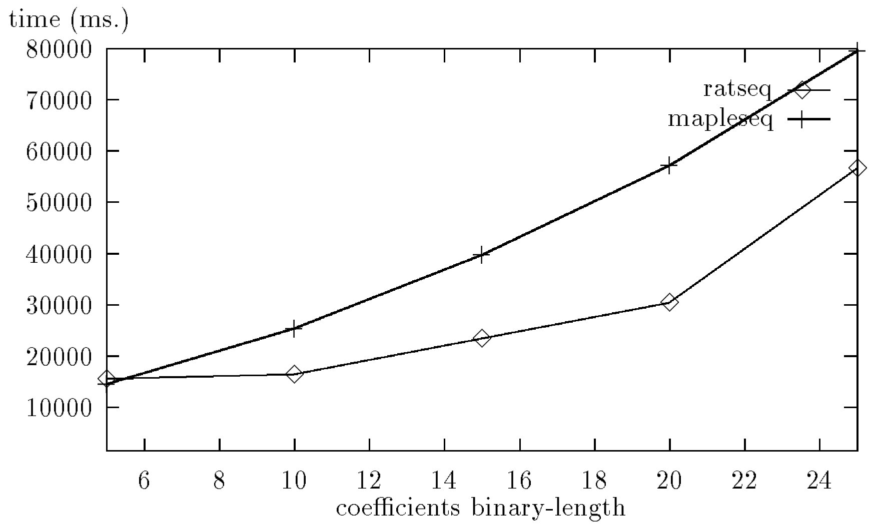

Figure 2.7: Comparison of MapleV and the sequential p-adic algorithm with dim. =20.

3

On the Combinatorial Structure of Umbral Calculus

3.1 Introduction

Umbral calculus essentially can be seen as the study of sequences of polynomials $(p_i(x))_{i\in\mathbb{N}}$ which satisfy a certain convolution properties, namely

$$p_n(x+y) = \sum_i p_i(x)\, p_{n-i}(y) \qquad \text{(for all } n \in \mathbb{N}) \tag{3.1}$$

or similarly

$$p_n(x+y) = \sum_i \binom{n}{i} p_i(x)\, p_{n-i}(y) \qquad \text{(for all } n \in \mathbb{N}) \tag{3.2}$$

Such sequences are called sequences of *integer* or *binomial type*, respectively. Most of the well known polynomial sequences arising in combinatorics or in numerical analysis, like for instance the Bernoulli or the Legendre polynomials, are strictly related to sequences of integer type.

The convolution property of sequences of integer type suggests to make use of a structure of coalgebra on the vector space $\mathcal{K}[x]$ of polynomials, say $\mathcal{C} = (\mathcal{K}[x], \Delta, \varepsilon)$, where the comultiplication Δ is defined in analogy to the substitution $p_n(x) \mapsto p_n(x+y)$, i.e.,

$$\Delta x^n = \sum_i \binom{n}{i} x^i \otimes x^{n-i}$$

and the counit ε is the evaluation at zero: $\varepsilon(p(x)) = p(0)$ (for a precise definition see [JR79] and [NS82], we give a recall in Section 3.8). In this framework umbral calculus is then the study of all linear bases of $\mathcal{K}[x]$ which behave like the standard one $(x^i)_{i\in\mathbb{N}}$ with respect to Δ, or, in other words, which satisfy (3.2). Such bases are in some sense associated to so-called *shift-invariant* operators, i.e., operators which can be expressed as formal power series in the usual differential operator d/dx.

This approach, presented in several articles (see, besides the works cited above, also [CNP84, CNP86, CP84]), already gives an elegant linear algebraic description of umbral calculus.

Our aim in this paper is to give a description starting with less assumptions on the structure of the underlying set, in order to point out which requirements are really necessary for an *umbral* structure. Instead of the polynomial ring we consider a general vector space $\mathbb{V}$ of countably infinite dimension over a field of arbitrary characteristic. The analog of the differential operator on polynomials will be any *sharply nesting* operator on $\mathbb{V}$, as described later. In principle, this is sufficient to develop the whole structure and shows, in our eyes, its *simplicity* in a more direct way.

In particular, this way one easily convinces himself that the structure of all so-called *non-standard* umbral calculi (for instance, the q-analogs presented in [Il81, Kir79, Rom85]) is always the same.

In addition, our approach is completely characteristic free, so that not only polynomials are included into the model. For instance, linear recurrent sequences over finite fields fit into the presented framework as well.

A further aim of this work is to state clearly the connection between the sequences studied in umbral calculus and the so called *recursive matrices*. As a matter of fact, in the last years such matrices (also called Riordan arrays or convolution matrices) found renewed interest (see [Knu93] and [Spr94]). We will derive several properties of the two dimensional sequences defined by recursive matrices and show that they naturally follow from the proposed description.

Future work will concentrate on the algorithmic study of recursive matrices, with a particular attention to their relationship to inverse relations of sequences. From a symbolic computation point of view, several questions are still open.

This chapter reports on joint work with Giorgio Nicoletti.

For a extensive bibliography on umbral calculus we refer to the PhD thesis by A. Di Bucchianico [Buc91], also available on-line as part of the dynamic survey on umbral calculus published by the Electronic Journal of Combinatorics.*

3.2 Summary

In this section we give a short introduction to the whole chapter on umbral calculus.

We will consider as basic structure a vector space $\mathbb{V}$ of countably infinite dimension. Most of the work is concerned with the study of linear functionals on $\mathbb{V}$, i.e., elements of the linear dual $\mathbb{V}^*$, or with endomorphisms in $\mathrm{End}(\mathbb{V})$, i.e., linear operators $S : \mathbb{V} \longrightarrow \mathbb{V}$.

In order to give a meaning to infinite series over operators, in Section 3.3 we introduce the **finite topology** on $\mathbb{V}^*$, which naturally extends to a topology on $\mathrm{End}(\mathbb{V})$. In this topology convergence of sequences and therefore series of linear operators over any index set can be naturally defined. Although the finite topology turns out to describe just what one usually does, we study it in some detail in order to make clear that the topological structure directly follows from the underlying vector space structure. It should be noticed that most of the properties presented in this section still hold when the dimension of $\mathbb{V}$ is not countable. In this case we state the propositions according to the more general situation.

*WWW address `http://ejc.math.gatech.edu:8080/Journal/Surveys/index.html`

On the other hand the reader may, as a matter of fact, skip Section 3.3 if he is not interested in a precise topological justification of the introduced computational framework.

Particularly important in our approach are such operators $S \in \mathrm{End}(\mathbb{V})$, for which the sequence of kernels $\left(\ker S^{i+1}\right)_{i \in \mathbb{N}}$ builds a **nesting sequence**† for $\mathbb{V}$, that is, when $\sum_i \ker S^{i+1} = \mathbb{V}$ and $\ker S^i \subsetneq \ker S^{i+1}$, where we denote by $\sum_i B_i$ the space generated by the union of all B_i's. We then call S a **nesting operator** on $\mathbb{V}$. For the vector space generated by a set B we write $\langle B_i \rangle$, or just $\langle b \rangle$ if $B = \{b\}$.

In the case that the dimension of $\mathbb{V}$ is countably infinite and $\dim\left(\ker S^{i+1} / \ker S^i\right) = 1$ we speak of the **sharply nesting** operator S, which then plays the role of the delta operators in the classic umbral calculus, since for some basis $(\vec{b}_i)_{i \in \mathbb{N}}$ of $\mathbb{V}$ we have $S\vec{b}_0 = \vec{0}$ and $S\vec{b}_{i+1} = \vec{b}_i$. Such a basis is then called S**-compatible**. See the sections 3.5 and 3.6 for detailed proofs and definitions.

Already in this framework it is possible to prove, for instance, that the ring of all S-invariant operators of a sharply nesting S, i.e., all T such that $TS = ST$, is isomorphic to the ring of formal power series in one variable, since such an operator T can be expressed as a series $T = \sum_i t_i S^i$ in the finite topology. In the usual umbral calculus a particular S is fixed and chosen as the differential operator, and the S-invariant T then are the *shift-invariant* operators.

At this point we are almost ready to introduce the central concept of **umbral coalgebra** over $\mathbb{V}$. For a coalgebra $\mathcal{C} = (\mathbb{V}, \Delta, \varepsilon)$ we say that $S \in \mathrm{End}(\mathbb{V})$ is a **hemimorphism** if $\Delta S = (id \otimes S)\Delta$ holds (see Section 3.8). This definition is strongly motivated by the analogy to the behaviour of delta operators with respect to the substitution $p(x) \mapsto p(x+y)$.

In Section 3.9 we define a coalgebra $\mathcal{C}$ to be **umbral** if it admits a sharply nesting hemimorphism S, which turns out to be equivalent to the existence of an **umbral basis** $(\vec{b}_i)_{i \in \mathbb{N}}$, that is, a basis for which

$$\Delta \vec{b}_n = \sum_i \vec{b}_i \otimes \vec{b}_{n-i}$$

holds. As it is to be expected, such a basis will be S-compatible for some sharply nesting hemimorphism S and corresponds to a *basic sequence* of polynomials in the usual umbral calculus (S then corresponds to the associated delta operator).

We want to point out that the behaviour of a sharply nesting hemimorphism S with respect to the comultiplication Δ can be described in a natural, in essence combinatorial way. This is based on the property of S to act like a *shift* on any S-compatible basis $(\vec{b}_i)_{i \in \mathbb{N}}$. If we represent an element in $\mathbb{V} \otimes \mathbb{V}$ by the bi-infinite tableau of coordinates with respect to the basis $\left(\vec{b}_i \otimes \vec{b}_j\right)_{i,j \in \mathbb{N}}$, then $(id \otimes S)$ (or $(S \otimes id)$, respectively) acts like a *shift to the top* (or *to the left*, respectively) on this tableau (see Section 3.8).

†In other contexts, like, e.g., in projective geometry, such a structure is usually called a *flag*.

This point of view allows us to give most of the proofs in a more pictorial, and hopefully more understandable way. Most of the basic properties of umbral sequences are shown in this framework, see Section 3.10.

In Section 3.11 we study the matrix representation of automorphisms U of $\mathcal{C}$. Such an operator U maps umbral bases into umbral bases and is called **umbral operator**. The framework described here allows to derive without any extra effort that the matrix with respect to the umbral basis $(\vec{p}_i)_{i\in\mathbb{N}}$ representing such a transformation $(\vec{p}_i)_{i\in\mathbb{N}} \mapsto (\vec{q}_i)_{i\in\mathbb{N}}$ is a **recursive matrix** (in the sense of [BBN82], or a convolution matrix, like other authors say), see Section 3.12. Since the matrices corresponding to U and to the inverse U^{-1} are inverse to each other, they describe an inverse relation on arbitrary sequences. In this context properties of the matrices (and so, of the inverse relations as well) can be derived from the study of some natural questions about the operators involved.

In Section 3.13 we present the *factorial functions* (introduced in [BBN86]) as an example of a umbral structure which is not isomorphic to the polynomial coalgebra.

3.3 The Finite Topology on $\mathbb{V}^*$

Let $\mathbb{V}$ be a vector space with infinite, not necessarily countable dimension. In this paper all vector spaces are meant over some arbitrary but fixed field $\mathcal{K}$ of arbitrary characteristic, if no particular field is specified. The set $\mathbb{N}$ is the set of nonnegative numbers, while $\mathbb{N}^+ = \mathbb{N}\setminus\{0\}$. For the standard topological notions we refer to [Bou89].

We define the *finite topology* on the *linear dual* $\mathbb{V}^*$ of $\mathbb{V}$ by a certain basis of neighbourhoods of zero. This topology naturally extends to a topology on $\mathrm{End}(\mathbb{V})$. We show later that concepts like summability and formal power series of operators and linear functionals on polynomials arising in the umbral calculus are based on this topology. We point out that the finite topology does not depend on the chosen base of $\mathbb{V}$ and does not even require that $\mathbb{V}$ has countable dimension, since it can be defined on any vector space $\mathbb{V}$. It is no additional effort to define series in $\mathbb{V}^*$ over an arbitrary index set.

Consider the set $\mathfrak{W}^0$ of subspaces of $\mathbb{V}^*$ defined as follows

$$\mathfrak{W}^0 := \left\{\mathbb{X}^0 \; ; \; \mathbb{X} \text{ is finite-dimensional subspace of } \mathbb{V}\right\} \tag{3.3}$$

where $\mathbb{X}^0$ denotes the *annihilator* of $\mathbb{X}$, i.e. $\mathbb{X}^0 := \left\{f \in \mathbb{V}^* \; ; \; f(\mathbb{X}) = \{\vec{0}\}\right\}$. Then $\mathfrak{W}^0$ defines a topology on $\mathbb{V}^*$, as the following proposition states.

Proposition 3.1 *The family $\mathfrak{W}^0$ is a basis of neighbourhoods of zero for a Hausdorff vector topology on $\mathbb{V}^*$.*

Proof. Since all elements $\mathbb{X}^0 \in \mathfrak{W}^0$ are vector spaces, we have that $\mathbb{X}^0 - f = \mathbb{X}^0$ for all $f \in \mathbb{X}^0$. So, if $\mathbb{X}^0$ is a neighbourhood of zero, then it is also a neighbourhood of all its elements. Furthermore any finite intersection of elements from $\mathfrak{W}^0$ is in $\mathfrak{W}^0$. In fact, it is easy to prove that for any finite sequence $\mathbb{X}_1, \ldots, \mathbb{X}_n \in \mathfrak{W}^0$ the following holds

$$\mathbb{X}_1^0 \cap \cdots \cap \mathbb{X}_n^0 = \left(\sum_{i=1}^{n} \mathbb{X}_i \right)^0$$

Since $\sum_{i=1}^{n} \mathbb{X}_i$ is again a finite-dimensional subspace of $\mathbb{V}$, we have $\mathbb{X}_1^0 \cap \cdots \cap \mathbb{X}_n^0 \in \mathfrak{W}^0$. This shows that $\mathfrak{W}^0$ is a suitable basis of neighbourhoods of zero for a vector topology.

Now we prove that the vector topology defined by $\mathfrak{W}^0$ is Hausdorff. Let f and g be distinct elements of $\mathbb{V}^*$. Since $f \neq g$ there exists $\vec{v}$ in $\mathbb{V}$ such that $f(\vec{v}) \neq g(\vec{v})$. Consider now the neighbourhoods $f + \langle \vec{v} \rangle^0$ and $g + \langle \vec{v} \rangle^0$ and assume that there is a linear functional h in $\left(f + \langle \vec{v} \rangle^0\right) \cap \left(g + \langle \vec{v} \rangle^0\right)$. This implies that $h(\vec{v}) = f(\vec{v})$ and $h(\vec{v}) = g(\vec{v})$, which gives a contradiction to $f(\vec{v}) \neq g(\vec{v})$. This proves that any distinct f and g have disjoint neighbourhoods, so the topology is Hausdorff. ∎

Definition 3.1 *We call* **finite topology** *on $\mathbb{V}^*$ the topology defined by $\mathfrak{W}^0$ as a basis of neighbourhoods of zero.*

The fact that the finite topology is Hausdorff implies that any limit point over any filter is unique and, in particular, that any sequence has at most one limit point.

We define the concept of *series* in a more general way than we will later need in our investigations, i.e., allowing any infinite, not necessarily countable set of indices. Let I be in the following any infinite set of indices. The family $\mathfrak{F}(I)$ of all finite subsets of I is a directed set with respect to set inclusion. For this reason I-indexed series can be associated to generalized sequences over $\mathfrak{F}(I)$ and their sums can be defined in the finite topology as follows.

Definition 3.2 *Let $\phi := (\varphi_i)_{i \in I}$ be an I-indexed family of functionals in $\mathbb{V}^*$. For any finite subset J of I the functional*

$$\varphi_J := \sum_{j \in J} \varphi_j$$

is called the **finite partial sum** *of ϕ with respect to J.*

The family $(\varphi_J)_{J \in \mathfrak{F}(I)}$ of all finite partial sums of ϕ is then a generalized sequence and is called the **series associated to** *ϕ, denoted by $\sum_{i \in I} \varphi_i$.*

The family ϕ is said to be **summable** *if the series $\sum_{i \in I} \varphi_i$ converges in the finite topology to a (unique) functional φ. In this case φ is called the* **sum** *of the series $\sum_{i \in I} \varphi_i$ (or, equivalently, of ϕ).*

Note that the definition of series does not require any ordering of the index set I, and that the limit φ, if it exists, is unique.

Due to the structure of the finite topology on $\mathbb{V}^*$, a series $\sum_{i \in I} \varphi_i$ is summable if, and only if for all $\vec{v} \in \mathbb{V}$ we have $\varphi_i(\vec{v}) = 0$ for almost all $i \in I$. In the following lemma several characterizations of this property are shown, justifying the choice of the name *finite* for this topology.

Lemma 3.1 *Let $(\varphi_i)_{i\in I}$ be an I-indexed family of functionals in $\mathbb{V}^*$ and $\varphi \in \mathbb{V}^*$. Then the following statements are equivalent.*

1. *The family $(\varphi_i)_{i\in I}$ is summable and $\varphi = \sum_{i\in I} \varphi_i$.*
2. *For every finite dimensional subspace $\mathbb{X}$ of $\mathbb{V}$ there exists a finite subset $J_{\mathbb{X}}$ of I such that for all $J \in \mathfrak{F}(I)$ we have*

$$J_{\mathbb{X}} \subseteq J \implies \varphi_J \in \varphi + \mathbb{X}^0$$

3. *For every finite dimensional subspace $\mathbb{X}$ of $\mathbb{V}$ the set*

$$J'_{\mathbb{X}} := \{i \in I \; ; \; \varphi_i \notin \mathbb{X}^0\}$$

is finite and $\varphi_{J'_{\mathbb{X}}}(\vec{v}) = \varphi(\vec{v})$ for all $\vec{v} \in \mathbb{X}$.

4. *For every $\vec{v} \in \mathbb{V}$ the set*

$$J_{\vec{v}} := \{i \in I \; ; \quad \varphi_i(\vec{v}) \neq 0\}$$

is finite and $\varphi_{J_{\vec{v}}}(\vec{v}) = \varphi(\vec{v})$.

Proof. $1 \Longleftrightarrow 2$: Condition 2 is just a reformulation of the definition of convergence for a generalized sequence. Recall that the neighbourhoods of zero in our topology are precisely those of the form $\mathbb{X}^0$ for $\mathbb{X}$ a finite dimensional vector subspace of $\mathbb{V}$.

