Contributions to Symbolic Summation

Diplomarbeit
zur Erlangung des akademischen Grades
“Diplom-Ingenieur”
in der Studienrichtung
Technische Mathematik

verfaßt von
Markus Schorn

Angefertigt am Institut für Mathematik
der Technisch-Naturwissenschaftlichen Fakultät
der Johannes Kepler Universität Linz

bei
O.Univ.-Prof. Dr. Dr.h.c. Bruno Buchberger

Hagenberg, 1.12.1995

Contents

Chapter 1

Introduction

1.1 Symbolic Summation I

The answer to combinatorial enumerations often most easily can be expressed in terms of sums. For instance, counting the number of multiplication operations in the polynomial expression $a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0$, the direct answer is

$$\sum_{k=0}^{n} k.$$

Rewriting the polynomial expression, for instance, in Hornerform, and counting multiplications again results in the more explicit answer n. In order to compare complexities, a representation of the sum expression in closed form is desired.

In fact, everyone knows how to find a simpler expression for the sum: Since the average of the summand is given by $\frac{n}{2}$ and the summation range is of size $n+1$, we have, that the sum is equal to $\frac{n(n+1)}{2}$.

It is desirable to provide algorithms that look for *simpler representations* of sum expressions. Symbolic Summation indeed is a task of its own beauty and will be studied in the following.

Considering the task of Symbolic Summation, the next example is a little more exiting:

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

It can be solved by the observation that $\frac{1}{(k+1)(k+2)} = \frac{1}{k+1} - \frac{1}{k+2}$, which yields

$$\sum_{k=0}^{n} \frac{1}{(k+1)(k+2)} = (1-\frac{1}{2}) + (\frac{1}{2}-\frac{1}{3}) + \cdots + (\frac{1}{n+1}-\frac{1}{n+2}) = 1-\frac{1}{n+2}.$$

It turns out that we can also solve the first example by this telescoping trick. (Observe that $k = \frac{k(k+1)}{2} - \frac{(k-1)k}{2}$.) This is not by accident! Whenever we have $\sum_{k=0}^{n} f(k) = g(n)$, we find that $g(n+1) - g(n) = f(n+1)$, or stated with use of difference and shift operator, $\Delta g = Ef$.

Sum expressions of the type $\sum_{k=0}^{n} f(k)$ are called indefinite sums. In order to find a sum quantifier free representation for the indefinite sum over f, you have to construct g, such that $\Delta g = f$, and g free of sum quantifiers. g is called the anti-difference of f. Then you have $\sum_{k=0}^{n} f(k) = g(n+1) - g(0)$.

Let us consider the polynomials

$$(x)_i := \prod_{j=0}^{i-1} (x-j).$$

If $\mathbb{F}$ is a field, then the polynomials $\{(x)_0, (x)_1, \ldots\}$ are a basis of the vector space $\mathbb{F}[x]$ over $\mathbb{F}$, i.e., any $p \in \mathbb{F}[x]$ can be represented as a linear combination of the polynomials $(x)_i$. Now we find that

$$\Delta(x)_i = [(x+1) - (x-i+1)] \prod_{j=0}^{i-2} (x-j) = i\,(x)_{i-1}, \tag{1.1}$$

and therefore the anti-difference of a polynomial is constructible: Compute the representation of the given polynomial p in terms of the basis above and use equation (1.1) to replace the polynomials of the basis by linear expressions in the Δ operator! Since the operator Δ is linear too, you find that your polynomial is represented as the result of a Δ operation. The argument of this operation is the anti-difference of p. This algorithm was already found by J. Bernoulli in the 17^{th} century.

An algorithmic approach to find an anti-difference of a rational function, in case it exists, was a byproduct of the work of Moenck in 1977 who found an algorithm to compute a canonical form of indefinite sums over rational functions. Peter Paule discovered that Moenck's algorithm is not working in every case and gave a correct solution in [Pau93]. The problem was solved by a different approach before by S. Abramov.

In [Gos78] Gosper describes an algorithm for constructing the anti-difference for hypergeometric sequences. The algorithm is very simple and does not reveal the theory behind. You can use it and be fascinated by the fact that it works.

In various papers as for instance [Zei90], D. Zeilberger introduces an algorithm based on Gosper's algorithm for finding polynomial recurrences for definite hypergeometric sum expressions. His idea, however, has a more general background and together with H. Wilf he was able do develop a method for finding recurrences for hypergeometric multi-sums and integrals, see [WZ92].

Due to the work of M. Petkovšek, one can transform such recurrences back to hypergeometric expressions, if they admit a hypergeometric sequence solution, see [Pet92].

Recurrence relations are a convenient representation, for instance, for proving. That is why it may be considered to be a more simple representation for sum-expressions, anyhow.

1.2 Goals of the thesis

Together with Peter Paule I wrote an implementation of Zeilberger's algorithm. Unlike other implementations it can handle nonstandard boundary conditions. Since no precise theory for the treatment of the summation bounds is available, we developed one, which is part of my diploma thesis now. The objective is to provide a profound theoretical background to the already well working implementation; see [PS95].

In chapter 2 we present the notation used; In chapter 3 we formalize the problem of finding *simpler representations* for sum expressions. Chapter 4 is more or less independent; there we set up a calculus to compute with hypergeometric terms. All the knowledge of the chapters 3 and 4 is used in chapter 5 to discuss how Gosper's algorithm computes *closed forms* for indefinite hypergeometric sums, if they exist. Our insight is backed up by two interesting examples. In chapter 6 we prove Zeilberger's algorithm for definite sums with nonstandard boundary conditions, again providing a few examples, that are off the road. In chapter 7 we proudly introduce the new algorithm MS to simplify hypergeometric sum expressions. It is a very fast tool for proving double-sum identities.

Chapter 2

Notation

In literature for some notions there are different conventions. Here we make clear how we use notation.

2.1 Constants

Constant	Description
$\mathbb{N} := \{0, 1, \ldots, \}$	
$\mathbb{Z}, \mathbb{C}$	Integers, complex numbers.
$\mathbb{Z}^- := \{-1, -2, \ldots\}$	
$\mathbb{Z}_0^- := \{0, -1, \ldots\}$	
$x, x_1, \ldots, y, y_1, \ldots$	To build up polynomial rings and rational function fields we need transcendental extensions of simpler domains. These are extended by an algebraic independent indeterminate. To make this construction convenient we use $x, x_1, \ldots, y, y_1, \ldots$ as indeterminates, which are always considered to be algebraic independent.
$\mu \in \mathbb{N}$	largest index $i \in \mathbb{N}$ used for x_i.
$\nu \in \mathbb{N}$	see page 17
$\vec{x} := (x_1, \ldots, x_\mu)$	
$\vec{x_2} := (x_2, \ldots, x_\mu)$	
$\vec{x_{\nu+1}} := (x_{\nu+1}, \ldots, x_\mu)$	

Constant	Description
φ	Denotes the Pochhammer function defined as follows: Let p be some field element, $z \in \mathbb{Z}$ and let $\prod_{j\in\{\}} p := 1$; then $$\varphi(p,z) := \Big(\prod_{j=0}^{z-1}(p+j)\Big)\Big(\prod_{j=1}^{-z}(p-j)\Big)^{-1}.$$

2.2 Typed Variables

I use typed variables. This is a handy way to get rid of lot of assertions like "let $n \in \mathbb{N}$". The problem with this technique is, that it is usually hard to find out whether a variable is typed or not. I cannot type all variables right at the beginning, because some notions needed for a special type are not defined at this point. Nevertheless I list all typed variables here, so the reader can always refer to this list. If the typing done below uses notions not yet defined, then the typing is repeated in a later chapter, after introducing the notion.

Typed Variables 1 *(Overview)*

Throughout this thesis assert:

$\mathcal{A}$ is a set,
R is a ring of characteristic 0,
$\mathbb{F}$ is a field of characteristic 0,
$i, j, m, n \in \mathbb{N}$,
$z \in \mathbb{Z}$,
$u, w \in \mathbb{C}$,
$\vec{u} \in \mathbb{C}^\mu$,
$\vec{w} \in \mathbb{C}^{\mu-1}$,
$\varrho, \varsigma \in \mathbb{C}(\vec{x})$,
$f, g, h \in \mathrm{HT}$,
$f^-, g^-, h^- \in \mathrm{HT}^-$,
$f^+, g^+, h^+ \in \mathrm{HT}^+$.

For the chapters 3 and 4 assert:

$r, r_1, \ldots, s, s_1, \ldots \in \mathrm{IL}$.

2.3 Notions

Notion	Description
$p[c]$	If $p \in R[x], c \in R$ then by $p[c]$ we denote the ring element found by substituting c for x in p.
$p(x+1)$	If $p \in R[x]$ then by $p(x+1)$ we denote the composition of the polynomial p and the polynomial $x+1$. If we have a multivariate polynomial $q \in R[x,y]$ then $q(x+1)$ is ambiguous. Therefore we prefer writing $q(x,y) \in R[x,y]$ to assert that x is the first variable, in order to distinguish between $q(x,x+1)$ and $q(x+1,y)$. There is no difference between $p[c]$ and $p(c)$.
$p \circ (x+1)$	Denotes the same as above. For instance $(x+1) \circ y = y+1$ cannot be denoted by $(x+1)(y)$.
$\mathbb{F}^*$	Multiplicative group of the field $\mathbb{F}$.
$\langle \mathcal{A} \rangle$	Abelian group generated by $\mathcal{A}$.
$\mathcal{A}^{\mathbb{N}}$	$\mathcal{A}$-sequences.
$(u, \vec{w})$	Is an element of $\mathbb{C}^{\mu}$. It is not a pair, it is a tuple of length μ.

Chapter 3

Sequences

3.1 Basic facts about sequences

In Symbolic Summation we typically consider a sum with a symbolic upper bound and therefore the sum over any finite starting section of a sequence of values. A *closed form* to a summation problem actually represents a sequence of sums. In the first example, for instance, we computed $\sum_{k=0}^{0} k$, $\sum_{k=0}^{1} k$, $\sum_{k=0}^{2} k$, etc. The problem was formulated using the sequence $(k)_{k\geq 0}$ and also the answer is a sequence: $\left(\frac{k(k+1)}{2}\right)_{k\geq 0}$. Both sequences are represented by terms.

As a sequence is just a special type of function, we find that many terms represent sequences. For instance, any polynomial $p \in R[x]$ defines the R-sequence, $(p[k])_{k\geq 0}$. Consider a polynomial $q(x,y) \in R[x,y]$. It defines the $R[y]$-sequence, $(q(k,y))_{k\geq 0}$ and the $R[x]$-sequence, $(q(x,k))_{k\geq 0}$. But also $(\sin(k))_{k\geq 0}$ is a sequence.

3.1.1 Operations on sequences

For a fixed ring R, we define the following operations on R-sequences:

Definition 1 *(Operations on R-sequences)*

Let $(a_i)_{i\geq 0}, (b_i)_{i\geq 0} \in R^{\mathbb{N}}$, then we define:

1. Addition

$$(a_i)_{i\geq 0} + (b_i)_{i\geq 0} := (a_i + b_i)_{i\geq 0},$$

$$-(a_i)_{i\geq 0} := (-a_i)_{i\geq 0}.$$

2. Hadamard product

$$(a_i)_{i\geq 0} \cdot (b_i)_{i\geq 0} := (a_i \cdot b_i)_{i\geq 0}.$$

3. First element sequence

$$L(a_i)_{i\geq 0} := (a_0)_{i\geq 0}.$$

4. Shift operator

$$E(a_i)_{i\geq 0} := (a_{i+1})_{i\geq 0}.$$

5. Difference operator

$$\Delta(a_i)_{i\geq 0} := E(a_i)_{i\geq 0} - (a_i)_{i\geq 0}.$$

6. Sum operator

$$\sum(a_i)_{i\geq 0} := \Big(\sum_{k=0}^{i} a_k\Big)_{i\geq 0}.$$

Note, that for any R also $(R^{\mathbb{N}}, +, \cdot)$ is a ring. Following the approach of [PN95] we can more formally establish what we have found out in the introduction. In any ring of R-sequences, we have a close relationship between a sum and the anti-difference of a sequence.

Theorem 1 *(Sum quantifier elimination)*

If $a, b \in R^{\mathbb{N}}$ then

$$\sum a = b - Lb + La \;\Leftrightarrow\; Ea = \Delta b,$$
$$\sum a = Eb - Lb \;\Leftrightarrow\; a = \Delta b.$$

Proof

1. "$\Rightarrow$": Assuming $\sum a = b - Lb + La$, we find $Ea = \Delta b$ by applying the Δ operator.

   "$\Leftarrow$": Assuming $(Ea = \Delta b)$, we have

   $$E\sum a = La + \sum Ea = La + \sum \Delta b = Eb - Lb + La = E(b - Lb + La),$$
   $$L\sum a = La = L(b - Lb + La).$$

Combining this we have $\sum a = b - Lb + La$.

2. "⇒": Assuming $\sum a = Eb - Lb$, we find

$$Ea = E\Delta b, \text{ by applying the } \Delta\text{-operator;}$$
$$La = LEb - Lb, \text{ by applying the } L\text{-operator;}$$

We combine this to $a = \Delta b$.

"⇐": Assuming $(a = \Delta b)$, by direct application of the $\sum$-operator we have $\sum a = b - Lb$.

□

Note, that if $\sum a = b$ we immediately derive $b = b - Lb + La$ and also $b = Ec - Lc$, with $Lc = 0$ and $Ec = b$. So again, given $\sum a = b$, we can apply the theorem above to find the anti-differences $Ea = \Delta b$ and $a = \Delta c$. In the world of sequences looking for anti-differences is just the same thing as looking for sums!