$3 \Longleftrightarrow 4$: Condition 4 directly follows from condition 3 by considering $\mathbb{X} = \langle \vec{v} \rangle$. Assume now that Condition 4 holds. Consider an arbitrary $\mathbb{X} \subseteq \mathbb{V}$ with finite dimension and let $\{\vec{v}_0, \ldots, \vec{v}_n\}$ be a basis for $\mathbb{X}$. Then it is easy to see that $J'_{\mathbb{X}} = J_{\vec{v}_0} \cup \cdots \cup J_{\vec{v}_n}$ and condition 3 holds.

$3 \Longrightarrow 2$: Assume that Condition 3 holds. We prove that for $J_{\mathbb{X}} = J'_{\mathbb{X}}$ we have $J_{\mathbb{X}} \subseteq J \implies \varphi_J \in \varphi + \mathbb{X}^0$ for all $J \in \mathfrak{F}(I)$. Let now J be arbitrary finite subset of I with $J'_{\mathbb{X}} \subseteq J$. Observe that for all $j \in J \setminus J'_{\mathbb{X}}$ we have $\varphi_j \in \mathbb{X}^0$. This implies

$$(\varphi - \varphi_J)(\vec{v}) = \varphi(\vec{v}) - \sum_{j\in J'_{\mathbb{X}}} \varphi_j(\vec{v}) - \sum_{j\in J\setminus J'_{\mathbb{X}}} \varphi_j(\vec{v}) = \sum_{j\in J\setminus J'_{\mathbb{X}}} \varphi_j(\vec{v}) = 0$$

for all $\vec{v} \in \mathbb{X}$, so $\varphi - \varphi_J \in \mathbb{X}^0$.

$3 \Longleftarrow 2$: Let $\mathbb{X}$ be any finite dimensional subspace of $\mathbb{V}$ and $J_{\mathbb{X}}$ like in Condition 2. Assume that Condition 2 holds and consider then $J'_{\mathbb{X}}$ from Condition 3. For any $j \in J'_{\mathbb{X}}$ such that $j \notin J_{\mathbb{X}}$ we have $J_{\mathbb{X}} \subseteq J_{\mathbb{X}} \cup \{j\}$. By Condition 2 this implies that $\varphi_{J_{\mathbb{X}}} + \varphi_j \in \varphi + \mathbb{X}^0$, which is in contradiction to $\varphi_j \notin \mathbb{X}^0$. So, $J'_{\mathbb{X}} \subseteq J_{\mathbb{X}}$ and, in particular, $J'_{\mathbb{X}}$ is finite. Furthermore, for all $j \in J_{\mathbb{X}} \setminus J'_{\mathbb{X}}$ we have $\varphi_j \in \mathbb{X}^0$. This means that $\varphi_{J'_{\mathbb{X}}}(\vec{v}) = \varphi_{J_{\mathbb{X}}}(\vec{v}) = \varphi(\vec{v})$ for all $\vec{v} \in \mathbb{X}$. ∎

From now on, let $\mathcal{I}$ be a suitable index set for a basis of $\mathbb{V}$ and consider a basis $\mathcal{B} := \{\vec{b}_i;\ i \in \mathcal{I}\}$ of $\mathbb{V}$. We denote by β^i the linear functional defined by

$$\beta^i(\vec{b}_j) = \delta_{ij}$$

where δ_{ij} is Kronecker's symbol. Then the set $\{\beta^i;\ i \in \mathcal{I}\}$ forms a *pseudobasis* for $\mathbb{V}^*$, as explained in the following.

Proposition 3.2 *Let $\mathcal{B} = \{\vec{b}_i;\ i \in \mathcal{I}\}$ be a basis for $\mathbb{V}$. Then the family $\mathcal{B}^* := \{\beta^i;\ i \in \mathcal{I}\}$ is a* **pseudobasis** *for $\mathbb{V}^*$, i.e., every functional $\varphi \in \mathbb{V}^*$ can be written in a unique way as*

$$\varphi = \sum_{i \in \mathcal{I}} c_i \beta^i$$

Moreover, for every $i \in \mathcal{I}$ we have

$$c_i = \varphi(\vec{b}_i)$$

Proof. We prove that the sum

$$\varphi = \sum_{i \in \mathcal{I}} \varphi(\vec{b}_i)\beta^i$$

exists and converges to φ. From Condition 4 of Lemma 3.1 it is sufficient to prove that for every $\vec{b}_j \in \mathcal{B}$ the set $J_{\vec{b}_j} = \left\{ i \in \mathcal{I}\ ;\ \ \beta^i(\vec{b}_j) \neq 0 \right\}$ is finite and $\varphi_{J_{\vec{b}_j}}(\vec{b}_j) = \varphi(\vec{b}_j)$. From the definition of β^i it is evident that $J_{\vec{b}_j} = \{j\}$ and $\varphi(\vec{b}_j) = \varphi(\vec{b}_j)\beta^j(\vec{b}_j) = \varphi_{J_{\vec{b}_j}}(\vec{b}_j)$. ∎

Definition 3.3 *For all bases $\mathcal{B}$ of $\mathbb{V}$ we call $\mathcal{B}^*$ the* **dual pseudobasis** *of $\mathcal{B}$.*

At this point it is natural to study the structure of the set of operators on $\mathbb{V}^*$, which are continuous with respect to the finite topology. We will see in the next section that the adjoint map provides an isomorphism between the operators on $\mathbb{V}$ and the set of such operators.

Let $\mathrm{End}_C(\mathbb{V}^*)$ denote the space of all endomorphisms of $\mathbb{V}^*$ which are continuous in the finite topology and T be an operator in $\mathrm{End}_C(\mathbb{V}^*)$. For any series $(\varphi_J)_{J \in \mathfrak{F}(\mathcal{I})}$ converging to φ we have that $(T\varphi_J)_{J \in \mathfrak{F}(\mathcal{I})}$ converges to $T\varphi$. For this reason if $(\varphi_i)_{i \in \mathcal{I}}$ is summable, then so is $(T\varphi_i)_{i \in \mathcal{I}}$. In other words the following lemma holds.

Lemma 3.2 *Let T be a continuous operator on $\mathbb{V}^*$ and $(\varphi_i)_{i \in \mathcal{I}}$ a family in $\mathbb{V}^*$. Then $(\varphi_i)_{i \in \mathcal{I}}$ is summable if and only if $(T\varphi_i)_{i \in \mathcal{I}}$ is summable. Furthermore, in this case we have*

$$\sum_{i \in \mathcal{I}} T\varphi_i = T\left(\sum_{i \in \mathcal{I}} \varphi_i \right)$$

Example 3.1 *Consider the linear functional φ defined by $\varphi(\vec{b}_i) = 1$ for all $i \in \mathcal{I}$. From Proposition 3.2 we have*

$$\varphi = \sum_{i \in \mathcal{I}} \beta^i$$

If $T \in \mathrm{End}_C(\mathbb{V}^*)$, then $T\varphi = \sum_{i\in\mathcal{I}} T\beta^i$. In particular, this implies that for all $\vec{v}$ in $\mathbb{V}$ we have $T\beta^i(\vec{v}) = 0$ for almost all i in $\mathcal{I}$. We will need this remark in the proof of Proposition 3.3.

Our main attention in the rest of this work will be devoted to the algebra of $\mathrm{End}(\mathbb{V})$ of operators on $\mathbb{V}$. An interesting property of the finite topology on $\mathbb{V}^*$ is that it naturally extends to a topology on $\mathrm{End}(\mathbb{V})$, providing a framework for handling series and sums of endomorphisms in the following sections. It is easy to see that: The set of subspaces of $\mathrm{End}(\mathbb{V})$

$$\{\mathbb{X}^\perp; \mathbb{X} \text{ is finite-dimensional subspace of } \mathbb{V}\}$$

where

$$\mathbb{X}^\perp = \{S \in \mathrm{End}(\mathbb{V}); \mathbb{X} \subseteq \ker S\}$$

forms a basis for the neighbourhoods of zero in a vector topology on $\mathrm{End}(\mathbb{V})$, which we also call **finite topology on** $\mathrm{End}(\mathbb{V})$. Families $(S_i)_{i\in\mathcal{I}}$ of operators on $\mathbb{V}$ are said to be summable in analogy to Definition 3.2 and the corresponding version of Lemma 3.1 holds.

3.4 Adjoint Operators

The finite topology on $\mathbb{V}^*$ shows an interesting relationship between the endomorphisms of $\mathbb{V}$ and the continuous endomorphisms of $\mathbb{V}^*$. We prove in Proposition 3.3 that the space $\mathrm{End}(\mathbb{V})$ is isomorphic to $\mathrm{End}_C(\mathbb{V}^*)$. This enables us to define a finite topology on $\mathrm{End}_C(\mathbb{V}^*)$ as the image of $\mathrm{End}(\mathbb{V})$.

Recall that the **adjoint map**

$$\mathrm{End}(\mathbb{V}) \longrightarrow \mathrm{End}(\mathbb{V}^*), \quad S \longmapsto S^*$$

is defined by $S^*\varphi := \varphi \circ S$ for all $\varphi \in \mathbb{V}^*$. The operator S^* is then called the **adjoint operator** of S.

Notice that the map is well-defined and linear. Furthermore one has that $(S\circ T)^* = T^* \circ S^*$, and $(S^{-1})^* = (S^*)^{-1}$ if S is invertible. This adjoint map turns out to be an isomorphism between $\mathrm{End}(\mathbb{V})$ and $\mathrm{End}_C(\mathbb{V}^*)$, as we state in the following proposition.

Proposition 3.3 *Let $\mathbb{V}$ be a vector space with infinite dimension. Then the adjoint map is a vector space isomorphism between* $\mathrm{End}(\mathbb{V})$ *and* $\mathrm{End}_C(\mathbb{V}^*)$.

Proof. We first prove that for all $S \in \mathrm{End}(\mathbb{V})$ the adjoint S^* is continuous, then we prove that the adjoint map is injective and surjective. The fact that the adjoint map is compatible with the vector space structure follows directly from the definition.

Let S be an operator from $\mathrm{End}(\mathbb{V})$. Since the set $\mathfrak{W}^0$ is a basis for the finite topology we only have to prove that $\mathbb{X}^0 \in \mathfrak{W}^0$ implies $(S^*)^{-1}(\mathbb{X}^0) \in \mathfrak{W}^0$. For all $\varphi \in \mathbb{V}^*$ and $\mathbb{X}^0 \in \mathfrak{W}^0$ we have

$$\begin{aligned}
\varphi \in (S^*)^{-1}\left(\mathbb{X}^0\right) &\Leftrightarrow \exists \psi \in \mathbb{X}^0 : \psi = S^*\varphi \Leftrightarrow S^*\varphi(\mathbb{X}) = \{\vec{0}\} \\
&\Leftrightarrow \varphi \circ S(\mathbb{X}) = \{\vec{0}\} \Leftrightarrow S(\mathbb{X}) \subseteq \ker \varphi \\
&\Leftrightarrow \varphi \in \left(S\left(\mathbb{X}\right)\right)^0
\end{aligned}$$

Since $S\left(\mathbb{X}\right)$ obviously has finite dimension, this proves $(S^*)^{-1}(\mathbb{X}^0) = \left(S\left(\mathbb{X}\right)\right)^0 \in \mathfrak{W}^0$.

Let $\zeta \in \mathbb{V}^*$ denote the trivial functional $\zeta(\vec{v}) = 0$ for all $\vec{v} \in \mathbb{V}$. Consider any operator S such that $S^*\varphi = \zeta$ for all $\varphi \in \mathbb{V}^*$. Since $S^*\varphi(\vec{v}) = \varphi \circ S(\vec{v}) = 0$ for all $\varphi \in \mathbb{V}^*$ and $\vec{v} \in \mathbb{V}$, it must hold $S(\vec{v}) = \vec{0}$ for all $\vec{v} \in \mathbb{V}$. This proves that the adjoint map is injective.

We prove that the adjoint map is surjective. To this purpose, let $T \in \mathrm{End}_C(\mathbb{V}^*)$. We have to find an $S \in \mathrm{End}(\mathbb{V})$ such that $T = S^*$. Consider a basis $\{\vec{b}_i;\ i \in \mathcal{I}\}$ of $\mathbb{V}$, then we claim that for the operator defined by

$$S(\vec{b}_j) := \sum_{i \in \mathcal{I}} T\beta^i(\vec{b}_j)\vec{b}_i$$

for all $j \in \mathcal{I}$ we have $T = S^*$. Remark that S is well-defined because T is continuous in the finite topology, so $T\beta^i(\vec{b}_j) = 0$ for almost all i and the sum at the right-hand side of the definition is a finite one. We have to check that for every $\varphi \in \mathbb{V}^*$ we have $T\varphi = \varphi \circ S$. From Proposition 3.2 we can write $\varphi = \sum_{i \in \mathcal{I}} \varphi(\vec{b}_i)\beta^i$. Then we have for all $j \in \mathcal{I}$

$$\begin{aligned}
T\varphi(\vec{b}_j) &= T\left(\sum_{i \in \mathcal{I}} \varphi(\vec{b}_i)\beta^i\right)(\vec{b}_j) = \sum_{i \in \mathcal{I}} \varphi(\vec{b}_i)\left(T\beta^i(\vec{b}_j)\right) \\
&= \varphi\left(\sum_{i \in \mathcal{I}} T\beta^i(\vec{b}_j)\vec{b}_i\right) = \varphi \circ S(\vec{b}_j)
\end{aligned}$$

This completes the proof of the proposition. ∎

Note that, since the adjoint map is a vector space isomorphism, the finite topology on $\mathrm{End}(\mathbb{V})$ induces a topology on the image $\mathrm{End}_C(\mathbb{V}^*)$.

3.5 Nested Vector Spaces

The umbral calculus is mainly based on the structure of a *nested vector space*. This concept is directly related to the notion of graded vector space.

Definition 3.4 *A pair* $\mathcal{V} = (\mathbb{V}, (\mathbb{V}_i)_{i\in\mathbb{N}})$ *is a* **nested vector space** *if*

1. $\mathbb{V}_i$ *is a subspace of* $\mathbb{V}$ *for all* $i \in \mathbb{N}$, *and*
2. $\dim \mathbb{V}_0 > 0$, *and*
3. $\mathbb{V}_i \subsetneq \mathbb{V}_{i+1}$ *for all* $i \in \mathbb{N}$, *and*
4. $\mathbb{V} = \sum_i \mathbb{V}_i$

In this case $(\mathbb{V}_i)_{i\in\mathbb{N}}$ *is the* **nesting sequence** *of* $\mathcal{V}$. *A nested vector space* $\mathcal{V} = (\mathbb{V}, (\mathbb{V}_i)_{i\in\mathbb{N}})$ *is* **sharply nested** *if* $\dim \mathbb{V}_i = i+1$ *holds for all* $i \in \mathbb{N}$.

In other words, a nested vector space $\mathcal{V}$ is sharply nested if $\dim \mathbb{V}_{i+1}/\mathbb{V}_i = 1$ for all $i \in \mathbb{N}$ and $\dim \mathbb{V}_0 = 1$. In the following, $\mathcal{V}$ denotes $\mathcal{V} = (\mathbb{V}, (\mathbb{V}_i)_{i\in\mathbb{N}})$ if the context is unambiguous. We explicitly observe that a nested vector space must have infinite, and a sharply nested vector space countably infinite dimension.

The bases of $\mathcal{V}$ which satisfy the nesting property play an important role in the following.

Definition 3.5 *A sequence* $\mathcal{B} = (\mathcal{B}_i)_{i\in\mathbb{N}}$ *is a* **nesting basis** *for a nested vector space* $\mathcal{V}$ *if* $\mathcal{B}_i$ *is a basis of* V_i *and* $\mathcal{B}_i \subsetneq \mathcal{B}_{i+1}$ *for all* $i \in \mathbb{N}$.

From the definition it follows in particular that $\bigcup_{i\in\mathbb{N}} \mathcal{B}_i$ is a basis for $\mathbb{V}$. In addition, it is easy to show that a nesting basis exists for any nested vector space:

Proposition 3.4 *For any nested vector space* $\mathcal{V}$ *there is a nesting basis* $\mathcal{B}$ *for* $\mathcal{V}$.

Proof. We define the components of $\mathcal{B} = (\mathcal{B}_i)_{i\in\mathbb{N}}$ inductively on i as follows. Since $\dim \mathbb{V}_0 > 0$ we define $\mathcal{B}_0$ as a basis of $\mathbb{V}_0$. Assume now that for any $n \in \mathbb{N}$ we already determined $\mathcal{B}_n$ as basis of $\mathbb{V}_n$.

Since $\mathbb{V}_n$ is a subspace of $\mathbb{V}_{n+1}$, $\mathcal{B}_n$ is an independent set of vectors in $\mathbb{V}_{n+1}$ and it can be extended to a basis $\mathcal{B}_{n+1}$ of $\mathbb{V}_{n+1}$. This proves the statement by induction. ∎

If $\mathbb{V}$ is sharply nested , then $\dim(\mathbb{V}_{i+1}/\mathbb{V}_i) = 1$. This means that $\mathrm{Card}(\mathcal{B}_{i+1} \setminus \mathcal{B}_i) = 1$, say $\mathcal{B}_0 = \{\vec{b}_0\}$ and $\mathcal{B}_{i+1} \setminus \mathcal{B}_i = \{\vec{b}_{i+1}\}$ for all $i \in \mathbb{N}$ and we write for simplicity $\mathcal{B} = (\vec{b}_i)_{i\in\mathbb{N}}$.

The proof of Proposition 3.4 shows the difference between a graded and a nested vector space. The definition of a graded vector space would consider a particular choice of the possible extension $\mathbb{V}_n^C$ of $\mathbb{V}_n$ such that $\mathbb{V}_{n+1} = \mathbb{V}_n \oplus \mathbb{V}_n^C$, while the nesting structure does not need this additional information. In the following proposition we explain this relationship.