In the following we will have to restrict the sequences in use to a subset that can be represented by a class of terms. In that case version 1 of the theorem above is preferable since there the related anti-difference (b) can be expressed with the same term as the sum. Using version 2, we do not know if the anti-difference (c) is representable by the class of terms in question, even under the assumption that the sum is.

The two versions switch sides, if the class of terms allows to define an inverse shift-operator. Then the anti-difference c is given by $L^{-1}b$. In that case the formula for the sum in version 2 ($Eb - Lb$) is a little bit simpler and should be prefered.

3.1.2 Symbolic Summation II

In the world of computers we do not know any sequences. Rather than that, we have terms (or data structures) that implicitly define sequences. It is a task of this thesis to give explicit definitions of the sequences represented by a special class of terms.

The input of a summation algorithm will be some term f defining a sequence $(a_i)_{i\geq 0}$. The problem is to find a term g, free of sum quantifiers defining the sequence $(b_i)_{i\geq 0}$, such that $\sum (a_i)_{i\geq 0} = (b_i)_{i\geq 0}$.

Definition 2 *(Sequence representations)*

If X is a set of terms, then the pair (X, ϕ) is an $\mathcal{A}$-sequence representation iff $\phi : X \to \mathcal{A}^{\mathbb{N}}$.

Definition 3 *(Indefinite summation problem)*

Let (Y, ϕ) be a R-sequence representation and $X \subseteq Y$, then the indefinite summation problem of (Y, ϕ) with input domain X is defined by

$$\begin{array}{ll} \text{Input}: & f \in X \\ \text{Output}: & g \in Y, \text{ such that } \sum \phi(f) = \phi(g) \\ & \text{or failure, if no such } g \text{ exists.} \end{array}$$

By theorem 1 we know that the indefinite summation problem of $(R^{\mathbb{N}}, \text{ id})$ with input domain $R^{\mathbb{N}}$ is equivalent to the problem of finding anti-differences of sequences.

Definition 4 *(Anti-difference problem)*

Let (Y, ϕ) be a R-sequence representation and $X \subseteq Y$, then the anti-difference problem is given by:

$$\begin{array}{ll} \text{Input}: & f \in X \\ \text{Output}: & g \in Y, \text{ such that } \phi(f) = \Delta\phi(g) \\ & \text{or failure, if no such } g \text{ exists.} \end{array}$$

In order to be able to solve the related problem of finding the anti-difference, it is of advantage to equip the set of terms with the operations of shifting ($\mathtt{E}$), adding $(+, -)$ and projection ($\mathtt{L}$). These operations must be such, that they can be performed by a computer. Furthermore the operations on the terms have to go along with the corresponding operations on the represented sequences.

For a set of terms with such operations we can relate the problem of summation with the problem of finding anti-differences once again:

Corollary 2 *(Anti-difference vs. indefinite summation)*

Let (Y, ϕ) be a R-sequence representation; $\mathtt{E}$, $\mathtt{L}$, $+, -$ are operators on Y as above, such that Y is closed under these operators and ϕ is an $(E, L, +, -)$-homomorphism. If $X \subseteq Y$ is closed under $\mathtt{E}$ then for $f \in X$ and $g \in Y$, we have:

$$\phi(\mathtt{E}f) = \phi(\mathtt{E}g - g) \Leftrightarrow \sum \phi(f) = \phi(g - \mathtt{L}g + \mathtt{L}f).$$

By this corollary we know that for a set of R-sequence representations containing the operators $\mathtt{E}, \mathtt{L}, +$ and $-$, the problem of finding the anti-difference is equivalent to the problem of finding the solution of the (backwards shifted) indefinite summation problem. An algorithm solving the indefinite summation problem for a special class of terms does not have to deal with sum quantifiers!

We give an informal summary of what we have to establish in order to solve the indefinite summation problem for a set of sequence defining terms.

Concept 1 *(Finding closed forms for sum-expressions)*

We need:

1. A R-sequence interpretation (Y, ϕ), and an input domain $X \subseteq Y$.
2. The operations $\mathtt{E}, \mathtt{L}, +$ and $-$, such that
   (a) Y is closed under $\mathtt{E}, \mathtt{L}, +$ and $-$ and X is closed under $\mathtt{E}$,
   (b) ϕ is an $(E, L, +, -)$-homomorphism.
3. An algorithm that solves the anti-difference problem of Y with input domain X.

To illustrate this procedure, we prove the summation algorithm for polynomial sequences: The terms are given by $Y = X = R[x]$, the interpretation of the terms as R-sequences is defined by

$$\forall p \in R[x]\colon\ \phi(p) := (p[i])_{i \geq 0}.$$

The operations are given by $\forall p \in R[x]\colon\ \mathtt{L}p := p[0]$, $\mathtt{E}p(x) := p(x+1)$, $-p := -p$ and $\forall p, q \in R[x]\colon\ p+q := p + q$.

It is an easy exercise to prove that ϕ is an $(E, L, +, -)$-homomorphism $\phi : R[x] \to R^{\mathbb{N}}$. As we showed in the beginning we have an algorithm that solves the anti-difference problem for polynomials. If $q(x)$ is the anti-difference of $p(x) \in R[x]$, then the result for the indefinite summation problem with input $p(x)$ is given by $(\mathtt{E} - \mathtt{L})q(x) = q(x+1) - q[0]$.

3.1.3 Sequences defined by recurrences

We already discussed that in order to work with sequences we have to be able to represent sequences. Any function defines a sequence, for instance $(\sin(i))_{i \geq 0}$ is one. Today algorithmic Symbolic Summation can be done only for a small class of such defined sequences.

The most celebrated algorithms in the field of Symbolic Summation are Gosper's and Zeilberger's algorithm for summing over hypergeometric sequences. Actually I do not like the notion of hypergeometric because scientists around the world do not agree what the definition of hypergeometric is. I do not want to contribute to more confusion, but use the notion of "order one sequences" instead. For headlines and names "hypergeometric" is just fine.

We deal with sequences that satisfy polynomial recurrences.

Definition 5 *(P-finite sequences)*

Let $a = (a_i)_{i\geq 0} \in R^{\mathbb{N}}$, then a is p-finite of order n iff

$$\begin{aligned}&\exists p_0, \dots p_n \in R[x], p_n \neq 0\colon\ \forall i\colon\\ &\quad p_0[i]a_i + p_1[i]a_{i+1} + \cdots + p_n[i]a_{i+n} = 0.\end{aligned}$$

The definition above suggests to ask for the minimal recurrence relation a sequence satisfies.

Definition 6 *(Order of a p-finite sequence)*

Let $a \in R^{\mathbb{N}}$, then

$$\mathrm{ord}(a) := \min\{n \mid a \text{ is p-finite of order } n\}.$$

As an example we give the sequence $(k!)_{k\geq 0}$. It is p-finite of order one since

$$\forall i\colon\ (i+1)\, i! - (i+1)! = 0.$$

It is the class of p-finite sequences of order one, we are interested in. Assume we have an R-sequence, $a = (a_i)_{i\geq 0}$, of order one, then it satisfies a recurrence relation of order one, i.e., for some $p, q \in R[x], q \neq 0$, we have

$$p[i]a_i - q[i]a_{i+1}.$$

If we assume that $\forall i\colon\ q[i] \neq 0$ then we have

$$a_i = a_0 \prod_{j=0}^{i-1} \frac{p[j]}{q[j]}.$$

We can do even more under the additional assumption, that p and q can be represented as product of linear factors in x. This is possible, for instance, if R is a rational function field over $\mathbb{C}$. So assume that $p = c\ \prod_{l=1}^{n}(x + c_l)$ and $q = d\ \prod_{l=1}^{m}(x - d_l)$, then we have

$$a_i = a_0 \Big(\frac{c}{d}\Big)^i \frac{\prod_{l=1}^{n} \varphi(c_l, i)}{\prod_{l=1}^{m} \varphi(d_l, i)}.$$

So we have a *representation* for many p-finite sequences of order one. This was the motivation for mathematicians to define terms, sequences or series, similar to the sequence above, to be hypergeometric. Note, that this representation – using rising factorials – can be given in terms of factorials or falling factorials, also.

Summarizing we found a set of terms defining many sequences of order one. These terms can serve as input class for a summation algorithm.

3.2 Hypergeometric terms representing sequences of order one

3.2.1 The definition of hypergeometric terms

Motivated by the fact that a lot of p-finite sequences of order one can be represented as a product of

- Rational functions
- Factorials
- Rising factorials
- Falling factorials
- Binomials
- Exponential expressions

we will use these ingredients to define the set of hypergeometric terms. A factorial or a binomial with an arbitrary rational function as an argument, however, is not necessarily of order one. For instance, $(\frac{1}{2}y)!/(\frac{1}{2}(y+1))!$ is not rational. Using integer linear rational functions, we are on the safe side and find that factorials and binomials define sequences of order one. Note, that for instance

$$\frac{1}{4\pi}\,(2\sqrt{2})^k\,\Gamma\Big(\frac{k}{4}\Big)\Gamma\Big(\frac{k+2}{4}\Big)\Gamma\Big(\frac{k+1}{2}\Big)$$

defines a p-finite sequence of order one, but cannot directly be rewritten in terms of integer linear factorials. (Hint: $\Gamma(x)\Gamma(x+\frac{1}{2}) = \frac{\sqrt{x}}{2^{2x-1}}\Gamma(2x)$, therefore the expression above reduces to $\Gamma(k)$.)

Here we fix the number of variables we are interested in to be μ. These variables are named $x_1, \ldots, x_\mu$. A term may be read as a sequence in several variables, for instance, $x + y$ defines the sequence $y, y+1, \ldots$ or the sequence $x, x+1, \ldots$. We fix $\nu \in \mathbb{N}, \nu < \mu$ such that $x_1, \ldots, x_\nu$ can be used to translate a term into a sequence of order one. We fix the following constants:

$$\vec{x} := (x_1, \ldots, x_\mu),$$
$$\vec{x_2} := (x_2, \ldots, x_\mu),$$
$$\vec{x_{\nu+1}} := (x_{\nu+1}, \ldots, x_\mu).$$

Typed Variables 2 *(Rational functions)*

$$\varrho, \varsigma \in \mathbb{C}(\vec{x}).$$

Definition 7 *(Integer linear functions)*

1. ϱ is integer linear in x_i iff

$$\exists z, \varsigma \colon \ \varsigma \text{ free of } x_i, \ \varrho = z x_i + \varsigma.$$

2. $\mathrm{IL} := \{\varrho \mid \varrho \text{ is integer linear in } x_1, \ldots, x_\nu\}$.

Typed Variables 3 *(Integer linear functions)*

$$r, r_1, \ldots, s, s_1, \ldots \in \mathrm{IL}.$$

Out of these rational functions we generate the multiplicative abelian group of hypergeometric terms by introducing new function symbols. Right now, we define only terms without meaning, afterwards we will map the terms to sequences. For instance, in the following the exclamation mark is just a symbol without any functional impact.

Definition 8 *(Hypergeometric terms)*

- Factorials

$$\mathrm{Fac} := \langle \{ r! \mid r \notin \mathbb{Z}^- \} \rangle,$$
$$\mathrm{FFac} := \langle \{ (r)_{\underline{s}} \mid r, (r-s) \notin \mathbb{Z}^- \} \rangle,$$
$$\mathrm{RFac} := \langle \{ (r)_{\overline{s}} \mid r+s, r \notin \mathbb{Z}_0^- \} \rangle.$$

- Binomials

$$\mathrm{Bin} := \left\langle \left\{ \binom{r}{s} \middle| r, s, r-s \notin \mathbb{Z}^- \right\} \right\rangle.$$

- Exponential expressions

$$\mathrm{Exp} := \{ r_1^{x_1} \, r_2^{x_2} \cdots r_m^{x_\nu} \mid r_1, \ldots, r_\nu \in \mathbb{C}(\vec{x_{\nu+1}}) \}.$$

- Multiplicative abelian group of hypergeometric terms

$$\mathrm{HT}^- := \mathrm{Fac} \times \mathrm{FFac} \times \mathrm{RFac} \times \mathrm{Bin} \times \mathrm{Exp},$$
$$\mathrm{HT} := \mathbb{C}(\vec{x})^* \times \mathrm{HT}^-,$$

In addition, we define

$$\mathrm{HT}_0 := \mathrm{HT} \cup \{0\}.$$

Typed Variables 4 *(Terms)*

$$f, g, h \in \mathrm{HT}$$
$$f^-, g^-, h^- \in \mathrm{HT}^-.$$

We had to exclude a couple of constructions, such as $(-1)!$ and $(-2)_{\underline{k}}$. There are three different types of terms that do not fit into the definition of hypergeometric terms. The easiest one is the type of expression that does not make sense, since the classic evaluation of the construction is not defined (compare definition 9). So do not use $(-1)!, (2)_{-3}$ or $\binom{-1}{0.5}$. The second type is an expression, which evaluates to 0 for any choice of values. This means it can be viewed as zero on the algebraic level, too. Write 0 instead of $\frac{1}{(-1)!}$ or $(-1)_2$. Calculating with $\frac{1}{(a)!}$ and then plugging in -1 for a will be no problem at all. To the third type belong expressions that can be rewritten, for example $(-3)_{\underline{2}} = 6$ or $(-2)_{\underline{a}} = (-1)^a (2)_{\overline{a}}$.

We give a complete table of rules that show how to rewrite an expression excluded in the definition above to a hypergeometric term. Note, that such a rewrite rule may enlarge the domain of evaluation, again compare definition 9, only the last rule leaves you with an expression evaluable over a smaller domain than the original. So we did find one type of expression, namely $\binom{s+z}{s}$, that cannot fully be expressed in our model.