Proposition 3.5 *Let* $\mathcal{V} = (\mathbb{V}, (\mathbb{V}_i)_{i\in\mathbb{N}})$ *be a nested vector space. Then there exists a sequence* $(\mathbb{W}_i)_{i\in\mathbb{N}}$ *of subspaces of* $\mathbb{V}$ *such that*

$$\mathbb{V} = \bigoplus_{i\in\mathbb{N}} \mathbb{W}_i \qquad \textit{and} \qquad \mathbb{V}_i = \bigoplus_{j\leq i} \mathbb{W}_j \qquad (\forall i \in \mathbb{N}) \tag{3.4}$$

Furthermore for any such sequence we have $\mathbb{W}_0 = \mathbb{V}_0$ *and for all* $i \in \mathbb{N}$

$$\mathbb{W}_{i+1} \cong \mathbb{V}_{i+1}/\mathbb{V}_i \tag{3.5}$$

Proof. Let $\mathcal{V} = (\mathbb{V}, (\mathbb{V}_i)_{i\in\mathbb{N}})$ be a nested vector space. By Proposition 3.4 there exists a nesting basis $\mathcal{B} = (\mathcal{B}_i)_{i\in\mathbb{N}}$ with respect to $(\mathbb{V}_i)_{i\in\mathbb{N}}$. Define $\mathbb{W}_0 = \langle \mathcal{B}_0 \rangle$ and $\mathbb{W}_{i+1} = \langle \mathcal{B}_{i+1} \setminus \mathcal{B}_i \rangle$ for all $i \in \mathbb{N}$. Obviously, $\mathbb{W}_i$ is a subvector space of $\mathbb{V}$ for all $i \in \mathbb{N}$. It is easy to see that from the definition of a nesting basis follows that $\mathcal{B}_n = \mathcal{B}_0 \dot\cup (\mathcal{B}_1 \setminus \mathcal{B}_0) \dot\cup \cdots \dot\cup (\mathcal{B}_n \setminus \mathcal{B}_{n-1})$, where $\dot\cup$ denotes the disjoint union of sets. So, $\mathbb{V}_n = \bigoplus_{j\le n} \mathbb{W}_j$ holds. Furthermore, since $\bigcup_{i\in\mathbb{N}} \mathcal{B}_i$ is a basis for the whole $\mathbb{V}$ we have $\mathbb{V} = \bigoplus_{i\in\mathbb{N}} \mathbb{W}_i$. Property (3.5) directly follows from (3.4). ∎

Example 3.2 *For umbral calculus the most interesting example is the vector space of univariate polynomials over* $\mathcal{K}$*, denoted by* $\mathcal{K}[x]$*. For* $\mathbb{V} = \mathcal{K}[x]$ *we can define*

$$\mathbb{V}_i = \{p(x) \in \mathcal{K}[x] \; ; \; \deg p \le i\}$$

where $\deg p$ *denotes the polynomial degree of* p*. Then* $\mathcal{V} = (\mathbb{V}, (\mathbb{V}_i)_{i\in\mathbb{N}})$ *is a sharply nested vector space and a nesting basis is given, for instance, by* $(x^i)_{i\in\mathbb{N}}$*. On the other hand, several grading sequences* $(\mathbb{W}_i)_{i\in\mathbb{N}}$ *as in Proposition 3.5 are possible for* $\mathcal{V}$*. Consider, for instance, the possibilities given by* $\mathbb{W}_i = \langle p_i(x) \rangle$ *for any* $p_i(x) \in \mathcal{K}[x]$ *of degree* i*.*

3.6 Nesting Operators

From the concept of a nested vector space it is straightforward to ask for particular operators which induce a nesting structure on the space. The concept of a **nesting operator** on a vector space is introduced here. Attention is given to computational methods for determining nesting bases with particular properties with respect to such operators.

Definition 3.6 *Let* $\mathbb{V}$ *be an infinite-dimensional vector space. A linear operator* S *on* $\mathbb{V}$ *is a* **nesting operator** *(resp.* **sharply nesting operator***) if* $\mathcal{V} = \left(\mathbb{V}, \left(\ker S^{i+1}\right)_{i\in\mathbb{N}}\right)$ *is a nested (resp. sharply nested) vector space. For any nesting operator* S *a nesting basis* $\mathcal{B} = (\mathcal{B}_i)_{i\in\mathbb{N}}$ *for* $\mathcal{V} = \left(\mathbb{V}, \left(\ker S^{i+1}\right)_{i\in\mathbb{N}}\right)$*, is called an* S**-nesting basis***.*

Observe that a nesting operator S is sharply nesting if $\dim\left(\ker S^{i+1}/\ker S^i\right) = 1$ holds for all $i \in \mathbb{N}$.

Definition 3.7 *Let* $\mathbb{V}$ *be an infinite-dimensional vector space and* S *a nesting operator on* $\mathbb{V}$*. Then an* S*-nesting basis* $\mathcal{B} = (\mathcal{B}_i)_{i\in\mathbb{N}}$ *of* $\mathcal{V} = \left(\mathbb{V}, \left(\ker S^{i+1}\right)_{i\in\mathbb{N}}\right)$ *is called* S**-compatible basis** *if* $S\mathcal{B}_0 = \{\vec{0}\}$ *and* $S\mathcal{B}_{i+1} = \mathcal{B}_i \cup \{\vec{0}\}$ *for all* $i \in \mathbb{N}$*.*

Equivalently, an S-nesting basis $\mathcal{B} = (\mathcal{B}_i)_{i\in\mathbb{N}}$ is S-compatible if $S\mathcal{B}_0 = \{\vec{0}\}$ and $S(\mathcal{B}_{i+1} \setminus \mathcal{B}_i) = \mathcal{B}_i \setminus \mathcal{B}_{i-1}$.

The case of a sharply nesting S is particularly important for umbral calculus. In this situation an S-nesting basis $\mathcal{B} = (\mathcal{B}_i)_{i\in\mathbb{N}}$ has the form $\mathcal{B}_i = \{\vec{b}_0, \vec{b}_1, \ldots, \vec{b}_i\}$ for some sequence $(\vec{b}_i)_{i\in\mathbb{N}}$. This means that $\mathcal{B}$ is S-compatible if $S(\vec{b}_0) = \vec{0}$ and $S(\vec{b}_{i+1}) = \vec{b}_i$ for all $i \in \mathbb{N}$. Let now S be sharply nesting and any element $\vec{v} \in \mathbb{V}$ be represented with respect to a given S-compatible basis $(\vec{b}_i)_{i\in\mathbb{N}}$, i.e., by a sequence of constants $(v_0, v_1, v_2, \ldots)$ such that $\vec{v} = \sum_{i\in\mathbb{N}} v_i \vec{b}_i$ and almost all v_i are zero. Then the action of S on $\vec{v}$ can be intuitively described by the following *shifting* property on sequences

$$S(\vec{v}) = S(v_0, v_1, v_2, \ldots) = (v_1, v_2, v_3, \ldots)$$

With respect to this property, it is clear that any $\vec{w}$ such that $S(\vec{w}) = \vec{v}$ will correspond to a sequence $(w_0, v_0, v_1, v_2, \cdots)$ for some w_0.

It is not true in general that for any nesting operator $S \in \mathrm{End}(\mathbb{V})$ there exists an S-compatible basis. In fact, from the existence of an S-compatible basis it easily follows that S is surjective. As we show in the next proposition, this already is a sufficient condition.

Proposition 3.6 *Let $\mathbb{V}$ be a vector space and S be a nesting operator on $\mathbb{V}$. Then an S-compatible basis exists if and only if S is surjective.*

Proof. One direction is obvious. Let now $S \in \mathrm{End}(\mathbb{V})$ be a surjective nesting operator. We define $\mathcal{B}$ as follows. Let $\mathcal{B}_0$ be an arbitrary basis of $\ker S$. We prove that for any basis $\mathcal{B}_n$ of $\ker S^{n+1}$ there exists a basis $\mathcal{B}_{n+1}$ of $\ker S^{n+2}$ such that $S\mathcal{B}_{n+1} = \mathcal{B}_n \cup \{\vec{0}\}$ and $\mathcal{B}_n \subsetneq \mathcal{B}_{n+1}$.

Define $\mathcal{B}_n^\circ := \{\vec{b}^\circ \ ; \ \vec{b} \in \mathcal{B}_n\}$, where $\vec{b}^\circ$ is an arbitrary chosen element of $S^{-1}(\vec{b})$ for all $\vec{b} \in \mathcal{B}_n$. Since S is surjective, $\mathcal{B}_n^\circ$ is well-defined. We prove that $\mathcal{B}_{n+1} := \mathcal{B}_n^\circ \cup \mathcal{B}_0$ satisfies the conditions above.

Consider any $\vec{v} \in \ker S^{n+2}$. Since $S(\vec{v}) \in \ker S^{n+1}$ we have

$$S(\vec{v}) = \sum_{\vec{b}\in\mathcal{B}_n} \lambda_{\vec{b}}\, \vec{b}$$

for some constants $\lambda_{\vec{b}}$. From $S(\vec{b}^\circ) = \vec{b}$ it follows

$$S(\vec{v}) = \sum_{\vec{b}\in\mathcal{B}_n} \lambda_{\vec{b}}\, S(\vec{b}^\circ) = S\left(\sum_{\vec{b}\in\mathcal{B}_n} \lambda_{\vec{b}}\, \vec{b}^\circ\right)$$

and therefore

$$\vec{v} = \sum_{\vec{b}\in\mathcal{B}_n} \lambda_{\vec{b}}\, \vec{b}^\circ + \vec{v}_0$$

for some $\vec{v}_0 \in \ker S$. This proves that $\mathcal{B}_n^\circ \cup \mathcal{B}_0$ generates $\ker S^{n+2}$. Furthermore the elements in $\mathcal{B}_n^\circ$ are linearly independent, since from $\sum_{\vec{b}^\circ \in \mathcal{B}_n^\circ} \lambda_{\vec{b}^\circ} \vec{b}^\circ = 0$ it follows $\sum_{\vec{b} \in \mathcal{B}_n} \lambda_{\vec{b}^\circ} \vec{b} = 0$ and so $\lambda_{\vec{b}^\circ} = 0$ for all $\vec{b}^\circ \in \mathcal{B}_n^\circ$. The linear independence of $\mathcal{B}_{n+1} = \mathcal{B}_n^\circ \cup \mathcal{B}_0$ completes the proof. ∎

The following result is a direct consequence of the proof of the last proposition.

Corollary 3.1 *Let S be an operator on $\mathbb{V}$. If S is surjective, then*

$$\ker S^{n+1} / \ker S^n \cong \ker S$$

holds for all $n \in \mathbb{N}^+$ and, in particular, $\dim \ker S^{n+1} = \dim \ker S^n + \dim \ker S$.

If S is a surjective nesting operator, we may speak of $\dim(\ker S)$ as the **nesting step** of S. As a trivial consequence, if S is a surjective nesting operator with $\dim(\ker S) = 1$, then S is sharply nesting . On the other hand, if S is decompasable into sharply nesting operators then S is surjective; more precisely:

Lemma 3.3 *Let $\mathbb{V}$ be a vector space, S a nesting operator on $\mathbb{V}$ and $\mathcal{I}$ a set with cardinality* $\mathrm{Card}(\mathcal{I}) = \dim(\ker S)$. *Then S is surjective if and only if S is decomposable by a sequence of sharply nesting operators $(S_i)_{i \in \mathcal{I}}$.*

Proof. Let S be a surjective nesting operator on $\mathbb{V}$. We have to prove that there exist $(\mathbb{W}_i)_{i\in\mathcal{I}}$ with $\mathbb{W}_i \subsetneq \mathbb{V}$ and $\mathbb{V} = \bigoplus_{i\in\mathcal{I}} \mathbb{W}_i$ such that

$$S = \bigoplus_{i\in\mathcal{I}} S_i$$

for S_i sharply nesting operator on $\mathbb{W}_i$.

Since S is surjective, an S-compatible basis $\mathcal{B}$ exists. Consider $\mathcal{B}_0 = \{\vec{b}_i \ ; \ i \in \mathcal{I}\}$ and define the sequences $(\mathcal{A}_i)_{i\in\mathcal{I}}$ and $(\mathbb{W}_i)_{i\in\mathcal{I}}$ by

$$\mathcal{A}_i := \{\vec{b} \in \bigcup_{j\in\mathbb{N}} \mathcal{B}_j \ ; \ S^n(\vec{b}) = \vec{b}_i \text{ for some } n \in \mathbb{N}\}$$

and $\mathbb{W}_i = \langle \mathcal{A}_i \rangle$ for all $i \in \mathcal{I}$. Then it is easy to see that $\mathbb{V} = \bigoplus_{i\in\mathcal{I}} \mathbb{W}_i$. Defining $S_i \in \mathrm{End}(\mathbb{W}_i)$ to be the restriction of S on $\mathbb{W}_i$, we have that S_i is sharply nesting on $\mathbb{W}_i$ and furthermore $S = \bigoplus_{i\in\mathcal{I}} S_i$.

On the other side, assume that such sequences $(\mathbb{W}_i)_{i\in\mathcal{I}}$ and $(S_i)_{i\in\mathcal{I}}$ exist. S_j is sharply nesting on $\mathbb{W}_j$, so let $\mathcal{B}^{(j)} = (\vec{b}_i^{(j)})_{i\in\mathbb{N}}$ be S_j-compatible basis of $\mathbb{W}_j$ for all $j \in \mathcal{I}$. Then the basis $\mathcal{B} = (\mathcal{B}_i)_{i\in\mathbb{N}}$ defined by

$$\mathcal{B}_i = \{\vec{b}_i^{(j)} \ ; \ j \in \mathcal{I}\}$$

is an S-compatible basis for $\mathbb{V}$. From this it follows that S is surjective on $\mathbb{V}$. ∎

A condition for a surjective nesting operator S to be sharply nesting is given in Proposition 3.12.

In the case where S has finite nesting step, an S-compatible basis can be inductively constructed from any S-nesting basis. In the following proposition we describe an algorithm for this construction, mainly based on the solution of linear systems of equations over $\mathcal{K}$.

Proposition 3.7 *Let $\mathbb{V}$ be a vector space and S a surjective nesting operator on $\mathbb{V}$ of finite nesting step. Then an S-compatible basis can be computed from any S-nesting basis.*

Proof. Let S be nesting operator on $\mathbb{V}$ of finite nesting step d and $\mathcal{B} = (\mathcal{B}_i)_{i\in\mathbb{N}}$ an S-nesting basis for $\mathbb{V}$. From Lemma 3.1 it follows that $\dim(\ker S^{i+1}) = \mathrm{Card}(\mathcal{B}_i) = (i+1)d$, or, equivalently, that $\mathrm{Card}(\mathcal{B}_{i+1} \setminus \mathcal{B}_i) = d$ for all $i \in \mathbb{N}$. Let now $\mathcal{B}^C_{i+1}$ denote the set $\mathcal{B}_{i+1} \setminus \mathcal{B}_i$ for all $i \in \mathbb{N}$ with $\mathcal{B}^C_0 = \mathcal{B}_0$, and let $\mathcal{B}^C_i = \{\vec{b}^{(1)}_i, \ldots, \vec{b}^{(d)}_i\}$.

We inductively construct an S-compatible basis $\mathcal{A} = (\mathcal{A}_i)_{i\in\mathbb{N}}$ from $\mathcal{B}$. First we show how to compute $\mathcal{A}^C_i$ such that $S\mathcal{A}^C_{i+1} \subseteq \mathcal{A}^C_i$, then how to choose the elements $\vec{a}^{(j)}_{i+1}$ such that $S(\vec{a}^{(j)}_{i+1}) = \vec{a}^{(j)}_i$. The computation mainly consists of solving systems of linear equations of dimension d over the ground field.

We proceed by induction on i. For $i = 0$ we have $\mathcal{A}_0 := \mathcal{B}_0$ and $\vec{a}^{(j)}_0 := \vec{b}^{(j)}_0$. Assume now that the components $\mathcal{A}_0, \ldots, \mathcal{A}_i$ of the S-compatible basis $\mathcal{A}$ have already been computed and consider the set $\mathcal{B}^C_{i+1} = \{\vec{b}^{(1)}_{i+1}, \ldots, \vec{b}^{(d)}_{i+1}\}$.

Since $S\mathcal{B}_{i+1} \subseteq \mathcal{B}_i$ holds, we have also $S\mathcal{B}_{i+1} \subseteq \langle\mathcal{A}_i\rangle$ for all $j = 1, \ldots, d$, so

$$S\left(\vec{b}^{(j)}_{i+1}\right) = \sum_{\substack{0\le n\le i\\ 1\le m\le d}} \lambda^{(j)}_{m,n}\, \vec{a}^{(m)}_n$$

holds for some constants $\lambda^{(j)}_{m,n}$. By induction we know that $S\mathcal{A}^C_{n+1} = \mathcal{A}^C_n$ for all $n < i$. This implies

$$\begin{aligned} S\left(\vec{b}^{(j)}_{i+1}\right) &= \sum_{1\le m\le d} \lambda^{(j)}_{m,i}\vec{a}^{(m)}_i + \sum_{\substack{0\le n\le i-1\\ 1\le m\le d}} \lambda^{(j)}_{m,n}\vec{a}^{(m)}_n \\ &= \sum_{1\le m\le d} \lambda^{(j)}_{m,i}\vec{a}^{(m)}_i + \sum_{\substack{0\le n\le i-1\\ 1\le m\le d}} \lambda^{(j)}_{m,n} S\vec{a}^{(m)}_{n+1} \end{aligned}$$

We now define

$$\vec{b}'_j := \vec{b}^{(j)}_{i+1} - \sum_{\substack{0\le n\le i-1\\ 1\le m\le d}} \lambda^{(j)}_{m,n}\,\vec{a}^{(m)}_{n+1}$$

Observe that $\vec{b}'_j$ is obtained by a linear combination of an element of $\mathcal{B}^C_{i+1}$ and elements from $\langle\mathcal{A}_i\rangle = \ker S^{i+1}$. This means that all $\vec{b}'_1, \ldots, \vec{b}'_d$ are linearly independent and $\mathcal{A}_i \cup \{\vec{b}'_1, \ldots, \vec{b}'_d\}$ is a basis for $\ker S^{i+1}$.

In addition $S(\vec{b}'_j) = \sum_{1\leq m\leq d} \lambda_m^{(j)} \vec{a}_i^{(m)}$, where we write for simplicity $\lambda_m^{(j)}$ instead of $\lambda_{m,i}^{(j)}$. We conclude the proof showing that constants α_k^j can be computed such that

$$\text{for } \vec{a}_{i+1}^{(j)} := \sum_{k=1}^{d} \alpha_k^j \vec{b}'_k \quad \text{we have} \quad S\left(\vec{a}_{i+1}^{(j)}\right) = \vec{a}_i^{(j)}$$

Define $\vec{a}_{i+1}^{(j)} := \sum_{k=1}^{d} \alpha_k^j \vec{b}'_k$ with indeterminate α_k^j. From the considerations above we can write

$$S\left(\vec{a}_{i+1}^{(j)}\right) = \sum_{k=1}^{d} \alpha_k^j \sum_{m=1}^{d} \lambda_m^{(k)} \vec{a}_i^{(m)} = \sum_{m=1}^{d} \left(\sum_{k=1}^{d} \alpha_k^j \lambda_m^{(k)}\right) \vec{a}_i^{(m)}$$

To determine constants α_k^j such that $\sum_k \alpha_k^j \lambda_m^{(k)} = \delta_{j,m}$ corresponds to solve a system of d linear equations in d indeterminates for each j. The solvability of the systems above is given by the linear independence of $\vec{b}'_1, \ldots, \vec{b}'_d$, which implies the linear independence of the coefficient rows $\lambda_m^{(1)}, \lambda_m^{(d)}$ for $m = 1, \ldots, d$.