If $z \in \mathbb{Z}^-$ then use the following rewrite rules:

$$
\begin{array}{rcl|rcl}
z! & \rightarrow & \text{is meaningless} & \frac{1}{z!} & \rightarrow & 0 \\
(z)_{\underline{s}} & \rightarrow & (-1)^s(-z)_{\overline{s}} & & & \\
(s)_{\underline{s-z}} & \rightarrow & (-1)^{s-z}(-s)_{\overline{s-z}} & & & \\
(z+1)_{\overline{s}} & \rightarrow & (-1)^s(-z-1)_{\underline{s}} & & & \\
(s)_{\overline{z-s+1}} & \rightarrow & (-1)^{z-s+1}(-s)_{\underline{z-s+1}} & & & \\
\binom{z}{s} & \rightarrow & (-1)^s\binom{s-z-1}{s} & & & \\
\binom{s}{z} & \rightarrow & 0 & \frac{1}{\binom{s}{z}} & \rightarrow & \text{is meaningless} \\
\binom{s+z}{s} & \rightarrow & (-1)^s\binom{-z-1}{s}. & & &
\end{array}
$$

A hypergeometric term is defined as a tuple of subexpressions. We do not always want to write $(x, (1, 1, 1, 1, 1))$ for the rational function x as an element of HT. Rather than that we express the elements of HT by the sloppily introduced canonical form,

$$
\varrho \prod_{i=0}^{n_1} (r_{i,1})!^{j_{i,1}} \prod_{i=0}^{n_2} (r_{i,2})_{\underline{s_{i,2}}}{}^{j_{i,2}} \prod_{i=0}^{n_3} (r_{i,3})_{\overline{s_{i,3}}}{}^{j_{i,3}} \prod_{i=0}^{n_4} \binom{r_{i,4}}{s_{i,4}}^{j_{i,4}} r_1^{x_1} r_2^{x_2} \cdots r_m^{x_m}.
$$

This is a canonical form because each subexpression appears in its obvious canonical form. Translating this canonical form back to the tuple notation we have to be careful. Multiplication of elements of HT will always be written using a dot, whereas the canonical form does not contain any dots. Thus with $\varrho\varsigma$ we denote $(\varrho\varsigma, (1, 1, 1, 1, 1))$ whereas $\varrho \cdot \varsigma$ must be translated to $(\varrho, (1, 1, 1, 1, 1))(\varsigma, (1, 1, 1, 1, 1))$. Since both terms are equal in HT the different meaning of $\varrho\varsigma$ and $\varrho \cdot \varsigma$ is usually of little importance.

3.2.2 Standard interpretation of hypergeometric terms

In order to be able to speak of a representations for sequences we need some functional interpretation of hypergeometric terms. Factorials, binomials and rational functions have a precise meaning, already. We acknowledge this fact by giving a definition of the evaluation of hypergeometric terms extending the classic meaning of the terms over the complex numbers. I must point out that this evaluation cannot be used to set up the sequence interpretation, as we see in section 4.1.4. This is why we define an alternative evaluation model in section 4.2. The following definition will give us an orientation for how evaluation has to look like.

Definition 9 *(Classic evaluation of hypergeometric terms)*

We define the c-evaluation of a hypergeometric term at $\vec{u}$ to be the product of the c-evaluation of its sub-terms. If any sub-term cannot be evaluated, then the overall evaluation is undefined, as well.

The evaluation of the sub-terms is given by the following, where the symbol ∞ is used to indicate that an evaluation is undefined:

$$s![\vec{u}] \quad := \quad \begin{cases} \Gamma(s[\vec{u}]+1), & \text{if } s[\vec{u}] \notin \mathbb{Z}^-, \\ \infty, & \text{if } s[\vec{u}] \in \mathbb{Z}^-; \end{cases}$$

$$\begin{aligned} (r)_{\underline{s}}[\vec{u}] \quad &:= \quad \lim_{r \to r[\vec{u}]} \frac{\Gamma(r+1)}{\Gamma(r - s[\vec{u}] + 1)} = \\ &= \begin{cases} \frac{\Gamma(r[\vec{u}]+1)}{\Gamma((r-s)[\vec{u}]+1)}, & \text{if } r[\vec{u}] \notin \mathbb{Z}^- \wedge (r-s)[\vec{u}] \notin \mathbb{Z}^-, \\ 0, & \text{if } r[\vec{u}] \notin \mathbb{Z}^- \wedge (r-s)[\vec{u}] \in \mathbb{Z}^-, \\ \infty, & \text{if } r[\vec{u}] \in \mathbb{Z}^- \wedge (r-s)[\vec{u}] \notin \mathbb{Z}^-, \\ (-1)^{s[\vec{u}]} \frac{\Gamma((s-r)[\vec{u}])}{\Gamma(-r[\vec{u}])}, & \text{if } r[\vec{u}] \in \mathbb{Z}^- \wedge (r-s)[\vec{u}] \in \mathbb{Z}^-; \end{cases} \end{aligned}$$

$$\begin{aligned} (r)_{\overline{s}}[\vec{u}] \quad &:= \quad \lim_{r \to r[\vec{u}]} \frac{\Gamma(r + s[\vec{u}])}{\Gamma(r)} = \\ &= \begin{cases} \frac{\Gamma((r+s)[\vec{u}])}{\Gamma(r[\vec{u}])}, & \text{if } (r+s)[\vec{u}] \notin \mathbb{Z}_0^- \wedge r[\vec{u}] \notin \mathbb{Z}_0^-, \\ 0, & \text{if } (r+s)[\vec{u}] \notin \mathbb{Z}_0^- \wedge r[\vec{u}] \in \mathbb{Z}_0^-, \\ \infty, & \text{if } (r+s)[\vec{u}] \in \mathbb{Z}_0^- \wedge r[\vec{u}] \notin \mathbb{Z}_0^-, \\ (-1)^{s[\vec{u}]} \frac{\Gamma(-r[\vec{u}]+1)}{\Gamma(1-(r+s)[\vec{u}])}, & \text{if } (r+s)[\vec{u}] \in \mathbb{Z}_0^- \wedge r[\vec{u}] \in \mathbb{Z}_0^-; \end{cases} \end{aligned}$$

$$\binom{r}{s}[\vec{u}] \quad := \quad \begin{cases} \frac{(r)_{\underline{s}}[\vec{u}]}{s![\vec{u}]}, & \text{if } s[\vec{u}] \in \mathbb{N} \vee r[\vec{u}] \notin \mathbb{Z}^-, \\ 0, & \text{if } s[\vec{u}] \in \mathbb{Z}^-, \\ \infty, & \text{if } s[\vec{u}] \notin \mathbb{Z} \wedge r[\vec{u}] \in \mathbb{Z}^-. \end{cases}$$

Further we give the domain of classic evaluation by

$$D_c(f) := \{\vec{u} \mid f[\vec{u}] \neq \infty\}.$$

It is up to the reader to verify that this definition, restricted to integers or some other domain meets his view of evaluating factorials and binomials.

Did you notice, that we interpreted hypergeometric terms as functions and not as sequences? We did so, because a term does not give an indication in which variable it should define a sequence. So it is easier to interpret terms as functions with several arguments. After fixing an index variable, we get a sequence of functions with one argument less.

3.2.3 $\mathbb{F}$-sequence interpretation of hypergeometric terms

In the traditional proofs of Gosper's and Zeilberger's algorithm one always assumes that a hypergeometric term represents a $\mathbb{F}$-sequence. Following this approach in [Koo92] you find a concise proof of Gosper's and Zeilberger's algorithm.

There is a problem with this approach. You have to impose unnecessary restrictions to the problems solved by Gosper's algorithm in order to make your proof work: Let $(a_i)_{i\geq 0}$ be the sequence defined (via c-evaluation) by a hypergeometric term f, and $(b_i)_{i\geq 0}$ the sequence interpretation of the anti-difference of f, then the proof of Gosper's algorithm works under the following restrictions:

- $\forall i\colon\ a_i \in \mathbb{F}$,
- $\forall i\colon\ a_i \neq 0$,
- $\forall i\colon\ b_i \neq 0$.

First, it is not discussed what field can be used to give a $\mathbb{F}$-sequence interpretation for an arbitrary hypergeometric term, therefore it is not clear for which terms the proof applies. (It is clear for which sequences though!) For instance, at the first glance it looks as if factorials, would be excluded from being part of the summand. $(\frac{1}{(a+k)!})_{k\geq 0}$ is a sequence of terms. We may interpret this as $(\frac{1}{\Gamma(a+k+1)})_{k\geq 0}$, a sequence of functions. To what kind of field do the sequence elements belong? Luckily, factorials can be expressed in terms of rising factorials. The transformation $(a+k)! = (a)!(a+1)_k$ and moving the k-free part out of the summation does the job. Instead of $\sum_{k=0}^{n} \frac{1}{(a+k)!}$ we are faced with $\frac{1}{a!}\sum_{k=0}^{n} \frac{1}{(a+1)_k}$, a sequence of rational functions. Therefore the summation problem can be interpreted via $\mathbb{C}(a)$-sequences. Notice that in opposition to the former representation the latter cannot be evaluated if a evaluates to a negative integer. So there are tricks to give $\mathbb{F}$-sequence interpretations to problems. Again they work only as long, as the sequences do not contain zeros.

Second, polynomials with an integer root in the summation variable are not allowed to be part of the input. This can be the case quite often. Equation (3.1) may serve as an example.

Third, it is important to see that we cannot conclude that a summation problem rejected by Gosper's algorithm has no closed form within the set of *hypergeometric terms*. Although the algorithm did not produce a solution to a problem there might be one, namely an anti-difference containing a polynomial factor with an integer root in the summation variable, as this occurs in equation (3.2), for instance.

Here are two examples where Gosper's algorithm is better than we can expect:

$$\sum_{k=0}^{a}(k-4) = \frac{(a-8)(a+1)}{2}, \tag{3.1}$$

$$\sum_{k=0}^{n} \frac{(1-2\,k^2+k\,n)}{1+k}\binom{n}{k} = 1. \tag{3.2}$$

In the first example, there is a zero among the summands. In the second one, the anti-difference contains a zero. Observe, that the following holds:

$$\Delta\left((k-1)\binom{n}{k}\right)_{k\geq 0} = \left(\frac{(1-2\,k^2+k\,n)}{1+k}\binom{n}{k}\right)_{k\geq 0}.$$

3.2.4 Symbolic Summation III

Let us interpret hypergeometric terms as sequences of functions, for instance, in the example above: $(\frac{1}{\Gamma(a+k+1)})_{k\geq 0}$. One indeterminate (namely k) is spent to index the sequence, thus the elements of the sequence are functions with $\mu-1$ arguments. Therefore, their domain is a subset of $\mathbb{C}^{\mu-1}$. If we restrict the domain of all sequence-elements to the same subset of $\mathbb{C}^{\mu-1}$, then all these functions belong to a function ring. In order to give a R-sequence interpretation to a hypergeometric term we have to find a common domain for the sequence-element functions. We specify this domain as large as possible, i.e. we take all tuples of values, to which all of the sequence-element functions can be applied.

Definition 10 *(Classic interpretation domain)*

$$D_C(f) := \{\vec{w} \mid \mathbb{N}\times\{\vec{w}\} \subseteq D_c(f)\}.$$

We give an example:

$$\begin{aligned} D_c\Big(\frac{x_1!}{x_1-x_2}\Big) &= \{(u,v)\in\mathbb{C}^2 \mid u\notin\mathbb{Z}^-, u-v\neq 0\},\\ D_C\Big(\frac{x_1!}{x_1-x_2}\Big) &= \mathbb{C}\setminus\mathbb{N}. \end{aligned}$$

Now we can interpret single terms as sequence of functions. In the example $f = \frac{x_1!}{x_1-x_2}$ this would be the sequence $(\frac{\Gamma(k+1)}{k-x})_{k\geq 0}$, where the domain of the sequence-element functions is given by $D_C(f) = \mathbb{C}\setminus\mathbb{N}$.

We can use the same domain to give an interpretation for other hypergeometric terms. This works for all terms g with $D_C(f)\subseteq D_C(g)$. These terms form a subset of HT and for a moment let us denote it by Y_f. Then we can define an interpretation $\phi_f : Y_f \to R^{\mathbb{N}}$, where $R = \mathbb{C}^{D_C(f)}$. Since this $R^{\mathbb{N}}$ is closed under the summation operator, we know that $\sum \phi_f(f)$ is an element of $R^{\mathbb{N}}$ again. Assume that there exists a hypergeometric term g that represents this sum then g must be evaluable over $\mathbb{N} \times D_C(f)$. This means that $D_C(f) \subseteq D_C(g)$ and therefore $g \in Y_f$. We may look for the representation of the sum sequence only among those terms that can be interpreted with ϕ_f.

We adjust concept 1 to the latest explorations.

Concept 2 *(Finding closed forms for sum-expressions)*

We take HT as input domain. We need

1. A solution domain Y.
2. A restricted solution domain Y_f, for each f.
3. An R-sequence interpretation (Y_f, ϕ_f), for each f.
4. The operations $\mathtt{E}, \mathtt{L}, +$ and $-$, such that
   (a) HT is closed under $\mathtt{E}$,
   (b) Y_f is closed under $\mathtt{E}, \mathtt{L}, +$ and $-$,
   (c) ϕ_f is a $(L, E, +, -)$-homomorphism.
5. An algorithm that constructs $g \in Y_f$ to a given f, such that $\phi_f(\mathtt{E}g - g) = \phi_f(\mathtt{E}f)$, if such a g exists.

In the next chapter we will define the operations and an alternative to c-evaluation, since we will show that the properties above cannot be established, using c-evaluation.

Chapter 4

Operating with hypergeometric terms

We build up the structure on HT and define an interpretation according to concept 2.

4.1 Calculus for hypergeometric terms

As we defined our hypergeometric terms, we already introduced some structure. HT is a multiplicative abelian group. We define three more operations.

4.1.1 Adding hypergeometric terms in an extension

Instead of constructing a group of hypergeometric terms we could have built up a field of terms. We avoided doing so, because we do not consider expressions like $\frac{1}{\binom{x}{y}+x!}$. We are interested in sums that can be simplified to hypergeometric terms again:

$$\begin{aligned} x! + (x+1)! &= (x+2)\,x! \\ \binom{x}{y-1} + \binom{x}{y} &= \binom{x+1}{y}, \end{aligned}$$

and sums that must be left as they are:

$$x! - 1,\ \binom{x}{y-1} + \binom{2y}{x}.$$

In order to be able to represent sums of the second type we have to extend the set HT by finite sums:

Definition 11 *(Extension of* HT*)*

HT^+ is defined to be the term algebra that is generated from HT_0 by the operations $+$ and $-$, and is afterwards reduced by the abelian ring-axioms

$$\begin{aligned}
&t \cdot 0 = 0,\\
&t + 0 = t,\\
&t - t = 0,\\
&t_1 + t_2 = t_2 + t_1,\\
&(t_1 + t_2) + t_3 = t_1 + (t_2 + t_3),\\
&t_1 \cdot (t_2 + t_3) = t_1 \cdot t_2 + t_1 \cdot t_3,
\end{aligned}$$

and the additional axiom

$$(\varrho, f^-) + (\varsigma, f^-) = (\varrho + \varsigma, f^-).$$

Typed Variables 5 *(Terms)*

$$f^+, g^+, h^+ \in \mathrm{HT}^+.$$

4.1.2 Shifting of hypergeometric terms

Imagine you want to shift the hypergeometric term $\binom{x}{y}$ in x. You have two possibilities to achieve this. You can say $\binom{x+1}{y}$ is the shifted version of it. On the other hand, since this binomial satisfies a recurrence relation of order 1 you have $\frac{x+1}{x-y+1}\binom{x}{y}$ as the shifted version. The first method displays the functional technique of computing the shift of a function by composing it with the function $x+1$. The second method, however, displays the functional approach of computing the action of the shift operator on the binomial by means of the recurrence relation. Note that the results of the two shifting methods are different in HT.

We have to choose one of the methods, and indeed it is not hard to decide between both of them. As Gosper's algorithm heavily depends on the observation that the shift operator just changes the rational part of a hypergeometric term, we will have to follow this approach.

First we define the shift quotient, Q, which displays the change of the rational part by the action of shifting. The shift quotient will be a group homomorphism $Q : \mathrm{HT} \to \mathbb{C}(\vec{x})^*$, compare proposition 1; the shift operator is a ring endomorphism on HT^+, see definition 14.

Definition 12 *(Shift quotient Q)*

In this definition assert that ς is free of x_i. We define the shift quotient Q_i for $i = 1, \ldots, \nu$:

- Rational functions

$$Q_i \varrho(x_i) := \frac{\varrho(x_i + 1)}{\varrho(x_i)}.$$

- Factorials

  Fac is free over the set of elements used to generate Fac (refer to definition 8). Therefore the definition of Q_i for these elements uniquely extends to a group-homomorphism. Q_i : Fac $\rightarrow \mathbb{C}(\vec{x})^*$ is the unique extension of the following base cases:

$$Q_i(zx_i + \varsigma)! := \varphi(zx_i + \varsigma + 1, z).$$

- Falling factorials

  Q_i : FFac $\rightarrow \mathbb{C}(\vec{x})^*$ is the unique extension of the following base cases:

$$Q_i(r)_{\underline{s}} := Q_i \frac{r!}{(r-s)!}.$$

- Rising factorials

  Q_i : RFac $\rightarrow \mathbb{C}(\vec{x})^*$ is the unique extension of the following base cases:

$$Q_i(r)_{\overline{s}} := Q_i \frac{(r+s-1)!}{(r-1)!}.$$

- Binomials

  Q_i : Bin $\rightarrow \mathbb{C}(\vec{x})^*$ is the unique extension of the following base cases:

$$Q_i \binom{r}{s} := Q_i \frac{r!}{s!(r-s)!}.$$

- Exponentials

  For all $r_1^{x_1} \cdots r_i^{x_i} \cdots r_\nu^{x_\nu} \in$ Exp we define

$$Q_i \left(r_1^{x_1} \cdots r_i^{x_i} \ldots r_\nu^{x_\nu}\right) := r_i.$$

- Hypergeometric terms For all $(r, (a, b, c, d, e)) \in$ HT we define

$$Q_i(r, (a, b, c, d, e)) := Q_i r \; Q_i a \; Q_i b \; Q_i c \; Q_i d \; Q_i e.$$

Proposition 1 *(Homomorphic property of the shift quotient)*

$Q_i : \mathrm{HT} \to \mathbb{C}(\vec{x})^*$ is a group-homomorphism.

Proof

First we prove that the shift quotients for the components of a hypergeometric term are homomorphisms to the group $\mathbb{C}(\vec{x})^*$: Fix i.

1. (Rational functions) We have to show that

   $$Q_i(\varrho(x_i)\varsigma(x_i)) = Q_i\varrho(x_i) \cdot Q_i\varsigma(x_i),$$

   or equivalently:

   $$\frac{(\varrho(x_i)\varsigma(x_i)) \circ (x_i+1)}{\varrho(x_i)\varsigma(x_i)} = \frac{\varrho(x_i+1)\varsigma(x_i+1)}{\varrho(x_i)\varsigma(x_i)}.$$

2. (Factorials, falling factorials, rising factorials, binomials) Obvious from definition.

3. (Exponentials) To show the homomorphic property, we take arbitrary but fixed $r_1, \ldots, r_\nu$, $s_1, \ldots, s_\nu$ from $\mathbb{C}(\vec{x_{\nu+1}})^*$ and show that

   $$Q_i((r_1 s_1)^{x_1} \ldots (r_\nu s_\nu)^{x_\nu}) = Q_i(r_1^{x_1} \ldots r_\nu^{x_\nu}) Q_i(s_1^{x_1} \ldots s_\nu^{x_\nu}),$$

   or equivalently $r_i s_i = r_i s_i$.

The shift quotient of a hypergeometric term is the product of the shift quotients of the subexpressions. These shift quotients are elements of $\mathbb{C}(\vec{x})^*$, therefore the shift quotient is an element of $\mathbb{C}(\vec{x})^*$ as well.

The following chain of equalities proves the homomorphic property:

$$\begin{aligned} & Q_i\Big((\varrho, a, b, c, d, e) \cdot (\varrho', a', b', c', d', e')\Big) = \\ = \quad & Q_i(\varrho\varrho', aa', bb', cc', dd', ee') = \\ = \quad & Q_i(\varrho\varrho') \cdot Q_i(aa') \cdots Q_i(ee') = \\ = \quad & Q_i\varrho \cdot Q_i\varrho' \cdot Q_i a \cdot Q_i a' \cdots Q_i e' = \\ = \quad & Q_i(\varrho, a, b, c, d, e)\, Q_i(\varrho', a', b', c', d', e'). \end{aligned}$$

□

Proposition 2 *(Shift quotients in several variables)*

$$Q_iQ_jf = Q_jQ_if.$$

Proof

We prove this for $i = 1$ and $j = 2$:

1. (Rational functions) We have

$$Q_1Q_2(\varrho(x_1, x_2)) = Q_1\frac{\varrho(x_1, x_2 + 1)}{\varrho(x_1, x_2)} = \frac{\varrho(x_1 + 1, x_2 + 1)\varrho(x_1, x_2)}{\varrho(x_1 + 1, x_2)\varrho(x_1, x_2 + 1)}.$$

2. (Factorials) With $r = z_1x_1 + z_2x_2 + \varsigma$ we have,

$$Q_1Q_2(r!) = Q_1\varphi(r + 1, z_2) = \frac{\varphi(r + z_1 + 1, z_2)}{\varphi(r + 1, z_2)} =$$
$$\frac{\varphi\Big(r + \max(z_1, z_2) + 1, \min(z_1, z_2)\Big)\,\varphi\Big(r + z_1 + 1, \max(0, z_2 - z_1)\Big)}{\varphi\Big(r + 1, \min(z_1, z_2)\Big)\,\varphi\Big(r + \min(z_1, z_2) + 1, \max(0, z_2 - z_1)\Big)} =$$
$$\frac{\varphi(r + \max(z_1, z_2) + 1, \min(z_1, z_2))}{\varphi(r + 1, \min(z_1, z_2))}.$$

3. (Falling factorials, rising factorials, binomials) Clear, since in these cases the definition of the shift quotient is fully based on the shift quotient of factorials.

4. (Exponentials) We have

$$Q_1Q_2(r_1^{x_1} \ldots r_\nu^{x_\nu}) = Q_1r_2 = 1.$$

The commutative property easily carries over to hypergeometric terms.

□

Definition 13 *(Shift operator on* HT_0*)*

We define the shift-operator E_i for $i = 1, \ldots, \nu$ by

$$E_if := (Q_if) \cdot f,$$
$$E_i0 := 0.$$

Proposition 3 *(Homomorphic property of the shift operator)*

$E_i : \mathrm{HT} \to \mathrm{HT}$ is a group-endomorphism.

Proof

We have to show

$$E_i(f \cdot g) = E_i f \cdot E_i g.$$

Using the definitions and the fact that the shift quotient is a homomorphism and HT an abelian group, the following chain of equalities is true:

$$E_i(f \cdot g) = Q_i(f \cdot g) \cdot (f \cdot g) = (Q_i f) \cdot f \cdot (Q_i g) \cdot g = E_i f \cdot E_i g.$$

□

Definition 14 *(Shift operator on* HT$^+$*)*

We define the operator E_i on HT$^+$ to be the homomorphic extension of E_i on HT.

We show that the homomorphic extension exists. The algebra, we generated by the operations $+$ and $-$, is free over HT$_0$ and has a homomorphic extension of E_i. For each equation, used to reduce the free algebra, we can prove that both sides remain equal under the shift operator. We have:

$$\begin{aligned}
&E_i(t \cdot 0) = (E_i t) \cdot E_i 0 = 0 = E_i 0, \\
&E_i(t + 0) = E_i t + E_i 0 = E_i t, \\
&E_i(t - t) = E_i t + E_i((-1) \cdot t) = E_i t + Q_i t \cdot (-1) \cdot t = E_i t - E_i t = 0 = E_i 0 \\
&E_i(t_1 + t_2) = E_i(t_2 + t_1), \\
&E_i(t_1 + (t_2 + t_3)) = E_i((t_1 + t_2) + t_3), \\
&E_i(t_1 \cdot (t_2 + t_3)) = E_i(t_1 \cdot t_2 + t_1 \cdot t_3), \\
&E_i((\varrho, e) + (\varsigma, e)) = E_i(\varrho, e) + E_i(\varsigma, e) = \\
&\qquad = Q_i \varrho \cdot Q_i(1, e) + Q_i \varsigma \cdot Q_i(1, e) = (Q_i \varrho + Q_i \varsigma) \cdot Q_i(1, e) = \\
&\qquad = Q_i(\varrho + \varsigma) \cdot Q_i(1, e) = E_i(\varrho + \varsigma, e).
\end{aligned}$$

Because E_i is well defined, it is a ring-endomorphism.

Proposition 4 *(Shifts in several variables)*

$$E_i E_j f^+ = E_j E_i f^+.$$

Proof

Without loss of generality we prove the property above for $i = 1$ and $j = 2$. In HT we have,

$$E_1 E_2 f = E_1(Q_2 f \cdot f) = Q_1 Q_2 f \cdot Q_2 f \cdot Q_1 f \cdot f.$$

Use proposition 2 to complete the proof for $f \in \mathrm{HT}$.

Because both E_1 and E_2 are homomorphisms on HT^+, the property easily carries over to sums of hypergeometric terms.

□

4.1.3 The operator L_c

Definition 15 *(First element)*

$$L_c f := f[0, \vec{x_2}].$$

In the definition above we partially evaluate the hypergeometric term f. Note that this is impossible for the term $g := 1/x_1$. On the other hand we also have $D_C(g) = \{\}$. And indeed, if the interpretation domain of a hypergeometric term is not empty, then the action of L_c is well defined: Since $f[\vec{u}]$ is defined by evaluating sub-term by sub-term, compare definition 9, the substitution lemma – known from logics (see [EFT84]) – guarantees, that $(L_c f)[\vec{w}] = f[(0, \vec{w})]$. Now, if the interpretation domain of f is not empty, then we have at least one $\vec{w}$, for which $f[(0, \vec{w})]$ is well defined and therefore $L_c f$ is well defined, too.

4.1.4 Classic evaluation is not enough

According to concept 2, we have to guarantee, that for the interpretation ϕ_f of hypergeometric terms, a couple of homomorphic properties hold. Assume, we define ϕ_f via c-evaluation:

$$\phi_f(g) := (a_i)_{i \geq 0},\ a_i : D_C(f) \to \mathbb{C},\ a_i(\vec{w}) = g[(i, \vec{w})].$$

Then, take property $\phi_f(\mathtt{E}g) = E\phi_f(g)$ with the choice $g = f$, then we must have

$$\phi_f(E_1 f) = E\phi_f(f).$$

This equivalently reads as

$$\forall \vec{w} \in D_C(f) \colon\ (E_1 f)[i, \vec{w}] = f[(i+1, \vec{w})].$$

With the choice $f = \frac{1}{(x_1+x_2)!}$ we find that $D_C(f) = \mathbb{C}$ and that

$$\frac{1}{(x_1+x_2)!(x_1+x_2+1)}[i,w] = \frac{1}{(x_1+x_2)!}[i+1,w],$$

has to hold. But for the choice $i = 0$ and $w = -1$ we immediately see, that the left-hand side is not evaluable, whereas the right-hand side equals 0.