An analogous reasoning shows that the elements $\vec{a}_{i+1}^{(1)}, \ldots, \vec{a}_{i+1}^{(d)}$ computed in this way are linearly independent and that $\mathcal{A}_{i+1} := \mathcal{A}_i \cup \{\vec{a}_{i+1}^{(1)}, \ldots, \vec{a}_{i+1}^{(d)}\}$ satisfies the conditions stated in the lemma. ∎

Since the most interesting case for umbral calculus is the sharply nesting one, we state the last proposition for S sharply nesting and give an equivalent method for computing an S-compatible basis.

Proposition 3.8 *Let $\mathbb{V}$ be a vector space and S a sharply nesting operator on $\mathbb{V}$. Then an S-compatible basis can be obtained from any S-nesting basis.*

Proof. Let S be sharply nesting operator on $\mathbb{V}$ and $\mathcal{V} = (\mathbb{V}, \left(\ker S^{i+1}\right)_{i\in\mathbb{N}})$. Consider an arbitrary S-nesting basis $\mathcal{B} = (\vec{b}_i)_{i\in\mathbb{N}}$ of $\mathcal{V}$. We construct an S-compatible basis $\mathcal{B}' = (\vec{b}'_i)_{i\in\mathbb{N}}$ from $\mathcal{B}$.

Since $S(\ker S) = \{\vec{0}\}$ we have $S(\vec{b}_0) = \vec{0}$, so we put $\vec{b}'_0 := \vec{b}_0$. Assume now that we already found the first part $\vec{b}'_0, \ldots, \vec{b}'_j$ of the S-compatible basis $\mathcal{B}'$. We obtain $\vec{b}'_{j+1}$ from that sequence as follows: Consider $S(\vec{b}_{j+1})$. It is straightforward to see that $S\left(\ker S^{j+2}\right) \subseteq \ker S^{j+1}$. Since $\vec{b}_{j+1} \in \ker S^{j+2}$ we have for some constants $\lambda_0, \ldots, \lambda_j$

$$S\left(\vec{b}_{j+1}\right) = \lambda_0 \vec{b}'_0 + \cdots + \lambda_j \vec{b}'_j \tag{3.6}$$

If $\lambda_j = 0$ holds, then $S(\vec{b}_{j+1}) \in \ker S^j$ and $\vec{b}_{j+1} \in \ker S^{j+1}$. This implies $\left\langle \vec{b}_0, \ldots, \vec{b}_{j+1} \right\rangle = \ker S^{j+1}$, which is a contradiction to the nesting property of $(\vec{b}_i)_{i\in\mathbb{N}}$. So $\lambda_j \neq 0$ necessarily holds.

Now it is sufficient to define

$$\vec{b}'_{j+1} := \lambda_j^{-1}\left(\vec{b}_{j+1} - \sum_{k=0}^{j-1}\lambda_k \vec{b}'_{k+1}\right)$$

and one easily verifies that from this definition and (3.6) it follows that

$$\begin{aligned} S\left(\vec{b}'_{j+1}\right) &= \lambda_j^{-1}\left(S\left(\vec{b}_{j+1}\right) - \sum_{k=0}^{j-1}\lambda_k S\left(\vec{b}'_{k+1}\right)\right) = \lambda_j^{-1}\left(\sum_{k=0}^{j}\lambda_k \vec{b}'_k - \sum_{k=0}^{j-1}\lambda_k \vec{b}'_k\right) \\ &= \lambda_j^{-1}\lambda_j \vec{b}'_j = \vec{b}'_j \end{aligned}$$

Since $\vec{b}'_{j+1}$ is a linear combination of $\vec{b}_{j+1}$ and $\vec{b}'_0, \ldots, \vec{b}'_j$ where $\vec{b}_{j+1}$ arises with nonzero coefficient, the sequence $\vec{b}'_0, \ldots, \vec{b}'_{j+1}$ is a basis for $\ker S^{j+2}$. This proves that the nesting basis $(\vec{b}'_i)_{i\in\mathbb{N}}$ defined this way is S-compatible. ∎

Proposition 3.9 *Let $\mathbb{V}$ be an infinite-dimensional vector space. Then a surjective $S \in \mathrm{End}(\mathbb{V})$ is a nesting operator on $\mathbb{V}$ iff there exists a nesting sequence $(\mathbb{V}_i)_{i\in\mathbb{N}}$ for $\mathbb{V}$ such that $S\mathbb{V}_0 = \{\vec{0}\}$ and $S\mathbb{V}_{i+1} = \mathbb{V}_i$ for all $i \in \mathbb{N}$. In addition, if $\dim \mathbb{V}_{i+1}/\mathbb{V}_i = 1$ for all $i \in \mathbb{N}$ then S is sharply nesting operator.*

Proof. If S is a surjective nesting operator, then the sequence $(\ker S^{i+1})_{i\in\mathbb{N}}$ satisfies the condition in the proposition. It is sufficient to consider an S-compatible basis $\mathcal{B}$ for the proof.

Let now $(\mathbb{V}_i)_{i\in\mathbb{N}}$ be a nesting sequence for $\mathbb{V}$ such that $S\mathbb{V}_0 = \{\vec{0}\}$ and $S\mathbb{V}_{i+1} = \mathbb{V}_i$ for all $i \in \mathbb{N}$. From this it follows directly that $\mathbb{V}_i \subseteq \ker S^{i+1}$, so if $\mathbb{V} = \sum_i \mathbb{V}_i$, then also $\mathbb{V} = \sum_i \ker S^{i+1}$.

For proving that $(\ker S^{i+1})_{i\in\mathbb{N}}$ is a nesting sequence for $\mathbb{V}$ it is now sufficient to prove that $\ker S^i \neq \ker S^{i+1}$ for all $i \in \mathbb{N}$. We show by induction that $(\ker S^{n+1} \setminus \ker S^n) \cap \mathbb{V}_n \neq \emptyset$ for all $n \in \mathbb{N}$. For $n = 0$ we have $\{\vec{0}\} \neq \mathbb{V}_0 \subseteq \ker S$ and $\ker S \setminus \ker S^0 = \ker S \setminus \{\vec{0}\}$, so the condition holds. Assume now that it holds for $n > 0$ and consider any $y \in (\ker S^{n+1} \setminus \ker S^n) \cap \mathbb{V}_n$. Since $S\mathbb{V}_{n+1} = \mathbb{V}_n$ and S is surjective, there exists a $\hat{y}$ in $S^{-1}(y) \cap \ker S^{n+2} \cap \mathbb{V}_{n+1}$. It is easy to see that $\hat{y} \notin \ker S^{n+1}$, since otherwise $y = S\hat{y} \in \ker S^n$, a contradiction to the choice of y. So $\hat{y} \in (\ker S^{n+2} \setminus \ker S^{n+1}) \cap \mathbb{V}_{n+1}$. This completes the proof. In general we have $\mathbb{V}_i \subseteq \ker S^{i+1}$ and $\ker S^{i+1} \subset \mathbb{V}_{i+2}$. If $\dim \mathbb{V}_{i+1}/\mathbb{V}_i = 1$, then $\mathbb{V}_i = \ker S^{i+1}$ always holds and S is sharply nesting . ∎

Example 3.3 *For the univariate polynomial space $\mathbb{V} = \mathcal{K}[x]$, which is nested as in Example 3.2, a corresponding nesting operator is $S = d/dx$, i.e., the usual differential operator on polynomials. In addition, S is also sharply nesting. The basis $(x^i)_{i\in\mathbb{N}}$ is S-nesting, but not S-compatible, since $S(x^n) = nx^{n-1} \neq x^{n-1}$. We apply Theorem 3.8 and compute an S compatible basis $(\vec{b}'_i)_{i\in\mathbb{N}}$. We have $\vec{b}'_0 = \vec{b}_0 = x^0 = 1$ and*

$S(\vec{b}_1) = S(x) = 1 = \vec{b}'_0$, so $\vec{b}'_1 = x$. From $S(x^2) = 2x = 2\vec{b}'_1$ we have $\vec{b}'_2 = 2^{-1}\vec{b}_2 = \frac{x^2}{2}$. This way $S(x^3) = 3x^2 = 6\vec{b}'_2$ and $\vec{b}'_3 = 6^{-1}\vec{b}_3 = \frac{x^3}{6}$. Iteration yields $\vec{b}'_i = \frac{x^i}{i!}$, the sequence of divided powers.

3.7 The Algebra of S-invariant Operators

If S is sharply nesting, then the S-invariant operators, i.e., all $T \in \mathrm{End}(\mathbb{V})$ such that $TS = ST$, build an algebra. In addition, this algebra is isomorphic to the univariate formal power series algebra with respect to addition and convolution. The property of a nesting operator S to provide a nesting sequence for $\mathbb{V}$ has the implication that all sums over powers of S are summable, as the following proposition states.

Proposition 3.10 *Let $S \in \mathrm{End}(\mathbb{V})$ be a nesting operator on $\mathbb{V}$. Then for any sequence $(a_i)_{i\in\mathbb{N}}$ in the scalar field $\mathcal{K}$ of $\mathbb{V}$, the sequence $(a_iS^i)_{i\in\mathbb{N}}$ is summable. In other words, for any formal power series $\alpha(t) = \sum_{i\in\mathbb{N}} a_it^i$ in $\mathcal{K}[[t]]$, the series*

$$\alpha(S) := \sum_{i\in\mathbb{N}} a_iS^i$$

converges in the finite topology on $\mathrm{End}(\mathbb{V})$.

Proof. By Lemma 3.1 it is sufficient to prove that for any $\vec{v} \in \mathbb{V}$ we have $a_iS^i(\vec{v}) = \vec{0}$ for almost all $i \in \mathbb{N}$. Since S is nesting, we have that $\sum_i \ker S^i = \mathbb{V}$, so the proposition holds. ∎

By abuse of notation let $\mathcal{K}[[S]]$ denote the set of all operators which can be written in the form $\sum_{i\in\mathbb{N}} a_iS^i$. Note that, obviously, not every operator in $\mathrm{End}(\mathbb{V})$ can be represented as such a sum and, if S is not nesting, not all formal series $\sum_{i\in\mathbb{N}} a_iS^i$ necessarily converge. In the case of a sharply nesting operator S, the operators of this form are precisely the S-invariant operators, i.e., the operators which commute with S with respect to composition of operators, and furthermore $\mathcal{K}[[S]] \cong \mathcal{K}[[t]]$.

Theorem 3.1 *Let S be a sharply nesting operator on $\mathbb{V}$. Then $\mathcal{K}[[S]]$ is the maximal commutative sub-ring of* $\mathrm{End}(\mathbb{V})$ *containing S.*

Proof. Assume that S is sharply nesting operator. We have to show that for all $T \in \mathrm{End}(\mathbb{V})$, $TS = ST$ holds if and only if $T = \sum_{i\in\mathbb{N}} a_iS^i$ for some $\sum_{i\in\mathbb{N}} a_it^i \in \mathcal{K}[[t]]$. It is evident that every operator of the form $\sum_{i\in\mathbb{N}} a_iS^i$ commutes with S. For the other direction, let $\mathcal{B} = (\vec{b}_i)_{i\in\mathbb{N}}$ be an S-compatible basis of $\mathbb{V}$ and $T \in \mathrm{End}(\mathbb{V})$ such that $TS = ST$. From the fact, that for all $n \in \mathbb{N}$

$$ST\left(\vec{b}_0\right) = TS\left(\vec{b}_0\right) = T\left(\vec{0}\right) = \vec{0} \qquad \text{and} \qquad ST\left(\vec{b}_{n+1}\right) = TS\left(\vec{b}_{n+1}\right) = T\left(\vec{b}_n\right)$$

it follows that $T(\vec{b}_n) \in \ker S^{n+1}$, i.e.,

$$T\left(\vec{b}_n\right) = \sum_{i=0}^{n} \lambda_{n-i}^{(n)} \vec{b}_i$$

for some sequence of constants $\lambda_j^{(n)}$ and all $n \in \mathbb{N}$. Since $ST(\vec{b}_{n+1}) = T(\vec{b}_n)$ holds, we have $\lambda_{n-i}^{(n)} = \lambda_{n-i}^{(n+1)}$. This means that there is a sequence $(\lambda_i)_{i\in\mathbb{N}}$ such that $T(\vec{b}_n) = \sum_{i=0}^{n} \lambda_{n-i}\vec{b}_i$.

To describe the situation more explicitely, we can represent this fact by the following diagram

$$\begin{array}{ccc}
(\overbrace{0,0,\ldots,0}^{n+1},1,0,\ldots) & \overrightarrow{T} & (\lambda_{n+1},\lambda_n,\ldots,\lambda_0,0,\ldots) \\
S\Big\downarrow & & S\Big\downarrow \\
(\underbrace{0,\ldots,0}_{n},1,0,\ldots) & \xrightarrow{T} & (\lambda_n,\ldots,\lambda_0,0,\ldots)
\end{array}$$

where the form of the upper right element $T(\vec{b}_{n+1})$ follows from the commutativity of the diagram and the shifting behaviour of S with to the chosen S-compatible basis chosen.

From this considerations it directly follows that $T = \sum_{i\in\mathbb{N}} a_i S^i$ for the sequence $(a_i)_{i\in\mathbb{N}}$ defined by $a_i = \lambda_i$ or, more precisely, by

$$a_i = \beta^0\left(T\left(\vec{b}_i\right)\right)$$

where $\beta^0(\vec{b}_j) = \delta_{0j}$ gives the coefficient of $\vec{b}_0$ in the representation of the argument with respect to $\mathcal{B}$. ∎

From the last few lines of the proof of Theorem 3.1 we can directly derive the following proposition.

Proposition 3.11 *Let S be a sharply nesting operator on $\mathbb{V}$ and T an S-invariant operator. Then T can be written as $T = \sum_{i\in\mathbb{N}} a_i S^i$ where*

$$a_i = \beta^0\left(T\left(\vec{b}_i\right)\right)$$

for any S-compatible basis $(\vec{b}_i)_{i\in\mathbb{N}}$.

For a surjective nesting operator S, the converse of Theorem 3.1 holds in the following sense.

Proposition 3.12 *Let S be a surjective nesting operator on $\mathbb{V}$. Then, if $\mathcal{K}[[S]]$ is the maximal commutative subring of* $\mathrm{End}(\mathbb{V})$ *containing S, then S is sharply nesting .*

Proof. Let S be a surjective nesting operator on $\mathbb{V}$ and assume that $\mathcal{K}[[S]]$ is the maximal commutative subring of $\mathrm{End}(\mathbb{V})$ containing S. From Proposition 3.6 there exists an S-compatible basis $\mathcal{B} = (\mathcal{B}_i)_{i\in\mathbb{N}}$ of $\mathbb{V}$. Let $\mathcal{I}$ be a suitable index set for the elements of $\mathcal{B}_0$. Then we can write $\mathcal{B}_{i+1} \setminus \mathcal{B}_i = \{\vec{b}^{(j)}_{i+1} \mid j \in \mathcal{I}\}$ with $S(\vec{b}^{(j)}_{i+1}) = \vec{b}^{(j)}_i$ for all $i \in \mathbb{N}$ and $j \in \mathcal{I}$.

We prove that $\mathrm{Card}(\mathcal{I}) = 1$ holds, i.e., S is sharply nesting . For this purpose let j_0 be an arbitrary but fixed element of $\mathcal{I}$. Then we define an operator $T \in \mathrm{End}(\mathbb{V})$ such that T behaves like S on the j_0-th *chain* in the basis $\mathcal{B}$ and maps all other elements to $\vec{0}$, viz.

$$T\left(\vec{b}^{(j)}_i\right) := \begin{cases} \vec{b}^{(j)}_{i-1} & \text{if } j = j_0 \text{ and } i > 0 \\ \vec{0} & \text{otherwise} \end{cases}$$

Observe that T is S-invariant, since we have

$$TS\left(\vec{b}^{(j)}_i\right) = \begin{cases} T\left(\vec{b}^{(j)}_{i-1}\right) & \text{if } i > 0 \\ T\left(\vec{0}\right) & \text{otherwise} \end{cases} = \begin{cases} \vec{b}^{(j)}_{i-2} & \text{if } i > 1 \text{ and } j = j_0 \\ \vec{0} & \text{otherwise} \end{cases}$$

$$ST\left(\vec{b}^{(j)}_i\right) = \begin{cases} S\left(\vec{b}^{(j)}_{i-1}\right) & \text{if } i > 0 \text{ and } j = j_0 \\ S\left(\vec{0}\right) & \text{otherwise} \end{cases} = \begin{cases} \vec{b}^{(j)}_{i-2} & \text{if } i > 1 \text{ and } j = j_0 \\ \vec{0} & \text{otherwise} \end{cases}$$

for all $i \in \mathbb{N}$ and all $j \in \mathcal{I}$. From the assumption above it follows that T is expressible as a series $\sum_{i\in\mathbb{N}} a_i S^i$ for some constants $(a_i)_{i\in\mathbb{N}}$. This implies that $\mathcal{I} = \{j_0\}$. In fact, assume that $j_1 \in \mathcal{I}$ exists with $j_1 \neq j_0$. Then from

$$T\left(\vec{b}^{(j_1)}_k\right) = \sum_{i\in\mathbb{N}} a_i S^i \left(\vec{b}^{(j_1)}_k\right) = \sum_{i=0}^{k} a_i \vec{b}^{(j_1)}_{k-i} = 0$$

for all $k \in \mathbb{N}$ it would follow that $a_i = 0$ for all $i \in \mathbb{N}$, a contradiction to the definition of T. This completes the proof of the proposition. ∎

We explicitly observe that the following proposition holds.

Proposition 3.13 *Let $S \in$* $\mathrm{End}(\mathbb{V})$ *be sharply nesting . Then the ring $\mathcal{K}[[S]]$, as a sub-ring of* $\mathrm{End}(\mathbb{V})$ *with composition, is isomorphic to the ring $\mathcal{K}[[t]]$ of formal power series with the usual convolution.*

Proof. Let $(\vec{b}_i)_{i\in\mathbb{N}}$ be an S-compatible basis and let U, V be S-invariant operators. Then from Theorem 3.1 it follows that $U = \alpha(S)$ and $V = \beta(S)$ for some $\alpha, \beta \in \mathcal{K}[[t]]$. Then one shows by straightforward verification that $UV(\vec{b}_n) = (\alpha(S)\cdot\beta(S))(\vec{b}_n)$ for all $n \in \mathbb{N}$. ∎

The interpretation of the inverse of a series with respect to composition is explained in the following corollary. For a formal power series $\alpha(t) \in \mathcal{K}[[t]]$ we write $\alpha^{inv}(t)$ for the compositional inverse of α, if it exists, i.e., $\alpha(\alpha^{inv}(t)) = \alpha^{inv}(\alpha(t)) = t$.

Corollary 3.2 *Let S and T be sharply nesting operators on $\mathbb{V}$. Then $ST = TS$ holds if and only if there exists an invertible $\alpha \in \mathcal{K}[[t]]$ such that*

$$S = \alpha(T) \qquad \text{and} \qquad T = \alpha^{inv}(S)$$

Proof. From Theorem 3.1 it directly follows that $S \in \mathcal{K}[[T]]$ and $T \in \mathcal{K}[[S]]$, so $S = \alpha(T)$ and $T = \beta(S)$ for some $\alpha, \beta \in \mathcal{K}[[t]]$. In addition it holds that $S = \alpha(\beta(S))$ and $T = \beta(\alpha(T))$, i.e., $\beta = \alpha^{inv}$. ∎

In the following proposition the S-invariant operators represented by a series, invertible under convolution, are characterized. They are precisely those operators, which map S-compatible bases onto S-compatible bases.

Proposition 3.14 *Let $S \in \mathrm{End}(\mathbb{V})$ be sharply nesting and let $\mathcal{B} = (\vec{b}_i)_{i\in\mathbb{N}}$ be an S-compatible basis. Then a basis $(\vec{s}_i)_{i\in\mathbb{N}}$ is S-compatible if and only if $(\vec{s}_i)_{i\in\mathbb{N}} = (T^{-1}\vec{b}_i)_{i\in\mathbb{N}}$ for some invertible and S-invariant operator T.*

Proof. Let S be sharply nesting and $\mathcal{B} = (\vec{b}_i)_{i\in\mathbb{N}}$ an S-compatible basis. Assume that for some invertible S-invariant operator T we have $\vec{s}_i = T^{-1}\vec{b}_i$ for all $i \in \mathbb{N}$. Then

$$S(\vec{s}_{i+1}) = ST^{-1}(\vec{b}_{i+1}) = T^{-1}S(\vec{b}_{i+1}) = T^{-1}(\vec{b}_i) = \vec{s}_i$$

and $S(\vec{s}_0) = T^{-1}S(\vec{b}_0) = T^{-1}(\vec{0}) = \vec{0}$. So, $(\vec{s}_i)_{i\in\mathbb{N}}$ is S-compatible.

For the other direction, assume that $(\vec{s}_i)_{i\in\mathbb{N}}$ is an S-compatible basis. We have to prove that there exists a family of constants $(a_i)_{i\in\mathbb{N}}$ such that $\vec{s}_i = a_0\vec{b}_i + \cdots + a_i\vec{b}_0$ for all $i \in \mathbb{N}$, with $a_0 \neq 0$. In this case we would have a suitable T defined by $T^{-1} = \sum_i a_i S^i$.

Let now $\vec{s}_i$ be associated to the sequence of coefficients of its representation with respect to $(\vec{b}_i)_{i\in\mathbb{N}}$. Then from the fact that $\vec{s}_{i+1} \in S^{-1}(\vec{s}_i)$ follows that

$$\vec{s}_i = (c_0, c_1, c_2, \ldots) \Longrightarrow \vec{s}_{i+1} = (d, c_0, c_1, c_2, \ldots)$$

where $d = \beta^0(\vec{s}_{i+1})$. With the fact that $\vec{s}_0 = (a_0, 0, 0, \ldots)$ for some $a_0 \neq 0$, because $\vec{s}_0 \in \ker S$ and $\vec{s}_0 \neq \vec{0}$ we have by induction that $\vec{s}_i = T^{-1}(\vec{b}_i)$ for the invertible operator defined by

$$T^{-1} = \sum_{i\in\mathbb{N}} \beta^0(\vec{s}_i)S^i$$

∎

On the other hand, the isomorphism $\mathcal{K}[[S]] \cong \mathcal{K}[[t]]$ holds only for nesting operators, as we state in the following proposition.

Proposition 3.15 *Let $S \in \mathrm{End}(\mathbb{V})$. If $\mathcal{K}[[S]] \subseteq \mathrm{End}(\mathbb{V})$ and $\mathcal{K}[[S]] \cong \mathcal{K}[[t]]$ then S is nesting.*

Proof. Let S be an operator on $\mathbb{V}$ and assume that $\mathcal{K}[[S]] \subseteq \mathrm{End}(\mathbb{V})$ and $\mathcal{K}[[S]] \cong \mathcal{K}[[t]]$. From the fact that all series $\sum_i a_i S^i$ converge in the finite topology on $\mathrm{End}(\mathbb{V})$ it follows that for all $\vec{v} \in \mathbb{V}$ we have $S^i\vec{v} = \vec{0}$ for almost all $i \in \mathbb{N}$, so $\mathbb{V} = \sum_i \ker S^i$ and, in particular, $\dim(\ker S) > 0$. In addition, from $\mathcal{K}[[S]] \cong \mathcal{K}[[t]]$ it follows that $\ker S^i \neq \ker S^{i+1}$ for all i, since otherwise for some $i_0 \in \mathbb{N}$ we would have $\sum_i a_i S^i = \sum_i a_i S^i + aS^{i_0}$ for all constants a. This shows that S is nesting. ∎

3.8 The Algebra of Hemimorphisms

As we said, sharply nesting operators on $\mathbb{V}$ mainly play the role of the delta operators in the classical umbral calculus, when we impose the following behaviour with respect to the convolution. Compared to the polynomial case, for any coalgebra the analogues of shift invariant operators are the hemimorphisms defined below, while the sharply nesting hemimorphisms correspond to the delta operators.

Let us first recall that for $\mathbb{V}$ a vector space, $\mathcal{C} = (\mathbb{V}, \Delta, \varepsilon)$ is a coalgebra if

$$\Delta : \mathbb{V} \longrightarrow \mathbb{V} \otimes \mathbb{V}, \quad \varepsilon : \mathbb{V} \longrightarrow \mathcal{K}$$

and the following diagrams commute

$$\begin{tikzcd}
\mathbb{V} \arrow[r, "\Delta"'] \arrow[d, "\Delta"] & \mathbb{V} \otimes \mathbb{V} \arrow[d, "id\otimes\Delta"] \\
\mathbb{V} \otimes \mathbb{V} \arrow[r, "\Delta\otimes id"'] & \mathbb{V} \otimes \mathbb{V} \otimes \mathbb{V}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbb{V} \arrow[r, "\cong"'] \arrow[d, "\cong"] \arrow[dr, "\Delta"] & \mathbb{V} \otimes \mathcal{K} \\
\mathcal{K} \otimes \mathbb{V} & \mathbb{V} \otimes \mathbb{V} \arrow[l, "\varepsilon\otimes id"] \arrow[u, "id\otimes\varepsilon"']
\end{tikzcd}$$

$$\begin{tikzcd}
 & \mathbb{V} \otimes \mathbb{V} \arrow[dd, "\mathrm{twist}"] \\
\mathbb{V} \arrow[ur, "\Delta"] \arrow[dr, "\Delta"] & \\
 & \mathbb{V} \otimes \mathbb{V}
\end{tikzcd}$$

Note that in the whole work we always consider cocommutative and counitary coalgebras. It is easy to show that, in addition, the Δ is injective. Let namely $\Delta\vec{v} = \Delta\vec{b}$, then we have $(\varepsilon \otimes \mathrm{id})\Delta\vec{v} = (\varepsilon \otimes \mathrm{id})\Delta\vec{b}$, and this implies $\vec{v} = \vec{b}$ by the properties of the counit ε.

Definition 3.8 *Let $S \in \mathrm{End}(\mathbb{V})$ and $\mathcal{C} = (\mathbb{V}, \Delta, \varepsilon)$ a coalgebra. Then S is called an* **hemimorphism** *of $\mathcal{C}$ if*

$$\Delta\, S = (\mathrm{id} \otimes S)\, \Delta$$

or, equivalently, $\Delta\, S = (S \otimes \mathrm{id})\, \Delta$. We denote by $\mathrm{Hem}(\mathcal{C})$ the algebra of all hemimorphisms of $\mathcal{C}$ (as a sub-algebra of $\mathrm{End}(\mathbb{V})$).

Note that in general $\mathrm{id} \otimes S \neq S \otimes \mathrm{id}$, but, since $\mathcal{C}$ is cocommutative, we have $(\mathrm{id} \otimes S)\, \Delta = (S \otimes \mathrm{id})\, \Delta$ for any operator S.

Let S be a sharply nesting hemimorphism of $\mathcal{C}$ and $(\vec{b}_i)_{i \in \mathbb{N}}$ an S-compatible basis. In order to describe more intuitively the meaning of Definition 3.8, we associate to each element $\vec{v} \otimes \vec{w} \in \mathbb{V} \otimes \mathbb{V}$ the bi-infinite sequence $(\tau_{ij})_{i,j \in \mathbb{N}}$ of coefficients with respect to the basis $(\vec{b}_i \otimes \vec{b}_j)_{i,j \in \mathbb{N}}$, i.e., $\vec{v} \otimes \vec{w} = \sum_{ij} \tau_{ij} \vec{b}_i \otimes \vec{b}_j$. Then, similarly to S, a *shifting* property characterizes the action of $S \otimes \mathrm{id}$ and $\mathrm{id} \otimes S$ on $(\vec{b}_i \otimes \vec{b}_j)_{i,j \in \mathbb{N}}$. This corresponds to *shifting* the bi-infinite sequence one step to the top, or to the left, respectively, as in the following simplified diagram

$$\begin{pmatrix} \tau_{10} & \tau_{11} & \tau_{12} & \cdots \\ \tau_{20} & \tau_{21} & \cdots & \\ \tau_{30} & \cdots & & \\ \cdots & & & \end{pmatrix} \overset{S \otimes \mathrm{id}}{\longleftarrow} \begin{pmatrix} \tau_{00} & \tau_{01} & \tau_{02} & \cdots \\ \tau_{10} & \tau_{11} & \cdots & \\ \tau_{20} & \cdots & & \\ \cdots & & & \end{pmatrix} \overset{\mathrm{id} \otimes S}{\longrightarrow} \begin{pmatrix} \tau_{01} & \tau_{02} & \tau_{03} & \cdots \\ \tau_{11} & \tau_{12} & \cdots & \\ \tau_{21} & \cdots & & \\ \cdots & & & \end{pmatrix}$$

Some interesting basic facts on the structure of the algebra $\mathrm{Hem}(\mathcal{C})$ are described in the following statements.

Lemma 3.4 *For any coalgebra $\mathcal{C}$ the algebra $\mathrm{Hem}(\mathcal{C})$ is commutative.*

Proof. Let $\mathcal{C} = (\mathbb{V}, \Delta, \varepsilon)$ and $T, S \in \mathrm{Hem}(\mathcal{C})$. Then we have

$$\Delta TS = (T \otimes \mathrm{id})\Delta S = (T \otimes \mathrm{id})(\mathrm{id} \otimes S)\Delta = (T \otimes S)\Delta = (\mathrm{id} \otimes S)(T \otimes \mathrm{id})\Delta = \Delta ST$$

Since Δ is injective, this proves that $TS = ST$. ∎

Proposition 3.16 *For any coalgebra $\mathcal{C}$ the algebra $\mathrm{Hem}(\mathcal{C})$ is isomorphic to the dual algebra $\mathcal{C}^* = (\mathbb{V}^*, \mu, u)$ of $\mathcal{C}$. Moreover, for any $T \in \mathrm{Hem}(\mathcal{C})$ the adjoint operator T^* corresponds to the multiplication by a constant functional, namely*

$$T^* \beta = \mu(\beta \otimes T^* u(1))$$

holds for all $\beta \in \mathbb{V}^*$.

Proof. Consider any $T \in \mathrm{Hem}(\mathcal{C})$. Since $\Delta\, T = (T \otimes \mathrm{id})\Delta$ we have the dual property for the adjoint operator

$$T^* \mu = \mu(T^* \otimes \mathrm{id}) \qquad (= \mu(\mathrm{id} \otimes T^*))$$

In particular, from $\mu(\beta \otimes u(1)) = \beta$ for all $\beta \in \mathbb{V}^*$ we have

$$T^*\beta = T^*\mu(\beta \otimes u(1)) = \mu(\mathrm{id} \otimes T^*)(\beta \otimes u(1)) = \mu(\beta \otimes T^*u(1))$$

So T^* corresponds to the multiplication with $T^*u(1)$ in $\mathcal{C}^*$. It is then straightforward to verify that this is an isomorphism of algebras.
∎

Proposition 3.17 *Let $\mathcal{C}$ be a coalgebra and $S \in \mathrm{Hem}(\mathcal{C})$. If S is sharply nesting then we have the following algebra isomorphism*

$$\mathrm{Hem}(\mathcal{C}) \cong \mathcal{K}[[t]]$$

and S is a pseudo-generator of $\mathrm{Hem}(\mathcal{C})$, *i.e.,* $\mathrm{Hem}(\mathcal{C}) = \mathcal{K}[[S]]$. *In other words, the hemimorphisms of $\mathcal{C}$ are precisely the S-invariant operators.*

Proof. Let S be sharply nesting . Then we know from Proposition 3.1 that $\mathcal{K}[[S]]$ contains precisely the S-invariant operators. From Lemma 3.4 and $S \in \mathrm{Hem}(\mathcal{C})$ we have that $\mathrm{Hem}(\mathcal{C}) \subseteq \mathcal{K}[[S]]$. On the other hand, it is easy to prove that all operators in $\mathcal{K}[[S]]$ are hemimorphisms in $\mathcal{C}$. This implies that $\mathrm{Hem}(\mathcal{C}) = \mathcal{K}[[S]] \cong \mathcal{K}[[t]]$. ∎

If S is a sharply nesting hemimorphism, then all S-compatible bases behave with respect to Δ in the way described in the following proposition. Recall that such bases correspond to the *Sheffer sequences* of polynomials in the classical umbral calculus [Rom84]. The proof of the assertion mainly rely on the shift-interpretation of $\Delta S = (\mathrm{id} \otimes S)\Delta$.

Proposition 3.18 *Let S be a sharply nesting hemimorphism of $\mathcal{C} = (\mathbb{V}, \Delta, \varepsilon)$. Then for any S-compatible basis $\mathcal{B} = (\vec{b}_i)_{i\in\mathbb{N}}$ there exists a sequence $(c_i)_{i\in\mathbb{N}}$ of constants in $\mathcal{K}$ such that*

$$\Delta\,\vec{b}_n \quad = \quad \sum_i c_{n-i} \sum_j \vec{b}_j \otimes \vec{b}_{i-j} \tag{3.7}$$

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

$$c_0 = \varepsilon(\vec{b}_0)^{-1} \quad \text{and} \quad c_n = -c_0 \sum_{i=0}^{n-1} c_i \varepsilon(\vec{b}_{n-i}) \tag{3.8}$$

Proof. Let S be sharply nesting hemimorphism of $\mathcal{C}$ and $(\vec{b}_i)_{i\in\mathbb{N}}$ an S-compatible basis. We show the assertion by induction.

It is easy to see that (3.7) holds for $n = 0$: Consider that $\Delta\,S(\vec{b}_0) = \vec{0} \otimes \vec{0}$. Since S is a hemimorphism, this implies $(S \otimes \mathrm{id})\Delta\,\vec{b}_0 = \vec{0} \otimes \vec{0}$. If we write $\Delta\,\vec{b}_0 = \sum_{i,j} \tau_{ij}\vec{b}_i \otimes \vec{b}_j$, then we have $\tau_{ij} = 0$ for all $i, j \in \mathbb{N}$ and $\tau_{00} \neq 0$. So, $\Delta\,\vec{b}_0 = c_0\vec{b}_0 \otimes \vec{b}_0$.

Let now (3.7) be already verified for n in $\mathbb{N}$, i.e., we can associate to $\Delta\,\vec{b}_n$ the following bi-infinite tableau with respect to the basis $(\vec{b}_i \otimes \vec{b}_j)_{i,j\in\mathbb{N}}$

$$\begin{pmatrix} c_n & c_{n-1} & c_{n-2} & \cdots & c_0 & 0 & \cdots \\ c_{n-1} & c_{n-2} & \cdots & c_0 & 0 & \cdots & \\ c_{n-2} & \cdots & c_0 & 0 & \cdots & & \\ \cdots & \cdots & \cdots & \cdots & & & \\ c_0 & 0 & \cdots & & & & \\ 0 & \cdots & & & & & \\ \cdots & & & & & & \end{pmatrix}$$

Recalling the shifting property described above, the fact that

$$(S\otimes \mathrm{id})\Delta\,\vec{b}_{n+1} = (\mathrm{id}\otimes S)\Delta\,\vec{b}_{n+1} = \Delta\,S(\vec{b}_{n+1}) = \Delta\,\vec{b}_n$$

directly implies that the representation of $\Delta\,\vec{b}_{n+1}$ as bi-infinite tableau must be

$$\begin{pmatrix} c & c_n & c_{n-1} & \cdots & c_0 & 0 & \cdots \\ c_n & c_{n-1} & \cdots & c_0 & 0 & \cdots & \\ c_{n-1} & \cdots & c_0 & 0 & \cdots & & \\ \cdots & \cdots & \cdots & \cdots & & & \\ c_0 & 0 & \cdots & & & & \\ 0 & \cdots & & & & & \\ \cdots & & & & & & \end{pmatrix}$$

for some constant c. This proves by induction that Equation (3.7) holds.