Shortly before reaching our goal we are faced with a wall forcing the detour of the additional section 4.2.

4.2 The evaluation model

According to concept 2 we are looking for an interpretation of hypergeometric terms. In section 4.1 we extended HT to HT^+ introducing the operators + and −, and also supplied the operators $\texttt{E} = E_1$ and $\texttt{L} = L_c$. We also gave an interpretation for hypergeometric terms as R-sequences. We have seen, that this interpretation (based on c-evaluation) is no $(E, L, +, -)$-homomorphism and does therefore not suffice.

Using the same approach as in section 4.1, but basing the interpretation upon a new evaluation, we will finally succeed. We are looking for a new method of evaluating hypergeometric terms. We give it a name and ...

$$\text{eval}$$

is born.

Obviously we want, that eval is just the same as c-evaluation, wherever both evaluations are defined. We also want that eval is a homomorphism. (Then also the interpretation based on eval will be a homomorphism.) Let's see, whether we can meet both requirements. First, we will focus on the latter.

For instance, we need that

$$\begin{aligned} \text{eval}(x!, 0) &= \text{eval}((x+1)x!, -1), \\ \text{eval}\left(\binom{x+2}{2}, 0\right) &= \text{eval}\left(\frac{x+3}{x+1}\binom{x+2}{2}, -1\right). \end{aligned}$$

For the last time we try out c-evaluation and also list how eval has to behave.

f	u	$f[u]$	$\text{eval}(f,u)$
$x!$	0	1	1
$(x+1)x!$	-1	$0\cdot\infty$	1
$\binom{x+2}{2}$	0	1	1
$\frac{x+3}{x+1}\binom{x+2}{2}$	-1	$\frac{2}{0}\cdot 0$	1

Here we encountered a problem that is similar to the problem of evaluating rational functions. The term $(x^2-1)\cdot\frac{1}{x-1}$, for instance, cannot be evaluated for $x := 1$, whereas it is possible to carry out this evaluation for the simplified term $x+1$. Both terms represent the same element in the rational function field, therefore both terms must have the same evaluation. In order to achieve this, we can calculate the canonical representation of a rational function ahead of evaluation. The standard canonical form of a term representing a rational function has the nice property that it can be evaluated whenever any other representation of the same term (with larger denominator) is evaluable. In this paper we have to do evaluation of rational functions, too. Let $\varrho[\vec{u}]$ denote the procedure of evaluating the canonical representation of the rational function ϱ at the tuple of values $\vec{u}$.

Unfortunately life is tougher with hypergeometric terms. There is this easy looking term $(2x)!/x!$ which cannot be evaluated for $x := -1$. Transforming this term to $(2x+2)!/(2(x+1)!(2x+1))$ we end up being able to evaluate for $x := -1$; but still we cannot evaluate for $x := -2$.

Therefore we decided that there is no canonical form for hypergeometric terms that is good enough – you might find one – to base our evaluation on. Our idea is neither to evaluate hypergeometric terms directly to a number field nor to calculate a canonical form before evaluation, but something in between.

4.2.1 Rational evaluation

We have to do the evaluation of a hypergeometric term in two steps. First we pick a rational function by means of a function ρ defined below, and then use c-evaluation ($\_[\_]$) to further evaluate the rational function to $\mathbb{C}$. This intermediate step, called rational evaluation, allows us to cancel out non-zero polynomial factors that evaluate to zero in the number field. The rational evaluation already depends on the point of evaluation. For instance, we have $\rho(x!,0) \neq \rho(x!,1)$.

The rational evaluation ρ of a rational function ϱ is defined as the function ϱ itself, since it is already rational ($\rho(\varrho,\vec{u}) = \varrho$). The rational evaluation of a factorial, $x!$ at i, is $\varphi(x-i+1,i)$ as we illustrate in a couple of examples:

Figure 4.1: Evaluation of hypergeometric terms

$$\begin{array}{rcrrcr}
\rho(x!,2) & = & (x-1)x & \rho(\frac{1}{x!},2) & = & \frac{1}{(x-1)x} \\
\rho(x!,1) & = & x & \rho(\frac{1}{x!},1) & = & \frac{1}{x} \\
\rho(x!,0) & = & 1 & \rho(\frac{1}{x!},0) & = & 1 \\
\rho(x!,-1) & = & \frac{1}{(x+1)} & \rho(\frac{1}{x!},-1) & = & x+1 \\
\rho(x!,-2) & = & \frac{1}{(x+1)(x+2)} & \rho(\frac{1}{x!},-2) & = & (x+1)(x+2)
\end{array}$$

As a second step, we do a standard evaluation of the rational evaluation. Both steps together, i.e., $\rho(x!,i)[i]$ result in what we expect from the evaluation of a factorial at i. The second step is undefined if $i=-1,-2$, and so on. Up to here we neither changed the domain of a factorial nor the result of its evaluation.

For the multiplicative inverse we already gained the evaluation to 0 for negative integers. We give an example: To evaluate $\frac{1}{x!}$ at -1 we first apply rational evaluation with -1 as evaluation-point: $\rho(\frac{1}{x!},-1)=x+1$. Evaluating this term at -1 we come up with the predicted zero: $(x+1)[-1]=0$.

Another advantage lies in the fact that once you multiply $x!$ with $x+1$ the rational evaluation will loose the $x+1$ in the denominator, since it cancels with the extra polynomial factor $x+1$. Thus the product $x!(x+1)$ becomes evaluable at -1:

We apply the evaluation method on the examples from the beginning of this section:

f	u	$\rho(f,u)$	$\text{eval}(f,u)=\rho(f,u)[u]$
$x!$	0	1	1
$x!$	-1	$\frac{1}{x+1}$	∞
$x+1$	-1	$x+1$	0
$(x+1)x!$	-1	$(x+1)\frac{1}{x+1}=1$	1
$\binom{x+2}{2}$	0	$\frac{(x+1)(x+2)}{2}$	1
$\frac{x+3}{x+1}\binom{x+2}{2}$	-1	$\frac{(x+3)(x+2)(x+1)}{(x+1)\,2}=\frac{(x+3)(x+2)}{2}$	1

Now there is enough work left to do:

- We define rational evaluation for all hypergeometric terms.
- Based on rational evaluation we define the functional interpretation of a hypergeometric term (i.e., domain and evaluation).
- We investigate the relation between the evaluation defined by this model and c-evaluation.

Definition 16 *(Rational evaluation)*

We define the rational evaluation ρ for fixed $\vec{u}$.

- Rational functions

  $$\rho(\varrho, \vec{u}) := \varrho.$$

- Factorials

  Fac is free over the set of elements that generate Fac (refer to definition 8). Therefore the definition of ρ for these elements uniquely extends to a group-homomorphism, $\rho(\_, \vec{u}) : \text{Fac} \to \mathbb{C}(\vec{x})^*$ is the unique extension of the following base cases:

  $$\begin{aligned} \rho(r!, \vec{u}) &:= \Gamma(r[\vec{u}] - c + 1)\ \varphi(r - c + 1, c), \\ & \ c = \lfloor \mathrm{Re}(r[\vec{u}]) \rfloor. \end{aligned}$$

- Falling/Rising factorials

  $\rho(\_, \vec{u}) : \text{FFac} \to \mathbb{C}(\vec{x})^*$ and $\rho(\_, \vec{u}) : \text{RFac} \to \mathbb{C}(\vec{x})^*$ are the unique extensions of the following base cases:

  $$\rho((r)_{\underline{s}}, \vec{u}) := \rho\left(\frac{r!}{(r-s)!}, \vec{u}\right),$$
  $$\rho((r)_{\overline{s}}, \vec{u}) := \rho\left(\frac{(r+s-1)!}{(r-1)!}, \vec{u}\right).$$

- Binomials

  $\rho(\_, \vec{u}) : \text{Bin} \to \mathbb{C}(\vec{x})^*$ is the unique extension of the following base cases:

  $$\rho\left(\binom{r}{s}, \vec{u}\right) := \rho\left(\frac{r!}{s!(r-s)!}, \vec{u}\right).$$

- Exponentials

  For all $r_1^{x_1} r_2^{x_2} \cdots r_\nu^{x_\nu} \in \mathrm{Exp}$ and $\vec{u} = (u_1, u_2, \ldots, u_\mu)$ we define:

  $$\rho\left(r_1^{x_1} r_2^{x_2} \cdots r_\nu^{x_\nu}, \vec{u}\right) := r_1^{u_1} r_2^{u_2} \cdots r_\nu^{u_\nu}.$$

- Hypergeometric terms

  For all $(\varrho, a, b, c, d, e) \in \mathrm{HT}$ we define:

  $$\rho((\varrho, a, b, c, d, e), \vec{u}) := \rho(r, \vec{u})\, \rho(a, \vec{u}) \, \cdots \, \rho(e, \vec{u}),$$
  $$\rho(0, \vec{u}) = 0.$$

If the rational evaluation of factorials is unequal zero, then it is clear that it is non-zero for all hypergeometric terms. Then, rational evaluation is indeed well defined:

Proposition 5 *(Rational evaluation of factorials is non-zero)*

$$\forall r! \in \mathrm{Fac}\colon\ \rho(r!, \vec{u}) \neq 0.$$

Proof

Let $\vec{u}$ and $r! \in \mathrm{Fac}$ be arbitrary but fixed, then we have

$$\rho(r!, \vec{u}) = \Gamma(r[\vec{u}] - c + 1)\, \varphi(r - c + 1, c),$$

where $c = \lfloor \mathrm{Re}(r[\vec{u}]) \rfloor$. The Gamma function is never 0, thus we have to consider only the Pochhammer function. It is obviously non-zero, unless $r[\vec{u}]$ is an integer. In that case we know, that $r[\vec{u}] \in \mathbb{N}$ (compare definition 8). Therefore the expression simplifies to the non-zero term $\varphi(1, r[\vec{u}])$.

□

Proposition 6 *(Homomorphic property of rational evaluation)*

For any $\vec{u}$ the function $\rho(\_, \vec{u}) : \mathrm{HT} \to \mathbb{C}(\vec{x})^*$ is a group-homomorphism.

Proof

Let $\vec{u}$ be arbitrary but fixed. First we prove that the rational evaluations for the components of a hypergeometric term are homomorphisms to the group $\mathbb{C}(\vec{x})^*$:

1. (Rational functions)

$$\rho(\varrho\varsigma, \vec{u}) = \rho(\varrho, \vec{u})\rho(\varsigma, \vec{u}),$$

is clear by looking at the equivalent equation $\varrho\varsigma = \varrho\varsigma$.

2. (Factorials, falling factorials, rising factorials, binomials) The homomorphic property is clear from construction.

3. (Exponentials) We take arbitrary but fixed $r_1, \ldots, r_\nu,\ s_1, \ldots, s_\nu \in \mathbb{C}(\vec{x_{\nu+1}})^*$ and prove

$$\rho((r_1s_1)^{x_1} \cdots (r_\nu s_\nu)^{x_\nu}, \vec{u}) = \rho(r_1^{x_1} \cdots r_\nu^{x_\nu}, \vec{u})\rho(s_1^{x_1} \cdots s_\nu^{x_\nu}, \vec{u})$$

by the equivalent and trivial equation

$$(r_1s_1)^{u_1} \ldots (r_\nu s_\nu)^{u_\nu} = r_1^{u_1} \ldots r_\nu^{u_\nu} s_1^{u_1} \ldots s_\nu^{u_\nu}.$$

Now using the facts above, we can prove the homomorphic property for any hypergeometric term by the following chain of equalities:

$$\begin{aligned}
&\rho((\varrho, a, b, c, d, e) \cdot (\varrho', a', b', c', d', e'), \vec{u}) = \\
&= \rho((\varrho\varrho', aa', bb', cc', dd', ee'), \vec{u}) = \\
&= \rho(\varrho\varrho', \vec{u})\ \rho(aa', \vec{u})\ \cdots\ \rho(dd', \vec{u})\rho(ee', \vec{u}) = \\
&= \rho(\varrho, \vec{u})\ \rho(\varrho', \vec{u})\ \rho(a, \vec{u})\ \cdots\ \rho(e, \vec{u})\ \rho(e', \vec{u}) = \\
&= \rho((\varrho, a, b, c, d, e), \vec{u})\rho((\varrho', a', b', c', d', e'), \vec{u}).
\end{aligned}$$

□

Definition 17 *(Rational evaluation on sums)*

For fixed $\vec{u}$ we define $\rho(f^+, \vec{u})$ to be the homomorphic extension of $\rho(\_, \vec{u})$ on HT_0.

We show that the homomorphic extension exists. The algebra, we generated by the operation + is free over HT_0 and has a homomorphic extension of $\rho(\_, \vec{u})$. For each equation, used to reduce this algebra, we can prove that both sides remain equal under rational evaluation. With $\rho(t)$ short for $\rho(t, \vec{u})$, we have:

$$\begin{aligned}
&\rho(t \cdot 0) = \rho(t) \cdot \rho(0) = 0 = \rho(0),\\
&\rho(t + 0) = \rho(t) + \rho(0) = \rho(t),\\
&\rho(t - t) = \rho(t) + \rho(-1 \cdot t) = \rho(t) - \rho(t) = 0 = \rho(0),\\
&\rho(t_1 + t_2) = \rho(t_2 + t_1),\\
&\rho(t_1 + (t_2 + t_3)) = \rho((t_1 + t_2) + t_3),\\
&\rho(t_1 \cdot (t_2 + t_3)) = \rho(t_1 \cdot t_2 + t_1 \cdot t_3),\\
&\rho((\varrho, t_0) + (\varsigma, t_0)) = \rho(\varrho, t_0) + \rho(\varsigma, t_0) =\\
&\qquad = \rho(\varrho)\rho(1, t_0) + \rho(\varsigma)\rho(1, t_0) = (\varrho + \varsigma)\rho(1, t_0) =\\
&\qquad = \rho(\varrho + \varsigma)\rho(1, t_0) = \rho(\varrho + \varsigma, t_0).
\end{aligned}$$

Since the homomorphic extension of the rational evaluation is well defined, it is a ring-homomorphism.