Consider now the behaviour of ε on $\Delta\,\vec{b}_n$. Since ε is counit in $\mathcal{C}$ we have

$$(\varepsilon\otimes\mathrm{id})\Delta\,\vec{b}_n = 1\otimes\vec{b}_n$$

which implies with (3.7) that

$$\sum_{i=0}^{n} c_{n-i}\varepsilon(\vec{b}_i) = \delta_{0n}$$

This means that the series $\sum_j c_j x^j$ and $\sum_j \varepsilon(\vec{b}_j)x^j$ are inverse to each other with respect to convolution, which is equivalent to (3.8). ∎

3.9 Umbral Coalgebras

In this section we define the main structure in our description of the umbral calculus, the *umbral coalgebra*. This is a coalgebra with a sharply nesting operator, or, equivalently, a coalgebra equipped with a basis which behaves in a certain way with respect to the comultiplication. The correspondence to the polynomial case and the convolution will be evident.

Definition 3.9 *A coalgebra $\mathcal{C} = (\mathbb{V}, \Delta, \varepsilon)$ is called* **umbral** *if $\mathcal{C}$ has a sharply nesting hemimorphism.*

The behaviour of S-compatible bases for a hemimorphism S suggests to define a reference basis for particularly simple values of the c_i's from Proposition 3.18, namely $c_i = \delta_{i0}$.

Definition 3.10 *Let $\mathcal{C} = (\mathbb{V}, \Delta, \varepsilon)$ be a coalgebra. Then a basis $\mathcal{B} = (\vec{b}_i)_{i\in\mathbb{N}}$ of $\mathbb{V}$ is called an* **umbral basis** *if*

$$\begin{aligned} \Delta\,\vec{b}_n &= \sum_i \vec{b}_i \otimes \vec{b}_{n-i} \\ \varepsilon(\vec{b}_n) &= \delta_{0n} \end{aligned}$$

*holds for all $n \in \mathbb{N}$. Moreover, if $\mathcal{B}$ is S-compatible for some sharply nesting hemimorphism S, then we call it an S-***umbral basis**.

As a matter of fact, the existence of such a basis in $\mathcal{C}$ is equivalent to the existence of a sharply nesting hemimorphism.

Theorem 3.2 *A coalgebra $\mathcal{C} := (\mathbb{V}, \Delta, \varepsilon)$ is umbral if and only if $\mathcal{C}$ has an umbral basis $\mathcal{B} = (\vec{b}_i)_{i\in\mathbb{N}}$.*

Proof. Assume that an umbral basis $\mathcal{B} = (\vec{b}_i)_{i\in\mathbb{N}}$ of $\mathcal{C}$ exists. Then it is easy to check that the operator $S \in \mathrm{End}(\mathbb{V})$ defined by $S(\vec{b}_{i+1}) = \vec{b}_i$ for all $i \in \mathbb{N}$ and $S(\vec{b}_0) = \vec{0}$ is sharply nesting. Furthermore

$$\Delta\, S(\vec{b}_{n+1}) = \Delta\vec{b}_n = \sum_i \vec{b}_i \otimes \vec{b}_{n-i} = \sum_i \vec{b}_i \otimes S(\vec{b}_{n-i+1}) = (\mathrm{id} \otimes S)\Delta\,\vec{b}_{n+1}$$

so, S is hemimorphism of $\mathcal{C}$ and $\mathcal{C}$ is umbral.

For the other direction, assume that $\mathcal{C}$ is umbral, i.e., a sharply nesting hemimorphism S exists. Let $(\vec{b}'_i)_{i\in\mathbb{N}}$ be an S-compatible basis of $\mathcal{C}$. We show that an (S-)umbral basis $(\vec{b}_i)_{i\in\mathbb{N}}$ can be obtained from $(\vec{b}'_i)_{i\in\mathbb{N}}$.

Let $T^{-1} = \sum_i \varepsilon(\vec{b}'_i)S^i$. T^{-1} is well-defined as invertible operator since $\varepsilon(\vec{b}'_0) \neq 0$ is given by the fact that $\vec{b}'_0$ is a nontrivial multiple of $\vec{b}_0$. Then, by Prop. 3.14 we know that $(\vec{b}_i)_{i\in\mathbb{N}}$ for $\vec{b} := T(\vec{b}'_i)$ is S-compatible. In addition, by Prop. 3.18 we know that for $T = \sum_i c_i S^i$. Recalling that by the cocommutativity we have $\Delta = (T \otimes \mathrm{id})(T^{-1} \otimes \mathrm{id})\Delta = (T \otimes \mathrm{id})(\mathrm{id} \otimes T^{-1})\Delta$ and so $(\mathrm{id} \otimes T^{-1})\Delta = (T \otimes \mathrm{id})\Delta$, we can write

$$\Delta\,\vec{b}_n \;=\; \Delta T(\vec{b}'_n) = (\mathrm{id} \otimes T^{-1})\Delta\vec{b}'_n = (\mathrm{id} \otimes T^{-1}) \sum_i c_{n-i} \sum_j \vec{b}'_j \otimes \vec{b}'_{i-j}$$

$$= \sum_i c_{n-i} \sum_j \vec{b}_j \otimes T^{-1}(\vec{b}_{i-j}) = \sum_j \vec{b}_j \otimes \sum_{i,k} c_{n-i}\varepsilon(\vec{b}'_k)\vec{b}_{i-j-k}$$

$$= \sum_{j,m} \left(\sum_i c_{n-i}\varepsilon(\vec{b}'_{i-j-m}) \right) = \sum_j \vec{b}_j \otimes \vec{b}_{n-j}$$

The last equality holds since

$$\left(\sum_i c_{n-i}\varepsilon(\vec{b}'_{i-j-m}) \right) = [t^{n-j-m}] \left(\sum_i c_i t^i \right) \left(\sum_i \varepsilon(\vec{b}'_i) t^i \right) = \delta_{n-j-m,0}$$

where we denote by $[t^l]\alpha(t)$ the coefficient of t^l in the formal power series α. ∎

Sometimes different bases are considered, and one speaks of *non-standard* umbral calculus. In the following proposition we make clear that such a difference is not substantial for many cases.

Proposition 3.19 *A coalgebra $\mathcal{C} := (\mathbb{V}, \Delta, \varepsilon)$ is umbral if and only if there exists a basis $(\vec{v}_i)_{i\in\mathbb{N}}$ of $\mathbb{V}$ and a sequence $(c_i)_{i\in\mathbb{N}}$ of non zero constants in $\mathcal{K}$ such that*

$$\begin{aligned} \Delta\,\vec{v}_n &= \sum_i \frac{c_n}{c_i c_{n-i}}\,\vec{v}_i \otimes \vec{v}_{n-i} \\ \varepsilon(\vec{v}_n) &= c_0\delta_{0n} \end{aligned}$$

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

Proof. The proof is straightforward by noticing that the basis defined by $\vec{b}_n = \vec{v}_n / c_n$ is umbral. ∎

We still have to prove that umbral bases are unique with respect to S-compatibility to a sharply nesting hemimorphism, so that we may speak of *the* S-umbral basis.

Proposition 3.20 *Let $\mathcal{C}$ be an umbral coalgebra. Then for any sharply nesting hemimorphism S of $\mathcal{C}$ there is precisely one S-umbral basis.*

Proof. From the proof of Theorem 3.2 we know that an S-umbral basis exists for all sharply nesting hemimorphisms S. Assume now that $(\vec{b}_i)_{i\in\mathbb{N}}$ and $(\vec{b}'_i)_{i\in\mathbb{N}}$ are S-umbral, then they are in particular S-compatible. It follows from Proposition 3.14 that $\vec{b}'_i = T^{-1}(\vec{b}_i)$ for some invertible $T \in \mathrm{Hem}(\mathcal{C})$ and so

$$\Delta\,\vec{b}'_n = \Delta\,T^{-1}(\vec{b}_n) = (T^{-1} \otimes \mathrm{id})\vec{b}_n = \sum_j T^{-1}(\vec{b}_j) \otimes \vec{b}_{n-j} = \sum_j \vec{b}'_j \otimes \vec{b}_{n-j}$$

With $\Delta\,\vec{b}'_n = \sum_j \vec{b}'_j \otimes \vec{b}'_{n-j}$ this implies $\vec{b}'_i = \vec{b}_i$ for all $i \in \mathbb{N}$. ∎

As we saw before, any S-compatible basis can be obtained from another one applying an invertible S-invariant operator. Referred to the S-umbral basis this operator is called *associated operator*, as described in the following definition.

Definition 3.11 *Let S be a sharply nesting hemimorphism of $\mathcal{C}$ and $(\vec{b}_i)_{i\in\mathbb{N}}$ the S-umbral basis. Then for any S-compatible basis $(\vec{s}_i)_{i\in\mathbb{N}}$ we call the* **operator associated** *to $(\vec{s}_i)_{i\in\mathbb{N}}$ the operator R such that $\vec{s}_n = R^{-1}(\vec{b}_n)$ for all n.*

3.10 Some Formulas

In this section some of the well-known formulas of the umbral calculus are stated and proved in the framework presented above. In the following we suppose $\mathcal{C} = (\mathbb{V}, \Delta, \varepsilon)$ to be an umbral coalgebra.

Theorem 3.3 (First Expansion Theorem) *Let S be a sharply nesting hemimorphism of $\mathcal{C}$. Then for any hemimorphism T we have*

$$T = \sum_{k\geq 0} a_k S^k$$

with

$$a_k = \varepsilon\left(T(\vec{b}_k)\right)$$

for the S-umbral basis $(\vec{b}_i)_{i\in\mathbb{N}}$.

Proof. From Theorem 3.1 we know that T can be expressed as series in S. The form of the coefficients a_k also follows from the proof of Theorem 3.1 recalling that for umbral $(\vec{b}_i)_{i\in\mathbb{N}}$ the counit behaves like β^0. ∎

Corollary 3.3 *Let S be a sharply nesting hemimorphism of $\mathcal{C}$, $(\vec{s}_i)_{i\in\mathbb{N}}$ an S-compatible basis and R the associated operator. Then*

$$R^{-1} = \sum_{n\geq 0} \varepsilon(\vec{s}_n) S^n$$

Theorem 3.4 (Second Expansion Theorem) *Let S be a sharply nesting hemimorphism of $\mathcal{C}$, $(\vec{s}_i)_{i\in\mathbb{N}}$ an S-compatible basis and R the associated operator. Then for any hemimorphism T and any $\vec{v} \in \mathbb{V}$ it holds that*

$$\Delta T(\vec{v}) = \sum_{n\geq 0} (S^n RT(\vec{v})) \otimes \vec{s}_n$$

Proof. Denote by $(\vec{b}_i)_{i\in\mathbb{N}}$ the S-umbral basis $b_i = R(\vec{s}_i)$. Since $\Delta T = (T\otimes \mathrm{id})\Delta$ we only need to prove

$$\Delta\,\vec{v} = \sum_{n\geq 0} (S^n R(\vec{v}))\otimes \vec{s}_n$$

Let $\vec{v} = \sum_i v_i\vec{b}_i$. Since $(\vec{b}_i)_{i\in\mathbb{N}}$ is umbral it follows that the representation of $\Delta\,\vec{v}$ with respect to $\left(\vec{b}_i\otimes\vec{b}_j\right)_{i,j\in\mathbb{N}}$ is

$$\Delta\,\vec{v} = \begin{pmatrix} v_0 & v_1 & v_2 & v_3 & \cdots \\ v_1 & v_2 & v_3 & \cdots & \\ v_2 & v_3 & \cdots & & \\ v_3 & \cdots\cdots & & & \end{pmatrix}$$

The n-th row (or column) is given by the n-fold shifted representation of $\vec{v}$, this means

$$\Delta\,\vec{v} = \sum_n S^n(\vec{v})\otimes\vec{b}_n$$

On the other hand, since R is an hemimorphism, we have $(R\otimes\mathrm{id})(\mathrm{id}\otimes R^{-1})\Delta = \Delta R^{-1}R = \Delta$ and we get the proposition

$$\Delta\,\vec{v} = (R\otimes\mathrm{id})(\mathrm{id}\otimes R^{-1})\sum_n S^n(\vec{v})\otimes\vec{b}_n = \sum_n S^n R(\vec{v})\otimes\vec{s}_n$$

∎

Theorem 3.5 (Binomial Theorem) *Let S be a sharply nesting hemimorphism of $\mathcal{C}$ and $(\vec{b}_i)_{i\in\mathbb{N}}$ the S-umbral basis. Then a basis $(\vec{s}_i)_{i\in\mathbb{N}}$ is S-compatible if and only if*

$$\Delta\vec{s}_n = \sum_{k=0}^{n}\vec{s}_k\otimes\vec{b}_{n-k}$$

Proof. One direction directly follows from Theorem 3.4 by choosing $\vec{v} = \vec{s}_n$, R the associated operator of $(\vec{s}_i)_{i\in\mathbb{N}}$ and $T = \mathrm{id}$.

For the other direction, assume that the equation above holds for all n. We prove that $(\vec{s}_i)_{i\in\mathbb{N}}$ is S-compatible. For $n\geq 1$ consider

$$\Delta\,S(\vec{s}_n) = (\mathrm{id}\otimes S)\Delta\vec{s}_n = \sum_{k=0}^{n}\vec{s}_k\otimes S(\vec{b}_{n-k}) = \sum_{k=0}^{n-1}\vec{s}_k\otimes\vec{b}_{n-1-k} = \Delta\vec{s}_{n-1}$$

Since Δ is injective, this proves that $S(\vec{s}_n) = \vec{s}_{n-1}$. A similar reasoning proves $S(\vec{s}_0) = \vec{0}$. ∎

Corollary 3.4 *Let S be a sharply nesting hemimorphism of $\mathcal{C}$, $(\vec{b}_i)_{i\in\mathbb{N}}$ the S-umbral basis and $(\vec{s}_i)_{i\in\mathbb{N}}$ an S-compatible basis. Then*

$$\vec{s}_n = \sum_k \varepsilon(\vec{s}_k)\vec{b}_{n-k}$$

Proof. From the Binomial Theorem above we know that

$$\Delta\vec{s}_n = \sum_{k=0}^{n} \vec{s}_k \otimes \vec{b}_{n-k}$$

Applying $(\varepsilon \otimes \mathrm{id})$ on both sides of the equation we get

$$(\varepsilon \otimes \mathrm{id})\Delta\vec{s}_n = \sum_{k=0}^{n} \varepsilon(\vec{s}_k) \otimes \vec{b}_{n-k}$$

Recalling the counitary property $(\varepsilon \otimes \mathrm{id})\Delta\vec{s}_n = 1 \otimes \vec{s}_n$ the assertion directly follows. ∎

In other words, any S-compatible basis $(\vec{s}_i)_{i\in\mathbb{N}}$ is determined by the value of ε on it (evaluation at zero in the usual polynomial case).

Proposition 3.21 (Recurrence Formula) *If $(\vec{s}_i)_{i\in\mathbb{N}}$ is an S-compatible basis for some sharply nesting hemimorphism S then for every sharply nesting hemimorphism T there exists a sequence $(a_i)_{i\in\mathbb{N}}$ of constants such that*

$$T(\vec{s}_n) = \sum_k a_{n-k}\vec{s}_k \qquad (*)$$

Conversely, if such a sequence exists for some sharply nesting hemimorphism T then $(\vec{s}_i)_{i\in\mathbb{N}}$ is S-compatible for some S.

Proof. Assume that $(\vec{s}_i)_{i\in\mathbb{N}}$ is S-compatible basis for some sharply nesting hemimorphism S. Since $T \in \mathrm{Hem}(\mathcal{C})$ we have

$$T = \alpha(S) = \sum_{k\geq 0} a_k S^k$$

for some $\alpha \in \mathcal{K}[[t]]$, so $(*)$ holds. Assume now that $(*)$ holds for some sharply nesting hemimorphism T and define S by $S(\vec{s}_0) = \vec{0}$ and $S(\vec{s}_{n+1}) = \vec{s}_n$ for all $n \in \mathbb{N}$. Since $(\vec{s}_i)_{i\in\mathbb{N}}$ is a basis, the operator S is sharply nesting . An easy verification on $(\vec{s}_i)_{i\in\mathbb{N}}$ shows that $TS = ST$, so S is a hemimorphism. ∎

3.11 The Umbral Group

In the usual polynomial framework for umbral calculus, the *umbral composition* of a polynomial $p(x)$ with respect to a basic sequence $(q_i(x))_{i\in\mathbb{N}}$ is defined as $p(\mathbf{q}) = \sum_i a_i q_i(x)$ for $p(x) = \sum_i a_i x^i$. An operator on polynomials acting like a substitution of x^i by $q_i(x)$ is then called an *umbral operator*. In our framework an umbral operator maps umbral bases of $\mathbb{V}$ onto umbral bases, or, equivalently, is an automorphism of the umbral coalgebra.

Definition 3.12 *Let $\mathcal{C}$ be an umbral coalgebra. Then we call the automorphisms of $\mathcal{C}$* **umbral operators** *on $\mathcal{C}$ and the group* $\mathrm{Aut}(\mathcal{C})$ *of automorphisms of $\mathcal{C}$ is also called the* **umbral group** *on $\mathcal{C}$.*

The umbral operators build a structure like $\mathcal{K}[[t]]^{inv}$, the set of formal power series closed under substitution, i.e., composition.

Theorem 3.6 *Let $\mathcal{C}$ be an umbral coalgebra. Then we have*

$$\mathrm{Aut}(\mathcal{C}) \cong \mathcal{K}[[t]]^{inv}$$

Proof. From Proposition 3.3 we know that $\mathrm{End}(\mathbb{V}) \cong \mathrm{End}_C(\mathbb{V}^*)$. This means that for the coalgebra $\mathcal{C}$ we have $\mathrm{Aut}(\mathcal{C}) \cong \mathrm{Aut}_C(\mathcal{C}^*)$. In addition we have the well-known fact that $\mathbb{V}^* \cong \mathcal{K}[[t]]$ (cf. also Proposition 3.2), so, we only need to prove that

$$\mathrm{Aut}_C(\mathcal{K}[[t]]) \cong \mathcal{K}[[t]]^{inv}$$

Consider an arbitrary $\tau \in \mathrm{Aut}_C(\mathcal{K}[[t]])$ and $\alpha = \sum_i a_i t^i \in \mathcal{K}[[t]]$. Since τ is an algebra isomorphism, from, say, $\tau(t) = \beta$ it follows that $\tau(t^i) = \beta^i$ for all i. Furthermore, since the β^i's must form a basis of $\mathcal{K}[[t]]^{inv}$, we have $\beta(0) = 0$ and t appears with nonzero coefficient in β. Since τ is continuous, we have

$$\tau(\alpha(t)) = \sum_i a_i \tau(t^i) = \sum_i a_i \beta^i(t)$$

Associating each τ to the corresponding β gives the isomorphism we wanted to find. Applying the automorphism τ means then to substitute a given formal power series. It is easy to see that the operation corresponding to the multiplication of continuous automorphisms on $\mathcal{K}[[t]]$ is the substitution. In fact, consider τ, the associated $\beta \in \mathcal{K}[[t]]^{inv}$ and $\tau' \in \mathcal{K}[[t]]$ with the associated β'. Then for all $\alpha \in \mathcal{K}[[t]]$ we have $(\tau\tau')(\alpha) = \tau(\alpha \circ \beta') = \alpha \circ \beta' \circ \beta$. So, $\beta' \circ \beta$ is associated to $\tau\tau'$. ∎

In the following proposition some more facts on hemimorphisms and umbral bases are described. We point out in particular the importance of item 3.

Proposition 3.22 *Let T be an umbral operator of $\mathcal{C}$. Then the following statements hold .*

1. *The map $S \mapsto TST^{-1}$ is an automorphism of* $\mathrm{Hem}(\mathcal{C})$.

2. *If $(\vec{b}_i)_{i\in\mathbb{N}}$ is an S-compatible basis for some sharply nesting S then $(T(\vec{b}_i))_{i\in\mathbb{N}}$ is Q-compatible for some Q.*

3. *If $(\vec{b}_i)_{i\in\mathbb{N}}$ is an umbral basis for $\mathcal{C}$, then also $(T(\vec{b}_i))_{i\in\mathbb{N}}$ is.*

4. *If S is a sharply nesting hemimorphism, then also TST^{-1} is.*

5. *If $S = \alpha(Q)$ for some $\alpha \in \mathcal{K}[[t]]$, then $TST^{-1} = \alpha(TQT^{-1})$.*

Proof. Items 2 and 3 directly follows from the definition of umbral operators. Item 1 follows from 4.

In order to prove 4, consider an S-compatible basis $(\vec{b}_i)_{i\in\mathbb{N}}$. Since T is umbral operator we have that $(T(\vec{b})_i)_{i\in\mathbb{N}}$ is Q-compatible for some Q. One easily verifies that $Q = TST^{-1}$, since $QT(\vec{b}_{n+1}) = TST^{-1}T(\vec{b}_{n+1}) = TS(\vec{b}_{n+1}) = T(\vec{b}_n)$ for all n. ∎

We give the next theorem without proof. It states a general version of the known theorem describing the connection between the operators associated to a compatible basis and those associated to the basis obtained applying an umbral operator (see, for instance, [Rom84]).

Theorem 3.7 *Let S be a sharply nesting hemimorphism of $\mathcal{C}$ and $(\vec{b}_i)_{i\in\mathbb{N}}$ an S-umbral basis. Let $Q, P \in \mathrm{Hem}(\mathcal{C})$ be sharply nesting and $(\vec{v}_i)_{i\in\mathbb{N}}$ and $(\vec{t}_i)_{i\in\mathbb{N}}$ be Q(resp. P)-compatible bases with associated operator V (resp. T). Let $(\vec{q}_i)_{i\in\mathbb{N}}$ and $(\vec{p}_i)_{i\in\mathbb{N}}$ be umbral with respect to Q and P, respectively. Furthermore, let $v, p, q, t \in \mathcal{K}[[t]]$ be such that*

$$V = v(S), \quad P = p(S), \quad Q = q(S), \quad T = t(S)$$

Define the operator U^t by $U^t\vec{b}_n := \vec{t}_n$ for all n and $(\vec{r}_i)_{i\in\mathbb{N}}$ by $\vec{r}_n = U^t\vec{v}_n$.

Then $(\vec{r}_i)_{i\in\mathbb{N}}$ is compatible basis with respect to the sharply nesting operator

$$Tv(P) = t(S)v(p(S))$$

with associated operator

$$q(p(S))$$

and corresponding umbral basis

$$(U^p\vec{q}_i)_{i\in\mathbb{N}}$$

where $U^p\vec{b}_n := \vec{p}_n$ for all $n \in \mathbb{N}$.

3.12 Umbral Operators and Recursive Matrices

In this section the matrix representation of the umbral operators is studied. Let $U \in \mathrm{Aut}(\mathcal{C})$ be such that $U\vec{b}_i = \vec{v}_i$ for some umbral bases $(\vec{b}_i)_{i\in\mathbb{N}}$ and $(\vec{v}_i)_{i\in\mathbb{N}}$. Then the following theorem describes the matrix representation of U with respect to the basis $(\vec{b}_i)_{i\in\mathbb{N}}$. As a matter of fact, the theorem describes the more general case where the basis $(\vec{v}_i)_{i\in\mathbb{N}}$ is not necessarily umbral but S-compatible for some sharply nesting hemimorphism S.

Theorem 3.8 *Let $(\vec{b}_i)_{i\in\mathbb{N}}$ be S-umbral, $(\vec{s}_i)_{i\in\mathbb{N}}$ Q-umbral and $(\vec{v}_i)_{i\in\mathbb{N}}$ Q-compatible bases with associated operator R, i.e., such that $\vec{v}_n = R^{-1}(\vec{s}_n)$ for all n. Let furthermore $Q = q(S)$, $S = q^{inv}(Q)$ and $R = r(Q)$ for some $q, r \in \mathcal{K}[[t]]$ and denote by $\left(\sigma_n^i\right)_{i,n\in\mathbb{N}}$ and $\left(\rho_n^i\right)_{i,n\in\mathbb{N}}$ the transformation coefficients defined by*

$$\vec{s}_n = \sum_i \sigma_n^i\, \vec{b}_i \quad \textit{and} \quad \vec{v}_n = \sum_i \rho_n^i\, \vec{b}_i$$

Then for the row generating functions of $\left(\sigma_n^i\right)_{i,n\in\mathbb{N}}$ and $\left(\rho_n^i\right)_{i,n\in\mathbb{N}}$ it holds that

$$\begin{aligned}
\mathfrak{S}^k(t) &:= \sum_n \sigma_n^k\, t^n = q^{inv}(t)^k \\
\mathfrak{R}^k(t) &:= \sum_n \rho_n^k\, t^n = r^{-1}(t)\, q^{inv}(t)^k
\end{aligned}$$

Proof. Let $r^{-1}(t) = \sum_i e_i t^i$. Then the assertion follows mainly by coefficient comparison from expressing $\Delta\vec{v}_n$ in three different ways with respect to the basis $\left(\vec{b}_i \otimes \vec{b}_j\right)_{i,j\in\mathbb{N}}$. From the fact that $(\vec{s}_i)_{i\in\mathbb{N}}$ is an umbral basis we have

$$\begin{aligned}
\Delta\,\vec{v}_n &= \sum_l e_l\,\Delta\, S^l(\vec{s}_n) = \sum_h e_{n-h}\Delta\,\vec{s}_h = \sum_h e_{n-h} \sum_i \vec{s}_i \otimes \vec{s}_{h-i} = \\
&= \sum_h e_{n-h} \sum_{i,j,k} \sigma_i^k\, \sigma_{h-i}^j\, \vec{b}_k \otimes \vec{b}_j = \sum_{j,k} \underbrace{\left(\sum_h e_{n-h} \sum_i \sigma_i^k\, \sigma_{h-i}^j\right)}_{=[t^n]r^{-1}(t)\mathfrak{S}^k(t)\mathfrak{S}^j(t)} \vec{b}_k \otimes \vec{b}_j
\end{aligned} \tag{3.9}$$

On the other hand, if we represent $\vec{s}_h$ by means of $(\vec{b}_i)_{i\in\mathbb{N}}$ we obtain

$$\begin{aligned}
\Delta\,\vec{v}_n &= \sum_h e_{n-h}\Delta\,\vec{s}_h = \sum_h e_{n-h} \sum_i \sigma_h^i\,\Delta\,\vec{b}_i = \sum_{h,i,j} e_{n-h}\sigma_h^i \vec{b}_j \otimes \vec{b}_{i-j} \\
&= \sum_{j,k} \underbrace{\left(\sum_h e_{n-h}\sigma_h^{j+k}\right)}_{=[t^n]r^{-1}(t)\mathfrak{S}^{j+k}(t)} \vec{b}_k \otimes \vec{b}_j
\end{aligned} \tag{3.10}$$

Finally, we directly have

$$\Delta\, \vec{v}_n = \sum_i \rho_n^i \Delta\, \vec{b}_i = \sum_{i,j} \rho_n^i\, \vec{b}_j \otimes \vec{b}_{i-j} = \sum_{j,k} \underbrace{\rho_n^{j+k}}_{=[t^n]\mathfrak{R}^{j+k}(t)} \vec{b}_k \otimes \vec{b}_j \tag{3.11}$$

Since (3.9) and (3.10) hold for all $n \in \mathbb{N}$ we can deduce that $\mathfrak{S}^{j+k}(t) = \mathfrak{S}^j(t)\mathfrak{S}^k(t)$ and so $\mathfrak{S}^n(t) = (\mathfrak{S}^1(t))^n$. With (3.11) this implies

$$\mathfrak{R}^n(t) = r^{-1}(t)\left(\mathfrak{S}^1(t)\right)^n$$

It is still left to prove that $\mathfrak{S}^1(t) = q^{inv}(t)$. We prove the equivalent statement

$$\mathfrak{S}^1(Q) = S, \qquad \text{i.e.,} \qquad \sum_n \sigma_n^1 Q^n = S$$

It is easy to check that for all k we have

$$\begin{aligned}\left(\sum_n \sigma_n^1 Q^n\right)\vec{s}_k &= \sum_n \sigma_n^1 \vec{s}_{k-n} = \sum_n \sigma_n^1 \sum_j \sigma_{k-n}^j\, \vec{b}_j \\ &= \sum_j \underbrace{\left(\sum_n \sigma_n^1 \sigma_{k-n}^j\right)}_{=[t^k]\mathfrak{S}^1(t)\mathfrak{S}^j(t)} \vec{b}_j = \sum_j \sigma_k^{j+1}\vec{b}_j = S\left(\sum_j \sigma_k^j \vec{b}_j\right) = S(\vec{s}_k)\end{aligned}$$

∎

Since Q and S play a somehow symmetric role in the last theorem, this can be restated as follows, describing a class of inverse relations.

Proposition 3.23 *Let the matrix $A := (a_n^i)_{i,n}$ be given by $r, q \in \mathcal{K}[[t]]$, such that q is invertible under composition and r is invertible under convolution, and*

$$\sum_n a_n^k t^n = r(t)q(t)^k$$

Then the matrix $B := (b_n^i)_{i,n}$ defined by

$$\sum_n b_n^k t^n = r^{-1}(t)q^{inv}(t)^k$$

is inverse to A. Explicitly, this means that

$$d_k = \sum_n a_n^k c_n \iff c_k = \sum_n b_n^k d_n$$

holds for any sequences $(d_i)_{i\in\mathbb{N}}$, $(c_i)_{i\in\mathbb{N}}$.

Proof. The statement is just a reformulation of the preceding theorem, considering the matrix $(b_n^k)_{k,n}(=(\rho_n^k)_{k,n})$ corresponding to the transformation $\vec{b}_n \mapsto \vec{s}_n$ and its inverse. ∎

We follow the notation from [BBN82] and say that $A = (r(t), q(t))$ is **the recursive matrix** defined by r and q if the elements of the matrix A satisfy the conditions given in the proposition. Note that such matrices have upper triangular shape. Many properties of recursive matrices are described in [BBN82].

An interesting property of such bi-dimensional sequences follows from a natural question: Given two sharply nesting hemimorphisms S, Q and an hemimorphism T. If we know the formal power series representation of T with respect to S, say $T = \alpha(S)$ for some $\alpha \in \mathcal{K}[[t]]$, how can we describe the coefficients of $\beta \in \mathcal{K}[[t]]$ for $T = \beta(Q)$.

We know by Theorem 3.3 that for $T = \sum_n b_n Q^n$ we have $b_n = \varepsilon(T(\vec{s}_n))$ for $(\vec{s}_i)_{i\in\mathbb{N}}$ Q-umbral basis. On the other hand, using the notation given in Theorem 3.8, we can write

$$T(\vec{s}_n) = \alpha(S)(\vec{s}_n) = \left(\sum_k a_k S^k\right) \sum_i \sigma_i^n \vec{b}_i = \sum_l \left(\sum_i \sigma_i^n a_{i-l}\right) \vec{b}_l$$

and from this it follows that

$$b_n = \varepsilon(T(\vec{s}_n)) = \sum_i \sigma_i^n a_i$$

(recall that for $(\vec{b}_i)_{i\in\mathbb{N}}$ umbral basis, $\varepsilon = \beta^0$ holds).

In other words, the formal power series representation of T with respect to Q is the generating function of the sums $\sum_i \sigma_i^n a_i$, where $T = \sum_i a_i S^i$. With the information that $S = q^{inv}(Q)$ we have $T = \alpha(S) = \alpha(q^{inv}(Q))$ and the following proposition is proven for the case of a recursive matrix of the form $(1, q^{inv}(t))$. The general case is straightforward, when we consider the basis associated to the operator $r^{-1}(Q)$.

Proposition 3.24 *Given a recursive matrix $B = (b_i^n)_{n,i} = (r^{-1}(t), q^{inv}(t))$ and a sequence of numbers $(a_i)_{i\in\mathbb{N}}$, the generating function of the sequence*

$$\left(\sum_i a_i b_i^n\right)_{n\in\mathbb{N}}$$

is given by $r^{-1}(t)\alpha(q^{inv}(t))$, where $\alpha(t) = \sum_i a_i t^i$ is the generating function for the sequence $(a_i)_{i\in\mathbb{N}}$.

As an example, we give two particular cases of the application of the last proposition. The first is the generating function for the column-sums of the matrix and the second for the polynomials in x with coefficients from the columns.

Corollary 3.5 *Let the sequence $(b_i^n)_{n,i}$ be given by the recursive matrix $B = (r(t), q(t))$, then*

$$r(t)\frac{1}{1-q(t)} \qquad \textit{is the generating function for} \qquad \sum_i b_i^n$$

$$r(t)\frac{1}{1-xq(t)} \qquad \textit{is the generating function for} \qquad \sum_i b_i^n x^i$$

Let us consider now, for two sharply nesting hemimorphisms S and Q as before, the operator W such that $S = WQ$. The existence of T is ensured by the corresponding equation over formal power series. We know that $Q = q(t)$, so we can equivalently determine $w(t) \in \mathcal{K}[[t]]$, such that

$$t = w(t)q(t)$$

since $q(0) = 0$ and $q(1) \neq 0$, and $W = w(S)$.

The existence of such an operator W has an interesting consequence for the recursive matrix $(\rho_n^i)_{n,i} = (r^{-1}(t), q^{inv}(t))$. If we write $W = \sum_i w_i S^i$ and apply both S and WQ to the Q-compatible basis corresponding to $r^{-1}(S)$, then we get (using the notation from Theorem 3.8)

$$\begin{aligned} S(\vec{v}_{n+1}) &= WQ(\vec{v}_{n+1}) = W(\vec{v}_n) = \left(\sum_i w_i S^i\right) \sum_j \rho_n^i \vec{b}_i \\ &= \sum_{i,k} w_k \rho_n^i \vec{b}_{i-k} = \sum_l \left(\sum_i \rho_n^i w_{l+k}\right) \vec{b}_l \end{aligned}$$

and

$$S(\vec{v}_{n+1}) = \sum_i \rho_{n+1}^{i+1} \vec{b}_i$$

Equating coefficients we get a recurrence along the columns

$$\rho_{n+1}^{i+1} = \sum_k w_k \rho_n^{i+k}$$

Surprisingly, this recurrence does not depend on $r(t)$.

Note that computing $w(t)$ as proposed above, i.e., to compute $w(t) = tq^{-1}(t)$ from a given $q^{inv}(t)$ corresponds to invert with respect to convolution the compositional inverse of the series $q^{inv}(t)$.

Corollary 3.6 *Let the sequence $(\rho_i^n)_{n,i}$ be a recursive matrix, then there exists a sequence of constants $(w_i)_{i\in\mathbb{N}}$ such that*

$$\rho_{n+1}^{i+1} = \sum_k w_k \rho_n^{i+k}$$

holds for all n and i in $\mathbb{N}$. In addition, if $(\rho_i^n)_{n,i} = (r(t), q^{inv}(t))$, then $\sum_i w_i t^i$ is the inverse of $q(t)$ with respect to convolution.

3.13 A non-polynomial application: Factorial functions

In the case that the field $\mathcal{K}$ has characteristic zero, the most natural, in some sense canonical, example of a nested vector space doubtlessly is the vector space $\mathcal{K}[x]$ of univariate polynomials. The polynomial coalgebra $(\mathcal{K}[x], \Delta, \varepsilon)$ is defined by $\Delta : p(x) \mapsto p(x+y) \in \mathcal{K}[x] \otimes \mathcal{K}[x]$ and $\varepsilon(p(x)) = p(0)$ and describes the classical umbral calculus in one variable.

On the other hand, the polynomial coalgebra is not the only structure which can be embedded into our framework.

An example of an umbral structure which is not isomorphic to the space of polynomials is given by the so-called *factorial functions*, as introduced in a work by Barnabei, Brini and Nicoletti [BBN86].

Let us consider bi-infinite sequences over $\mathcal{K}$, this means the set $\mathbb{H} := \mathcal{K}^{\mathbb{Z}}$, and define the shift operator E on $\mathbf{f} \in \mathbb{H}$ by $(E\mathbf{f})(x) = \mathbf{f}(x+1)$ for all $x \in \mathbb{Z}$. The *forward difference operator* Δ on $\mathbb{H}$ is defined by $\Delta := E - I$, where I is the identity operator, so $\Delta\mathbf{f}(x) = \mathbf{f}(x+1) - \mathbf{f}(x)$. *Factorial functions* are then sequences in $\mathbb{H}$ respecting the following definition.

Definition 3.13 *A function $\mathbf{f} \in \mathbb{H}$ is called a factorial function of degree $n \in \mathbb{N}$ if*

$$\Delta^{n+1}\mathbf{f} = \mathbf{0} \qquad \textit{and} \qquad \Delta^n \mathbf{f} \neq \mathbf{0}$$

Factorial functions of degree zero are precisely the non-zero constant functions, while for the zero function $\mathbf{0}$ we define the degree to be $+\infty$.

Denote by $\mathbb{F}$ the set of all factorial functions in $\mathbb{H}$. Then $\mathbb{F}$ has the structure of a $\mathcal{K}$-vector space, with the usual component-wise addition and constant multiplication.