4.2.2 Functional interpretation of hypergeometric terms

Based on the rational evaluation we define a domain for a hypergeometric term and an evaluation to the complex numbers.

Definition 18 *(Functional interpretation)*

We define m-evaluation of hypergeometric terms

$$\begin{aligned}
&D_m(f^+) := \{\vec{u} \mid \vec{u} \text{ is no root of } \mathrm{den}(\rho(f^+, \vec{u}))\},\\
&\forall \vec{u} \in D_m(f^+)\colon \ \mathrm{eval}(f^+, \vec{u}) := \rho(f^+, \vec{u})[\vec{u}].
\end{aligned}$$

In the following we will have to compare m-evaluation to c-evaluation, therefore it is convenient to define an abbreviation for the intersection of the domains of c- and m-evaluation.

In section 3.2.4 we had a discussion on interpreting hypergeometric terms as sequences. We found out that we can base such an interpretation upon an evaluation. Following these lines we define a domain for the sequence-interpretation of a hypergeometric term, and again for matters of comparison, it is convenient to have an abbreviation for the intersection of this domain with the corresponding domain, based on c-evaluation.

Finally, also motivated in section 3.2.4 by concept 2, we define the solution-domain for symbolic sums of a hypergeometric term.

Definition 19 *(Domains)*

- Domain of sequence-interpretation

$$D_M(f^+) := \{\vec{w} \mid \mathbb{N} \times \{\vec{w}\} \subseteq D_m(f^+)\}.$$

- Intersections
  $$D_{c,m}(f^+) := D_c(f^+) \cap D_m(f^+),$$
  $$D_{C,M}(f^+) := D_C(f^+) \cap D_M(f^+).$$

- Solution-domain for symbolic sums
  $$\mathrm{HT}_f^+ := \{g^+ \mid D_M(f) \subseteq D_M(g^+)\}.$$

The open problem is the operator `L`, since we cannot use L_c, which is defined using c-evaluation, compare definition 15. It is possible to give a definition of L_m based on the evaluation model. The resulting operation, though is impractical. As we see in the following we will not have to compute L_m, therefore I ask you to accept that it is possible to give an appropriate definition of L_m.

4.2.3 Properties of the evaluation model

It is time to reconsider what we had in mind when we started out defining the evaluation model. The purpose was to introduce an evaluation of hypergeometric terms that cooperates with shifting. In this subsection we will prove this and similar properties.

Lemma 1 *(Relation between ρ and Q_i)*

$$\frac{\rho(f,(u+1,\vec{w})) \circ (x_1+1)}{\rho(f,(u,\vec{w}))} = Q_1 f. \tag{4.1}$$

Proof

First we deal with the components of a hypergeometric term and then combine this knowledge to prove equation (4.1):

Let $\vec{w}$ and u be arbitrary but fixed.

1. (Factorials)
   Let $r = zx_1 + h$ and $c = \lfloor \mathrm{Re}(r[(u,\vec{w})]) \rfloor$, then we have
   $$\frac{\rho(r!,(u+1,\vec{w})) \circ (x_1+1)}{\rho(r!,(u,\vec{w}))} =$$
   $$= \frac{\Gamma(z(u+1)+h[\vec{w}]-c-z+1)\ \varphi(r-c-z+1,c+z) \circ (x_1+1)}{\Gamma(zu+h[\vec{w}]-c+1)\ \varphi(r-c+1,c)} =$$
   $$= \frac{\varphi(r-c+1,c+z)}{\varphi(r-c+1,c)} = \varphi(r+1,z) = Q_1 r!.$$

2. (Falling factorials, rising factorials and binomials) The definitions of shift quotient and the rational evaluation are both based on the ones of factorials. We carry out the proof for rising factorials:

$$\begin{aligned} &\frac{\rho((r)_{\overline{s}}, (u+1, \vec{w})) \circ (x_1+1)}{\rho((r)_{\overline{s}}, (u, \vec{w}))} = \\ &= \frac{\rho((r+s-1)!, (u+1, \vec{w})) \circ (x_1+1)\ \rho((r-1)!, (u, \vec{w}))}{\rho((r+s-1)!, (u, \vec{w}))\ \rho((r-1)!, (u+1, \vec{w})) \circ (x_1+1)} = \\ &= \frac{Q_1(r+s-1)!}{Q_1(r-1)!} = Q_1(r)_{\overline{s}}. \end{aligned}$$

3. (Exponentials) For arbitrary $r_1^{x_1} \cdots r_\nu^{x_\nu} \in \text{Exp}$ and with $\vec{w} = (w_2, \ldots, w_\mu)$, the following is true:

$$\begin{aligned} &\frac{\rho(r_1^{x_1} \cdots r_\nu^{x_\nu}, (u+1, \vec{w})) \circ (x_1+1)}{\rho(r_1^{x_1} \cdots r_\nu^{x_\nu}, (u, \vec{w}))} = \\ &= \frac{r_1^{u+1} r_2^{w_2} \cdots r_\nu^{w_\nu}}{r_1^{u} r_2^{w_2} \cdots r_\nu^{w_\nu}} = r_1 = Q_1(r_1^{x_1} \cdots r_\nu^{x_\nu}). \end{aligned}$$

4. It is left to combine the knowledge for arbitrary elements $(\varrho, a, b, c, d, e) \in \text{HT}$:

$$\begin{aligned} &\frac{\rho((\varrho, a, b, c, d, e), (u+1, \vec{w})) \circ (x_1+1)}{\rho((\varrho, a, b, c, d, e), (u, \vec{w}))} = \\ &= \frac{\rho(\varrho, (u+1, \vec{w})) \circ (x_1+1) \cdots \rho(e, (u+1, \vec{w})) \circ (x_1+1)}{\rho(\varrho, (u, \vec{w})) \cdots \rho(e, (u, \vec{w}))} = \\ &= Q_1\varrho \cdots Q_1 e = Q_1(\varrho, a, b, c, d, e). \end{aligned}$$

□

It is clear that the lemma holds for $x_1, \ldots, x_\nu$. Since this is hard to notate in a readable manner, it is written for x_1, only.

Proposition 7 *(Properties of the evaluation model)*

The following statements hold independently:

1. Multiplication:

$$\vec{u} \in D_m(f^+) \cap D_m(g^+) \Rightarrow \begin{cases} \vec{u} \in D_m(f^+ \cdot g^+), \\ \text{eval}(f^+ \cdot g^+, \vec{u}) = \text{eval}(f^+, \vec{u})\ \text{eval}(g^+, \vec{u}). \end{cases}$$

2. Inversion:

$$\vec{u} \in D_m(f),\ \text{eval}(f,\vec{u}) \neq 0 \Rightarrow \begin{cases} \vec{u} \in D_m(\frac{1}{f}), \\ \text{eval}(\frac{1}{f},\vec{u}) = 1/\text{eval}(f,\vec{u}). \end{cases}$$

3. Addition:

$$\begin{aligned} &\vec{u} \in D_m(f^+) \cap D_m(g^+) \Rightarrow \\ &\quad \Rightarrow \begin{cases} \vec{u} \in D_m(f^+ + g^+), \\ \text{eval}(f^+ + g^+,\vec{u}) = \text{eval}(f^+,\vec{u}) + \text{eval}(g^+,\vec{u}). \end{cases} \end{aligned}$$

4. Subtraction:

$$\vec{u} \in D_m(f^+) \Rightarrow \begin{cases} \vec{u} \in D_m(-f^+), \\ \text{eval}(-f^+,\vec{u}) = -\text{eval}(f^+,\vec{u}). \end{cases}$$

5. Shifting:

$$\begin{aligned} &(u+1,\vec{w}) \in D_m(f) \Leftrightarrow (u,\vec{w}) \in D_m(E_1 f), \\ &(u+1,\vec{w}) \in D_m(f) \Rightarrow \text{eval}(f,(u+1,\vec{w})) = \text{eval}(E_1 f,(u,\vec{w})). \end{aligned}$$

Proof

Let $f, g, f^+, g^+, \vec{u}$ be arbitrary but fixed.

1. Assume $\vec{u} \in D_m(f^+) \cap D_m(g^+)$ then we know

$$\begin{aligned} &\text{den}(\rho(f^+,\vec{u}))[\vec{u}] \neq 0, \\ &\text{den}(\rho(g^+,\vec{u}))[\vec{u}] \neq 0. \end{aligned}$$

   Since ρ is a ring-homomorphism we easily conclude

$$\text{den}(\rho(f^+ \cdot g^+,\vec{u}))[\vec{u}] \neq 0.$$

   So the following equality is true:

$$\begin{aligned} \text{eval}(f^+ \cdot g^+,\vec{u}) &= \rho(f^+ \cdot g^+,\vec{u})[\vec{u}] = (\rho(f^+,\vec{u})\rho(g^+,\vec{u}))[\vec{u}] = \\ &= \rho(f^+,\vec{u})[\vec{u}]\rho(g^+,\vec{u})[\vec{u}] = \text{eval}(f^+,\vec{u})\text{eval}(g^+,\vec{u}). \end{aligned}$$

2. Assume $\vec{u} \in D_m(f)$ with $\text{eval}(f,\vec{u}) \neq 0$ then we have

$$\begin{aligned} &\text{num}(\rho(f,\vec{u}))[\vec{u}] \neq 0, \\ &\text{den}(\rho(\frac{1}{f},\vec{u}))[\vec{u}] \neq 0. \end{aligned}$$

   So the following holds:

$$\text{eval}(\frac{1}{f}, \vec{u}) = \frac{1}{\rho(f,\vec{u})}[\vec{u}] = \frac{1}{\rho(f,\vec{u})[\vec{u}]} = \frac{1}{\text{eval}(f,\vec{u})}.$$

3. Assume that $\vec{u} \in D_m(f^+) \cap D_m(g^+)$, then analogously to before we conclude that $\text{den}(\rho(f^+ + g^+, \vec{u})) \neq 0$ and the following holds:

$$\begin{aligned} &\text{eval}(f^+ + g^+, \vec{u}) = \rho(f^+ + g^+, \vec{u})[\vec{u}] = \\ &= \rho(f^+, \vec{u})[\vec{u}] + \rho(g^+, \vec{u})[\vec{u}] = \text{eval}(f^+, \vec{u}) + \text{eval}(g^+, \vec{u}). \end{aligned}$$

4. Clear, by an analogous argument.

5. Let $u, \vec{w}$ be arbitrary but fixed, such that $(u+1, \vec{w}) \in D_m(f)$. Then we have

$$\begin{aligned} 0 &\neq \text{den}(\rho(f, (u+1, \vec{w})))[(u+1, \vec{w})] = \\ &= \text{den}(\rho(f, (u+1, \vec{w})) \circ (x_1 + 1))[(u, \vec{w})]. \end{aligned} \tag{4.2}$$

Using lemma 1, we have

$$\begin{aligned} &\rho(f, (u+1, \vec{w})) \circ (x_1 + 1) = \\ &= \rho(f, (u, \vec{w})) \cdot Q_1(f) = \rho(f, (u, \vec{w})) \cdot \rho(Q_1(f), (u, \vec{w})) = \\ &= \rho(f \cdot Q_1(f), (u, \vec{w})) = \rho(E_1 f, (u, \vec{w})). \end{aligned}$$

By replacing $\rho(f, (u+1, \vec{w})) \circ (x_1 + 1)$ in Equation (4.2) we find, that

$$\text{den}(\rho(E_1 f, (u, \vec{w})))[(u, \vec{w})] \neq 0.$$

The equalities

$$\begin{aligned} &\text{eval}(f, (u+1, \vec{w})) = \rho(f, (u+1, \vec{w}))[(u+1, \vec{w})] = \\ &= \rho(f, (u+1, \vec{w})) \circ (x_1 + 1)[(u, \vec{w})] = \rho(E_1 f, (u, \vec{w}))[(u, \vec{w})] = \\ &= \text{eval}(E_1 f, (u, \vec{w})). \end{aligned}$$

complete the proof.

□

4.2.4 Comparison with classic evaluation

Finally we try to prove that the m-evaluation is sound, i.e., it should coincide with the c-evaluation of factorials and binomials. To our pity the m-evaluation of falling factorials, rising factorials and binomials is bound to the additional condition, that for the terms $(r)_{\underline{s}}$ and $\binom{r}{s}$ the sub-term r must not evaluate to a negative integer and that for $(r)_{\overline{s}}$ the sub-term $r+s$ must not evaluate to a non-positive integer. But there is a way to work around this bug, compare proposition 9.