As a matter of fact, if $\mathcal{K}$ has characteristic zero, it is evident that one can look at factorial functions as the *restriction* of polynomial functions over the integers. If $\mathcal{K}$ has characteristic $p > 0$, then this correspondence is not valid any more. Namely, consider that in this case, if $\mathbf{f}$ is the restriction of a polynomial function, then $\mathbf{f}(x+p) = \mathbf{f}(x)$ must be true for all x. As it will become clear below, this periodicity property does not hold for all factorial functions $\mathbf{f}$, when the characteristic is non-zero.

First, note that from the definition of factorial functions it follows that Δ is a nesting operator for $\mathbb{F}$. In order to show that Δ is sharply nesting, we need a finitely countable Δ-compatible basis for $\mathbb{F}$.

Let us define a sequence $(\mathbf{b}_i)_{i\in\mathbb{N}}$ of functions in $\mathbb{H}$ by

$$\begin{aligned} \mathbf{b}_0(x) &:= 1 \quad \text{for every} \quad x \in \mathbb{Z} \\ \mathbf{b}_n(0) &:= 0 \quad \text{for every} \quad n \in \mathbb{Z}^+ \\ \mathbf{b}_n(x+1) = \mathbf{b}_{n-1}(x) + \mathbf{b}_n(x) & \quad \text{for every} \quad x \in \mathbb{Z}, n \in \mathbb{Z}^+ \end{aligned}$$

The functions $\mathbf{b}_n$ are the so-called *p-binomial coefficients*, where p is the characteristic of $\mathcal{K}$, this means

$$\mathbf{b}_n(x) = \binom{x}{n}_p$$

and the $\mathbf{b}_n$'s take values in $\mathbb{Z}_p$.

We have that $\Delta \mathbf{b}_n = \mathbf{b}_{n-1}$ for $n \geq 1$ and $\Delta \mathbf{b}_0 = \mathbf{0}$, so $\mathbf{b}_n$ is a factorial function of degree n.

We define a linear functional $\varepsilon : \mathbb{H} \longrightarrow \mathcal{K}$ by $\varepsilon(\mathbf{f}) = \mathbf{f}(0)$. Then the following result holds (see [MNR81] for a proof).

Theorem 3.9 *A function* $\mathbf{f}$ *in* $\mathbb{H}$ *is a factorial function of degree* $n \in \mathbb{N}$ *if and only if*

$$\mathbf{f} = \sum_{k=0}^{n} a_k \mathbf{b}_k$$

with $a_n \neq 0$. *Additionally, if this is the case we have*

$$a_k = \varepsilon(\Delta^k \mathbf{f})$$

This means that the $\mathbf{b}_n$'s form a basis of $\mathbb{F}$. Additionally, $(\mathbf{b}_i)_{i \in \mathbb{N}}$ is a nesting basis for $\mathbb{F}$ which is Δ-compatible.

This way we can impose an umbral structure on $\mathbb{F}$, taking as sharply nesting hemimorphism Δ, and as Δ-umbral basis the sequence $(\mathbf{b}_i)_{i \in \mathbb{N}}$.

All the assertions in [BBN86] then can be expressed in our framework. Also the classical umbral calculus, i.e., the umbral calculus on univariate polynomials, can be described by means of factorial functions in the following sense. Assume that $\mathcal{K}$ has characteristic zero, then to each polynomial $p \in \mathcal{K}[x]$ of, say, degree n with

$$p = \sum_{i=0}^{n} c_i x^i$$

we associate the factorial function $\mathbf{f} \in \mathbb{F}$ defined by

$$\mathbf{f} = \sum_{i=0}^{n} \frac{i!}{n!} c_i \mathbf{b}_i$$

This means that, if $\mathcal{K}$ has characteristic zero, then $\mathbb{F}$ is canonically isomorphic to the vector space $\mathcal{K}[x]$ of polynomial functions (as we said, each $\mathbf{f}$ is simply the sequence of values taken by a polynomial function over the integers).

In the case where $\mathcal{K}$ has non-zero characteristic the isomorphism does not hold any more, although the umbral coalgebraic structure still remains valid. In this sense, factorial functions over a field of non-zero characteristic provide an example of an umbral coalgebra which is not isomorphic to the polynomial coalgebra.

As a further remark, note that in particular finite linear recurrences over finite fields fall into this paradigm.

Bibliography

[Abr71] S. A. Abramov. On the summation of rational functions. *Zh. vychisl. mat. Fiz.*, 11:1071 – 1075, 1971. English transl. in USSR Comput. Math. Phys.

[Abr75] S. A. Abramov. Rational component of the solution of a first-order linear recurrence relation with a rational right hand side. *Zh. vychisl. mat. Fiz.*, 14:1035–1039, 1975. English transl. in USSR Comput. Math. Phys.

[BBN82] M. Barnabei, A. Brini, and G. Nicoletti. Recursive matrices and umbral calculus. *J. Algebra*, 75:546–573, 1982.

[BBN86] M. Barnabei, A. Brini, and G. Nicoletti. A general umbral calculus. *Adv. Math., Suppl. Stud*, 10:221–244, 1986.

[Bou89] N. Bourbaki. *Elements of Mathematics: General Topology. Chapters 1-4*. Springer-Verlag, Berlin Heidelberg, 1989. 2nd printing.

[BS93] Manuel Bronstein and Bruno Salvy. Full partial fraction decomposition of rational functions. In Manuel Bronstein, editor, *ISSAC'93*, pages 157–160. ACM Press, July 1993.

[Buc91] A. Di Bucchianico. *Polynomials of convolution type*. PhD thesis, University of Groningen, Netherlands, 1991.

[C+93] G. E. Collins et al. A SACLIB 1.1 user's guide. Technical report, RISC-Linz, Johannes Kepler University, Austria, 1993. RISC-Linz Report Series n. 93-19.

[CFG+86] B. W. Char, G. J. Fee, K. O. Geddes, G. H. Gonnet, M. B. Monagan, and S. M. Watt. A tutorial introduction to MAPLE. *J. Symb. Comp.*, Vol.2, n.2, 1986.

[CGGG83] B. W. Char, K. O. Geddes, W. M. Gentleman, and G. H. Gonnet. The design of maple: A compact, portable and powerful computer algebra system. In *EUROCAL '83, Proceedings of the ISSAC*, pages 101–115. Springer, 1983.

[CNP84] L. Cerlienco, G. Nicoletti, and F. Piras. Coalgebre e calcolo umbrale. *Rend. Sem. Mat. Fis. Milano*, 54:79–100, 1984.

[CNP85] L. Cerlienco, G. Nicoletti, and F. Piras. Umbral Calculus. In J. Grabmeier and A. Kerber, editors, *Actes du Séminaire Lotharingien de Combinatoire 11 session, Septembre 1985*, pages 1–27. Publications de l'I.R.M.A. 1985 266/S–11.

[CNP86] L. Cerlienco, G. Nicoletti, and F. Piras. Polynomial sequences associated with a class of incidence coalgebras. *Ann. Discr. Math.*, 30:159–169, 1986.

[CP84] L. Cerlienco and F. Piras. Coalgebraic aspects of the umbral calculus. *Rend. Sem. Mat. Brescia*, 7:205–217, 1984.

[CZ94] E. Clarke and X. Zhao. Combining symbolic computation and theorem proving: some problems of Ramanujan. In *12th International Conference on Automated Deduction CADE*, pages 758–763, 1994.

[Dix82] J. D. Dixon. Exact solution of linear equations using p-adic expansions. *Numer. Math.*, 40:137–141, 1982.

[FT89] I. Foster and S. Taylor. *Strand - New Concepts in Parallel Programming.* Prentice-Hall, 1989.

[GCL92] K. O. Geddes, S. R. Czapor, and G. Labahn. *Algorithms for Computer Algebra.* Kluwer, Boston, 1992.

[GK84] R. T. Gregory and E. V. Krishnamurthy. *Methods and Applications of Error-Free Computation.* Springer Verlag, 1984.

[Gos78] R. W. Gosper. Decision procedure for indefinite hypergeometric summation. *Proc. Natl. Acad. Sci. USA*, 75:40 –42, 1978.

[Gui79] A. Guinand. The umbral method: A survey of elementary mnemonic and manipulative uses. *Amer. Math. Monthly*, 86:187–195, 1979.

[H+92] H. Hong et al. A PACLIB user manual. Technical report, RISC-Linz, Johannes Kepler University, Austria, 1992. RISC-Linz Report Series n. 92-32.

[II81] E.C. Ihrig and M.E.H. Ismail. A q-umbral calculus. *J. Math. Anal. Appl.*, 84:178–207, 1981.

[JR79] S.A. Joni and G.-C. Rota. Coalgebras and algebras in combinatorics. *Stud. Appl. Math.*, 61:93–139, 1979.

[Kar81] M. Karr. Summation in finite terms. *Journal of the ACM*, 28:305–350, 1981.

[Kar85] M. Karr. Theory of summation in finite terms. *J. Symb. Comp.*, 1:303–315, 1985.

[Kir79] P. Kirschenhofer. Binomialfolgen, Shefferfolgen und Faktorfolgen in den q-Analysis. *Sitzungber. Abt. II Österr. Akad. Wiss. Math. Naturw. Kl.*, 188:263–315, 1979.

[Knu81] D. Knuth. *The Art of Computer Programming*, volume 2. Addison Wesley Publishing Company, 1981.

[Knu93] Donald E. Knuth. Convolution polynomials. *The Mathematica Journal*, 2(4):67–78, 1993.

[Kob77] N. Koblitz. *p-adic Numbers, p-adic Analysis and Zeta Functions*. Springer Verlag, 1977.

[Kri85] E. V. Krishnamurthy. *Error-Free Polynomial Matrix Computations*. Springer Verlag, 1985.

[Kri93] E. V. Krishnamurthy. Algebraic transformation approach for parallelism. In L. Kronsjo, editor, *Advances in Parallel Algorithms*, pages 151–178. Blackwell, 1993.

[Lim93a] C. Limongelli. *The Integration of Symbolic and Numeric Computation by p-adic Construction Methods*. PhD thesis, University of Rome "La Sapienza" Italy, 1993. Technical Report, Collection of theses, V-93-4.

[Lim93b] C. Limongelli. On an efficient algorithm for big rational number computations by parallel p-adics. *J. Symb. Comp.*, Vol.15, n.2, 1993.

[Lip88] J. D. Lipson. *Elements of Algebra and Algebraic Computing*. Addison Wesley Publishing Company, 1988.

[LL93] C. Limongelli and H. W. Loidl. Rational number arithmetic by parallel p-adic algorithms. In Springer Verlag, editor, *Proc. of Second International Conference of the Austrian Center for Parallel Computation (ACPC)*, volume Vol. n. 734 of *LNCS*, 1993.

[LP94] C. Limongelli and R. Pirastu. Exact solution of linear equation systems over rational number by parallel p-adic arithmetic. In B. Buchberger and J. Volkert, editors, *Parallel Processing: CONPAR 94 - VAPP VI*, volume 854 of *LNCS*. Springer Verlag, 1994.

[LP96] C. Limongelli and R. Pirastu. p-adic arithmetic and parallel symbolic computation: an implementation for solving linear systems over rationals. *Computers and Artificial Intelligence*, 14(1):35–62, 1996.

[LT92] C. Limongelli and M. Temperini. Abstract specification of structures and methods in symbolic mathematical computation. *Theoretical Computer Science*, 104:89–107, October 1992. Elsevier Science Publisher.

[LT93] C. Limongelli and M. Temperini. On the uniform representation of mathematical data structures. In A. Miola, editor, *Design and Implementation of Symbolic Computation Systems, International (Symposium Disco '93)*, volume 722 of *LNCS*. Springer Verlag, 1993.

[Luc91] E. Lucas. *Théorie des nombres*, volume 1. Gauthier Villars, 1891.

[Mig83] M. Mignotte. Some useful bounds. In B. Buchberger, G. E. Collins, and R. Loos, editors, *Computer Algebra Symbolic and Algebraic Computation*, pages 259–263. Springer Verlag, 1983.

[Mio84] A. Miola. Algebraic approach to p-adic conversion of rational numbers. *Information Processing Letters*, 18:167–171, 1984.

[MNR81] N. M. Metropolis, G. Nicoletti, and G.-C. Rota. A new class of symmetric functions. *Adv. Math. Suppl. Studies*, 7B:563–575, 1981.

[Moe77] R. Moenck. On computing closed forms for summations. In *Proceedings of MACSYMA Users Conference*, pages 225–236, Berkley, California, 1977.

[MS95] D. E. G. Malm and T. N. Subramaniam. The summation of rational functions by an extended Gosper algorithm. *J. Symb. Comp.*, 1995.

[MW94] Y.-K. Man and F. J. Wright. Fast polynomial dispersion computation and its application to indefinite summation. In *ISAAC*, pages 175–180, Oxford England UK, 7 1994. ACM.

[NS82] W. Nichols and M.E. Sweedler. Hopf algebras and combinatorics. In *Umbral calculus and Hopf algebras*, volume 6 of *Contemporary Mathematics*, pages 49–84. Amer. Math. Soc., Providence, 1982.

[Pau93] P. Paule. Greatest factorial factorization and symbolic summation I. RISC-Linz Report Series 93-02, J. Kepler University, Linz, 1993.

[Pau95] P. Paule. Greatest factorial factorization and symbolic summation. *J. Symb. Comp.*, 20:235–268, 1995.

[Pet92] M. Petkovšek. Hypergeometric solutions of linear recurrences with polynomial coefficients. *J. Symb. Comp.*, 14:243–264, 1992.

[Pir92] R. Pirastu. Algorithmen zur Summation Rationaler Funktionen. Master's thesis, Univ. Erlangen-Nürnberg, 1992. (in german).

[Pir94] R. Pirastu. A note on the minimality problem in indefinite summation of rational functions. In J. Zeng, editor, *Actes du Séminaire Lotharingien de Combinatoire 31 session, Saint-Nabor, Ottrott, October 1993*, pages 95–101. Publications de l'I.R.M.A. 1994/021, 1994.

[Pir95] R. Pirastu. Algorithms for indefinite summation of rational functions in Maple. *the Maple Technical Newsletter*, 2(1):29–38, 1995.

[PSa] R. Pirastu and K. Siegl. Parallel computation and indefinite summation: A ||MAPLE|| application for the rational case. J. Symb. Comp., special issue on Symbolic Computation in Combinatorics Δ_1, Vol. 20, Nos. 5 and 6, 1995, pp. 603–616.

[PSb] R. Pirastu and V. Strehl. Rational summation and Gosper-Petkovšek representation. To appear in J. Symb. Comp., special issue on Symbolic Computation in Combinatorics Δ_1, Vol. 20, Nos. 5 and 6, 1995, p. 616–636.

[RKO73] G.-C. Rota, D. Kahaner, and A. Odlyzko. On the foundations of combinatorial theory VI. Finite operator calculus. *J. Math. Anal. Appl.*, 42:684–760, 1973.

[Rom84] S.M. Roman. *The umbral calculus.* Academic Press, 1984.

[Rom85] S.M. Roman. More on the umbral calculus, with emphasis on the q-umbral calculus. *J. Math. Anal. Appl.*, 107:222–254, 1985.

[RR78] S.M. Roman and G.-C. Rota. The umbral calculus. *Adv. Math.*, 27:95–188, 1978.

[Sie93] K. Siegl. Parallelizing algorithms for symbolic computation using ||MAPLE||. In *Fourth ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming, San Diego*, pages 179–186, 1993.

[SM92] T. N. Subramaniam and D. E. G. Malm. How to integrate rational functions. *The American Math. Monthly*, 99(8):762–772, Oct. 1992.

[Spr94] R. Sprugnoli. Riordan arrays and combinatorial sums. *Discrete Mathematics*, 132, 1994.

[Vil88a] G. Villard. Parallel general solution of rational linear systems using p-adic expansions. In Barton Cosnard and Vanneschi, editors, *Procs of the IFIP WG 10.3 Working Conference on Parallel Processing.* Elsevier Sc.P., 1988.

[Vil88b] G. Villard. *Symbolic computation and parallelism: solution of linear systems (in French).* PhD thesis, Institut National Polytechnique de Grenoble, December 1988.

Roberto Pirastu

Research Institute for Symbolic Computation
Johannes Kepler University, A-4040 Linz, Austria
tel: +43(7236)3231-82 fax: +43(7236)3231-30
⟨Roberto.Pirastu@risc.uni-linz.ac.at⟩

Personal

Born December 25, 1968, Cagliari, Italy. German and Italian citizen.

Education

1992-	PhD student, Research Institute for Symbolic Computation, Linz, Austria.
1992	Diplom in Informatik, Friedrich–Alexander Universität Erlangen–Nürnberg, Germany.
1991	Maîtrise de Mathématiques Discrètes, Université L. Pasteur, Strasbourg, France.
1987	Maturità Scientifica, Liceo A. Pacinotti, Cagliari, Italy.

Grants and Positions

1/95–1/96	Grant "Pro Scientia" for scientific literature from the Austrian Bishop Conference.
6/93–6/95	Grant "Human Capital and Mobility" from the Commission of the European Communities for research at RISC-Linz, contract Nr. ERBCHBICT930501.
9/90–9/91	Grant "Erasmus" from the Commission of the European Communities for the maîtrise at the Université L. Pasteur, Strasbourg, France.
7/89–7/90	Tutor, Department of Computer Science, Universität Erlangen–Nürnberg, Germany.

Publications

A note on the minimality problem in indefinite summation of rational functions. In J. Zeng, editor, *Actes du Séminaire Lotharingien de Combinatoire 31 session, Saint-Nabor, Ottrott, October 1993*, pages 95–101. Publications de l'I.R.M.A. 1994/021, 1994.

Exact solution of linear equation systems over rational number by parallel p-adic arithmetic (with C. Limongelli). In B. Buchberger and J. Volkert, editors, *Parallel Processing: CONPAR 94 - VAPP VI*, volume 854 of *LNCS*. Springer Verlag, 1994.

Algorithms for indefinite summation of rational functions in Maple. *the Maple Technical Newsletter*, 2(1):29–38, 1995.

Parallel computation and indefinite summation: A ||MAPLE|| application for the rational case (with K. Siegl). *J. Symb. Comp.*, **20**(5-6):603–616, 1995. Special issue on Symbolic Computation in Combinatorics Δ_1.

Rational summation and Gosper-Petkovšek representation (with V. Strehl). *J. Symb. Comp.*, **20**(5-6):617–636, 1995. Special issue on Symbolic Computation in Combinatorics Δ_1.

p-adic arithmetic and parallel symbolic computation: an implementation for solving linear systems over rationals (with C. Limongelli). *Computers and Artificial Intelligence*, 14(1):35–62, 1996.