Proposition 8 *(Evaluation of hypergeometric terms is correct)*

The following statements hold for arbitrary $\vec{u}$.

$$\text{eval}(\varrho, \vec{u}) = \begin{cases} \varrho[\vec{u}], & \text{if } \text{den}(r)[\vec{u}] \neq 0, \\ \text{undefined}, & \text{else.} \end{cases}$$

$$\text{eval}(r!, \vec{u}) = \begin{cases} \Gamma(r[\vec{u}]+1), & \text{if } r[\vec{u}] \notin \mathbb{Z}^-, \\ \text{undefined}, & \text{if } r[\vec{u}] \in \mathbb{Z}^-; \end{cases}$$

$$\text{eval}(\frac{1}{r!}, \vec{u}) = \begin{cases} \frac{1}{\Gamma(r[\vec{u}]+1)}, & \text{if } x_0 \notin \mathbb{Z}^-, \\ 0, & \text{if } x_0 \in \mathbb{Z}^-; \end{cases}$$

$$\text{eval}((r)_{\underline{s}}, \vec{u}) = \begin{cases} \frac{\Gamma(r[\vec{u}]+1)}{\Gamma((r-s)[\vec{u}]+1)}, & \text{if } r[\vec{u}] \notin \mathbb{Z}^- \wedge (r-s)[\vec{u}] \notin \mathbb{Z}^-, \\ 0, & \text{if } r[\vec{u}] \notin \mathbb{Z}^- \wedge (r-s)[\vec{u}] \in \mathbb{Z}^-, \\ \text{undefined}, & \text{if } r[\vec{u}] \in \mathbb{Z}^-; \end{cases}$$

$$\text{eval}(\frac{1}{(r)_{\underline{s}}}, \vec{u}) = \begin{cases} \frac{\Gamma((r-s)[\vec{u}]+1)}{\Gamma(r[\vec{u}]+1)}, & \text{if } r[\vec{u}] \notin \mathbb{Z}^- \wedge (r-s)[\vec{u}] \notin \mathbb{Z}^-, \\ 0, & \text{if } r[\vec{u}] \in \mathbb{Z}^- \wedge (r-s)[\vec{u}] \notin \mathbb{Z}^-, \\ \text{undefined}, & \text{if } (r-s)[\vec{u}] \in \mathbb{Z}^-. \end{cases}$$

$$\text{eval}((r)_{\overline{s}}, \vec{u}) = \begin{cases} \frac{\Gamma((r+s)[\vec{u}])}{\Gamma(r[\vec{u}])}, & \text{if } (r+s)[\vec{u}] \notin \mathbb{Z}_0^- \wedge r[\vec{u}] \notin \mathbb{Z}_0^-, \\ 0, & \text{if } (r+s)[\vec{u}] \notin \mathbb{Z}_0^- \wedge r[\vec{u}] \in \mathbb{Z}_0^-, \\ \text{undefined}, & \text{if } (r+s)[\vec{u}] \in \mathbb{Z}_0^-; \end{cases}$$

$$\text{eval}(\frac{1}{(r)_{\overline{s}}}, \vec{u}) = \begin{cases} \frac{\Gamma(r[\vec{u}])}{\Gamma((r+s)[\vec{u}])}, & \text{if } r[\vec{u}] \notin \mathbb{Z}_0^- \wedge (r+s)[\vec{u}] \notin \mathbb{Z}_0^-, \\ 0, & \text{if } r[\vec{u}] \notin \mathbb{Z}_0^- \wedge (r+s)[\vec{u}] \in \mathbb{Z}_0^-, \\ \text{undefined}, & \text{if } r[\vec{u}] \in \mathbb{Z}_0^-. \end{cases}$$

$$\text{eval}(\binom{r}{s}, \vec{u}) = \begin{cases} \frac{(r)_{\underline{s}}}{s!}[\vec{u}], & \text{if } r[\vec{u}] \notin \mathbb{Z}^- \wedge s[\vec{u}] \notin \mathbb{Z}^-, \\ 0, & \text{if } r[\vec{u}] \notin \mathbb{Z}^- \wedge s[\vec{u}] \in \mathbb{Z}^-, \\ \text{undefined}, & \text{if } r[\vec{u}] \in \mathbb{Z}^-; \end{cases}$$

$$\text{eval}(\frac{1}{\binom{r}{s}}, \vec{u}) = \begin{cases} \frac{s!}{(r)_{\underline{s}}}[\vec{u}], & \text{if } (r-s)[\vec{u}] \notin \mathbb{Z}^- \wedge s[\vec{u}] \notin \mathbb{Z}^-, \\ \text{undefined}, & \text{if } (r-s)[\vec{u}] \in \mathbb{Z}^- \vee s[\vec{u}] \in \mathbb{Z}^-. \end{cases}$$

Let $\vec{u} = (u_1, \ldots, u_\mu)$, then

$$\text{eval}(r_1^{x_1} \cdots r_\nu^{x_\nu}, \vec{u}) = r_1[\vec{u}]^{u_1} \cdots r_\nu[\vec{u}]^{u_\nu}.$$

Proof

1. (Rational functions) The proposition is obvious.

2. (Factorials) To examine the domain of $r!$ we have to take a look at the denominator of the rational evaluation of $r!$. Recall that

   $$\rho(r!, \vec{u}) = \Gamma(r[\vec{u}] - c + 1)\, \varphi(r - c + 1, c),\ c = \lfloor \mathrm{Re}(r[\vec{u}]) \rfloor.$$

   The Pochhammer function contributes to the denominator only if c is negative. These contributions are $(r+1), (r+2), \ldots, (r-c)$; thus their product evaluates to 0 if and only if $r[\vec{u}] \in \mathbb{Z}^-$.

   Otherwise:

   $$\begin{aligned} &\mathrm{eval}(r!, \vec{u}) = \\ &= \Gamma(r[\vec{u}] - c + 1)\, \varphi(r - c + 1, c)[\vec{u}] = \Gamma(r[\vec{u}] + 1). \end{aligned}$$

   The domain of $\frac{1}{r!}$ is not restricted: If the Pochhammer function contributes to the denominator of the rational evaluation, c has to be positive and therefore also $\mathrm{Re}(r[\vec{u}]) \geq 1$. The contributions are of form $r, (r-1)$, and so on. The smallest factor will be $r - c + 1$ which evaluates positively for positive $r[\vec{u}]$. Due to the equation above the evaluation of the inverse is $\frac{1}{\Gamma(r[\vec{u}]+1)}$.

3. (Falling factorials) Again we analyze the denominator of the rational evaluation,

   $$\rho((r)_{\underline{s}}, \vec{u}) = \rho\left(\frac{r!}{(r-s)!}, \vec{u}\right):$$

   The factorial in the denominator never contributes factors evaluating to zero to the rational evaluation. However, $r!$ does, if and only if $r[\vec{u}] \in \mathbb{Z}^-$ (as discussed above).

   For the multiplicative inverse obviously the critical term is given by $(r-s)!$; So the domain is restricted by $(r-s)[\vec{u}] \notin \mathbb{Z}^-$.

   The evaluation for $\vec{u} \in D_m(\cdots)$ is obvious.

4. (Rising factorials) This case can be treated analogously to the case ahead.

5. (Binomials)

   $$\rho\left(\binom{r}{s}, \vec{u}\right) = \rho\left(\frac{r!}{s!(r-s)!}, \vec{u}\right):$$

   Thus the domain is restricted by $r[\vec{u}] \notin \mathbb{Z}^-$.

   Again the multiplicative inverse can be treated analogously, and the evaluations are obvious.

6. (Exponentials) The proposition is obvious.

□

As you can see by proposition 8 the evaluation model handles rational functions, factorials and exponentials correctly.

Falling factorials, rising factorials and binomials are evaluated correctly over their domain D_m. This domain, however is smaller than the standard evaluation suggests.

Falling factorials $(r)_{\underline{s}}$ are additionally undefined for $r[\vec{u}], (r-s)[\vec{u}] \in \mathbb{Z}^-$, Rising factorials $(r)_{\overline{s}}$ for $(r+s)[\vec{u}] \in \mathbb{Z}_0^-$ and binomials $\binom{r}{s}$ for $r[\vec{u}] \in \mathbb{Z}^-$.

To have a tool to deal with these cases we state some limit properties for the evaluation of these terms:

Proposition 9 *(Limit properties)*

1. (Falling factorials)

$$\lim_{\varepsilon \to 0} \operatorname{eval}\Big((r+\varepsilon)_{\underline{s}}, \vec{u}\Big) = (r)_{\underline{s}}[\vec{u}].$$

2. (Rising factorials)

$$\lim_{\varepsilon \to 0} \operatorname{eval}\Big((r+\varepsilon)_{\overline{s}}, \vec{u}\Big) = (r)_{\overline{s}}[\vec{u}].$$

3. (Binomials)

$$\lim_{\varepsilon \to 0} \operatorname{eval}\left(\binom{r+\varepsilon}{s}, \vec{u}\right) = \binom{r}{s}[\vec{u}].$$

Proof

1. (Falling factorials) Using proposition 8 we have:

$$\begin{aligned} &\lim_{\varepsilon \to 0} \operatorname{eval}\Big((r+\varepsilon)_{\underline{s}}, \vec{u}\Big) = \\ &= \lim_{\varepsilon \to 0} \frac{\Gamma(r[\vec{u}]+\varepsilon+1)}{\Gamma((r-s)[\vec{u}]+\varepsilon+1)} = \\ &= \lim_{r \to r[\vec{u}]} \frac{\Gamma(r+1)}{\Gamma(r-s[\vec{u}]+1)} = (r)_{\underline{s}}[\vec{u}]. \end{aligned}$$

2. (Rising factorials) The proof can be done in just the same way.
3. (Binomials) Here we have to distinguish between the cases $s[\vec{u}] \in \mathbb{Z}^-$ and $s[\vec{u}] \notin \mathbb{Z}^- \wedge r[\vec{u}] \in \mathbb{Z}^-$. There is a third case where neither of the two previous cases applies; there, however, both possibilities of evaluating the Binomial are not defined.

$$s[\vec{u}] \in \mathbb{Z}^- :$$
$$\lim_{\varepsilon \to 0} \text{eval}\left(\binom{r+\varepsilon}{s}, \vec{u}\right) = 0 = \binom{r}{s}[\vec{u}],$$
$$s[\vec{u}] \notin \mathbb{Z}^- \wedge r[\vec{u}] \in \mathbb{Z}\backslash\mathbb{N} :$$
$$\lim_{\varepsilon \to 0} \text{eval}\left(\binom{r+\varepsilon}{s}, \vec{u}\right) = \lim_{\varepsilon \to 0} \frac{(r[\vec{u}]+\varepsilon)_{\underline{s[\vec{u}]}}}{s![\vec{u}]} \binom{r}{s}[\vec{u}].$$

□

We did not discuss the problematic cases of the multiplicative inverses of binomials, falling and rising factorials. These expressions can be rewritten (within the theory of the evaluation model) according to the following proposition, and are therefore also subject to the limit properties stated above.

Proposition 10 *(Hypergeometric exponential laws)*

1. Falling factorials

$$D_m\left(\frac{1}{(r)_{\underline{s}}}\right) = D_m((r-s)_{\underline{-s}}),$$
$$\text{eval}\left(\frac{1}{(r)_{\underline{s}}}, \vec{u}\right) = \text{eval}((r-s)_{\underline{-s}}, \vec{u});$$

2. Rising factorials

$$D_m\left(\frac{1}{(r)_{\overline{s}}}\right) = D_m((r+s)_{\overline{-s}}),$$
$$\text{eval}\left(\frac{1}{(r)_{\overline{s}}}, \vec{u}\right) = \text{eval}((r+s)_{\overline{-s}}, \vec{u});$$

3. Binomials

$$\begin{aligned} D_m\left(\frac{1}{\binom{r}{s}}\right) &= D_m(s!) \cap D_m((r-s)_{\underline{-s}}), \\ \mathrm{eval}\left(\frac{1}{\binom{r}{s}}, \vec{u}\right) &= \mathrm{eval}(s!\,(r-s)_{\underline{-s}}, \vec{u}); \end{aligned}$$

Proof

If two different hypergeometric terms have the same rational evaluation, then they clearly have the same domain and the same evaluation. So we show equality of rational evaluation for each pair of hypergeometric terms above.

$$\rho\left(\frac{1}{(r)_{\underline{s}}}, \vec{u}\right) = \rho\left(\frac{(r-s)!}{r!}, \vec{u}\right) = \rho\left((r-s)_{\underline{-s}}, \vec{u}\right),$$
$$\rho\left(\frac{1}{(r)_{\overline{s}}}, \vec{u}\right) = \rho\left(\frac{(r-1)!}{(r+s-1)!}, \vec{u}\right) = \rho\left((r+s)_{\overline{-s}}, \vec{u}\right),$$

$$\rho\left(\frac{1}{\binom{r}{s}}, \vec{u}\right) = \rho\left(\frac{r!(r-s)!}{r!}, \vec{u}\right) = \rho\left(s!\,(r-s)_{\underline{-s}}, \vec{u}\right).$$

□

We found, that the domains of c- and m-evaluation are different. On their intersection $D_{c,m}$, they yield the same result. On $D_c \backslash D_m$ c-evaluation usually can be retrieved by approaching it by a limit of m-evaluations. On $D_m \backslash D_c$ we defined additional evaluations.

4.3 Simplification of hypergeometric terms

In Gosper's algorithm we construct rational multiples of given hypergeometric terms. In the theory of the evaluation model we will eventually be able to prove that such constructed terms are evaluable. By the example of section 5.3.2 we know that in general we cannot evaluate these terms via classic evaluation. I invented a simplifier, which transfers the c-unevaluable terms back to c-evaluable ones.

I want to be more precise. We start with some term f. It has a domain $D_c(f)$. We are interested in the anti-difference g of f. Such an anti-difference is a rational multiple of f as we see later, and is supposed to be defined over $D_c(f)$. Using the evaluation model, we will be able to prove that g has almost the same domain as f. So for $D_{c,m}(f)$ we can m-evaluate the anti-difference; but not by classic means! We only know that $D_m(g) \approx D_m(f)$. Now we use the simplifier Ψ. It simplifies g such that the m-evaluation does not change but with the property that $D_c(\Psi(g)) \supseteq D_{c,m}(f)$. Where the evaluation of f is continuous we can enlarge the domain of validity to usually all of $D_c(f)$ by using the limit properties. In section 6.3.5, however, we present an interesting example that is not continuous.

Figure 4.2: The simplifier Ψ

The simplifier we want to construct has to enlarge the c-domain of a given term to all of the m-domain. This works only if we may assume that all but the rational part of the term is already c-evaluable. Actually we only have to get rid of denominators that evaluate to zero somewhere in the m-domain of the term. For instance $\frac{1}{x(x-1)!}$ is better written as $\frac{1}{x!}$. In the following we establish simplification rules, that can be carried out by some algorithm. Matching of the rules is considered to be part of the algorithm and is therefore also described in the following definition:

Definition 20 *Simplifier* Ψ

Given $f = (\varrho, f^-)$, let $\mathrm{den}(\varrho)$ be monic in $\mathbb{C}(x_{\nu+1}, \ldots x_\mu)[x_1, \ldots, x_\nu]$, then we define

$$\begin{aligned} \Psi(f) &:= \Psi_1(f) \cdot \Psi_2(f), \\ (\Psi_1(f), \Psi_2(f)) &:= \Psi_0\left(\frac{1}{\mathrm{den}(\varrho)}, \mathrm{num}(\varrho) \cdot t\right). \end{aligned}$$

The underlying simplifier Ψ_0 acts on a pair consisting of an inverted polynomial and a hypergeometric term. Ψ_0 of such a pair is the result of iteratively applying the simplification rules below, up to the point, where no rule fires. The first and second component of the result of Ψ_0 is denoted by Ψ_1 and Ψ_2, respectively.

A rule fires, if the pair can be written as the left-hand side of the rule. This is more than just rewriting, since in general a polynomial can be written as various products.

Assert in the following, that $\varrho, \varsigma, r, s \in \mathbb{C}(x_{\nu+1}, \ldots x_\mu)[x_1, \ldots, x_\nu]$ monic, and $j \neq 0$.

$$
\begin{aligned}
R_1 &: (\frac{1}{\varrho\varsigma}, \varrho \cdot g) \to (\frac{1}{\varrho}, g);\\
R_2 &: (\frac{1}{\varrho r}, \frac{1}{(jr-i-1)!} \cdot g) \to (\frac{1}{\varrho}, \frac{j\,\varphi(jr-i,i)}{(jr)!} \cdot g);\\
R_3 &: (\frac{1}{\varrho r}, (jr-i)_{\overline{s}} \cdot g) \to (\frac{1}{\varrho}, j\,\varphi(jr-i,i) \cdot (jr+1)_{\overline{s-i-1}} \cdot g);\\
R_4 &: (\frac{1}{\varrho r}, \frac{1}{(jr-i-s)_{\overline{s}}} \cdot g) \to (\frac{1}{\varrho}, \frac{j\,\varphi(jr-i,i)}{(jr-s+1)_{\overline{s}}} \cdot g);\\
R_5 &: (\frac{1}{\varrho r}, (jr-i-1+s)_{\underline{s}} \cdot g) \to (\frac{1}{\varrho}, j\,\varphi(jr-i,i)(jr+s)_{\underline{s}} \cdot g);\\
R_6 &: (\frac{1}{\varrho r}, \frac{1}{(jr-i-1)_{\underline{s}}} \cdot g) \to (\frac{1}{\varrho}, \frac{j\,\varphi(jr-i,i)}{(jr)_{\underline{s}}} \cdot g);\\
R_7 &: (\frac{1}{\varrho r}, \binom{s}{jr-i-1} \cdot g) \to (\frac{1}{\varrho}, \frac{j\,\varphi(jr-i,i)}{\varphi(s+1,i+1)} \binom{s+i+1}{jr} \cdot g);\\
R_8 &: (\frac{1}{\varrho r}, \binom{jr-i-1+s}{s} \cdot g) \to (\frac{1}{\varrho}, \frac{j\,\varphi(jr-i,i)}{\varphi(jr+s-i,i+1)} \binom{jr+s}{s} \cdot g);
\end{aligned}
$$

It is easy to see, that the iterative application of the rules above terminates, as in each step the degree of the first component is decreased.

The most important fact is, that the simplification process Ψ does not change the m-evaluation of the term. This can be seen, since both sides of any rule above have the same rational evaluation.

Lemma 2 (Invariant 1 of Ψ)

$$\rho(f, \vec{u}) = \rho(\Psi(f), \vec{u}).$$

Proof

The proof is given for rule R_2, only. Let r, j and i be fixed, such that R_2 may fire. Now, with fixed u and $c = \lfloor \mathrm{Re}(jr[\vec{u}]) \rfloor$ for the left-hand side of R_2 we have:

$$\rho\left(\frac{1}{r \cdot (j\,r-i-1)!}, \vec{u}\right) = \frac{1}{\Gamma(jr[\vec{u}]-c+1)\ \varphi(j\,r-c+1, c-i-1)\,r}.$$

For the right-hand side we have:

$$\rho\left(\frac{j\,\varphi(jr-i,i)}{(jr)!}, \vec{u}\right) = \frac{j\,\varphi(jr-i,i)}{\Gamma(jr[\vec{u}]-c+1)\ \varphi(j\,r-c+1,c)} =$$

$$= \frac{j\,\varphi(jr-i,i)}{\Gamma(jr[\vec{u}]-c+1)\,\varphi(j\,r-c+1,c-i-1)\,\varphi(j\,r-i,i)\,j\,r} =$$
$$= \frac{1}{\Gamma(jr[\vec{u}]-c+1)\,\varphi(j\,r-c+1,c-i-1)\,r}.$$

□

Lemma 3 (Invariant 2 of Ψ)

$$D_{c,m}(f) \subseteq D_{c,m}(\Psi_2(f)).$$

Proof

The technical proof is given for rule R_7, only. Let r, s, i and j be fixed, such that rule R_7 may fire. Now we fix $\vec{u} \in D_{c,m}\Big(\binom{s}{jr-i-1}\Big)$, then by proposition 8 we have, that $s[\vec{u}] \notin \mathbb{Z}^-$. Obviously we also have, that $(s+i+1)[\vec{u}] \notin \mathbb{Z}^-$. Therefore, $\vec{u}$ is part of both the c- and the m-domain of $\frac{j\varphi(jr-i,i)}{\varphi(s+1,i+1)}\binom{s+i+1}{j\,r}$.

□

Theorem 3 *Simplifier* Ψ

Let $g = \varrho \cdot f$, then

$$D_m(g) \cap D_{c,m}(f) \subseteq D_c(\Psi(g)).$$

Proof

Fix f, ϱ and g as above and take arbitrary but fixed $\vec{u} \in D_m(g) \cap D_{c,m}(f)$. Let $\frac{1}{\varrho'} := \Psi_1(g)$ and $f' := \Psi_2(g)$. (So we have $\Psi(g) = \frac{1}{\varrho'} \cdot f'$.)

1. By lemma 3 we know that $\vec{u} \in D_{c,m}(f')$. If $\varrho'[\vec{u}] \neq 0$, then we find that $\vec{u} \in D_c(\Psi(g))$, and we are done.

2. We assume that ϱ' has an irreducible factor d, such that $d[\vec{u}] = 0$. With this assumption we derive a contradiction:

   By lemma 2 we know, that $\vec{u} \in D_m(\Psi(g))$, therefore $\text{den}(\frac{1}{\varrho'}\rho(f',\vec{u}))[\vec{u}] \neq 0$. In order that d does not appear in the denominator of the rational evaluation above it has to appear in the numerator of $\rho(f',\vec{u})$.

   We analyze what kind of sub-terms of f' can cause the factor d in the numerator of the rational evaluation $\rho(f',\vec{u})$:

(a) The rational part contributes itself to the rational evaluation, therefore we find d as factor of the numerator of the rational evaluation, iff it is a factor of the numerator of the rational part.

(b) For a sub-term $s!$ the rational evaluation yields $\varphi(s-z+1,z)\Gamma(s-z+1)$, where $z = \lfloor \mathrm{Re}(s[\vec{u}]) \rfloor$. We know that $\Gamma(\_) \neq 0$ therefore we concentrate on the other part. If $z \geq 0$, then we have $s(s-1)\cdots(s-z+1)\Gamma(\cdot)$ and we can guarantee, that none of the factors evaluates to zero ($\mathrm{Re}(s[\vec{u}]) \geq z$). On the other hand, if $z < 0$, we have $\frac{\Gamma(\_)}{(s+1)(s+2)\cdots(s-z)}$ and we find that the denominator of this evaluates to 0, iff $s[\vec{u}] \in \mathbb{Z}^-$. Then more precisely $(s-z)[\vec{u}] = 0$. The factor $(s-z)$ is integer-linear in x_1 thus it contains d, iff for some $j \neq 0$ we have $s - z = jd$, i.e., $s = jd + z$.

We summarize: If $j \neq 0$ then d is a factor of $\mathrm{num}\rho(\frac{1}{(jd-i-1)!}, \vec{u})$, and there are no other factorials, that contribute d to the numerator of the rational evaluation.

(c) The rational evaluation of an exponential expression is free of x_1 and thus cannot contribute d.

(d) The rational evaluation of all the other sub-terms is defined via the rational evaluation of factorials, thus it can be read off, that

$$\begin{array}{cc} (jd-i)_{\overline{s}}, & \frac{1}{(jd-i-s)_{\overline{s}}}, \\ (jd-i-1+s)_{\underline{s}}, & \frac{1}{(jd-i-1)_{\underline{s}}}, \\ \binom{s}{jd-i-1}, & \binom{jd-i-1+s}{s} \end{array}$$

contribute d to the numerator of the rational evaluation, and no others.

Now you can see, that in any of the cases where f' contains a sub-term capable of causing the factor d in the rational evaluation, there is a rule in definition 20 that fires for $(\frac{1}{\varrho'}, f')$. This is a contradiction to the fact that $(\frac{1}{\varrho'}, f')$ is the final result of the iterative application of the rules.

□

Chapter 5

Indefinite sums of hypergeometric terms

5.1 What do we want to compute?

We want to find "closed forms" for indefinite sums of hypergeometric terms. If we give a R-sequence representation, we can make this more precise in terms of Symbolic Summation and follow concept 2. We have the ingredients and in the following we establish this interpretation.

Given f, we have an interpretation of f as a sequence of functions with domain $D_M(f)$. This sequence is an element of $(\mathbb{C}^{D_M(f)})^{\mathbb{N}}$, which forms a ring and therefore we have a R-sequence interpretation of f. Note that this representation can be used for a term g^+ only, iff $g^+ \in \mathrm{HT}_f^+$, as discussed in section 3.2.4 before.

Definition 21 *(Sequence-interpretation of a term)*

For f, we define:

$$\begin{aligned}
&\forall g^+ \in \mathrm{HT}_f^+ :\\
&\quad \phi_f(g^+) := (a_i)_{i\geq 0},\\
&\qquad a_i : D_M(f) \to \mathbb{C},\\
&\qquad\quad a_i(\vec{w}) := \mathrm{eval}(g^+, (i, \vec{w})).
\end{aligned}$$

$$\begin{aligned}
&\forall g^+ \in \mathrm{HT}_f^+ :\\
&\quad \phi_f(L_m g^+) := (a_0)_{i\geq 0},\\
&\qquad a_0 : D_M(f) \to \mathbb{C},\\
&\qquad\quad a_0(\vec{w}) := \mathrm{eval}(g^+, (0, \vec{w})).
\end{aligned}$$

Due to the fact that the sequence interpretation depends on the summand, we can discuss only one instance of the indefinite summation problem of HT^+ with input domain HT. Since the argument works in any instance we still say that we solve the indefinite summation problem of HT over HT^+.

The operators are given by $\mathtt{E} := E_1$, $\mathtt{L} := L_m$, $+, -$; to make it easier, the operator L_m was not defined, see discussion in section 5.4. Thanks to the evaluation model the homomorphic properties of these operators hold.

All we need now, is an algorithm that computes the anti-difference of f.

5.2 Gosper's algorithm computes anti-differences

For a given hypergeometric term f, Gosper's algorithm sets up a polynomial difference equation and solves it by computing a degree bound and using the method of undetermined coefficients. From the solution of the equation it constructs an anti-difference to f.

Given such a f, we specify how Gosper's equation looks like and furthermore how the algorithm constructs the anti-difference to f from the solution of the equation:

Definition 22 *(Gosper-Petkovšek canonical form)*

(p, q, r) is the *GP-representation* of $u(x_1) \in \mathbb{C}(\vec{x})$ in x_1, iff

1. $p(x_1), q(x_1), r(x_1) \in \mathbb{C}(x_2, \ldots, x_\mu)[x_1]$,
2. $u(x_1) = \frac{p(x_1+1)}{p(x_1)} \frac{q(x_1)}{r(x_1+1)}$.
3. $\forall i \neq 0$: $\gcd(q(x_1), r(x_1+i)) = 1$.
4. p, r monic.
5. $\gcd(p, q) = 1$, $\gcd(p, r) = 1$,

This representation is due to Gosper and exists for any rational function. Petkovšek refined it by supplying condition 4 and 5. In [Pet92] he proves that then this representation is unique. Furthermore it is computable by an algorithm.

Definition 23 *(Gosper's equation and solution)*

Given f, let (p, q, r) be the GP-representation of $Q_1 f$. Then for f we define:

- *Gosper's equation* to f is given by

  $$p(x_1) = q(x_1)y(x_1 + 1) - r(x_1)y(x_1),$$

  it contains one unknown, namely y.

- If Gosper's equation to f has a polynomial solution, i.e. $y \in \mathbb{C}(x_2, \ldots, x_\mu)[x_1]$, then by taking the polynomial solution, y, with minimal degree in x_1 we define the Gosper-term, g, to f by:

  $$g := \frac{r\,y}{p} \cdot f.$$

In [LPS93] it is shown that the solution to Gosper's equation to f is unique, if f is not rational.

5.2.1 Correctness

Proposition 11 *(Gosper's algorithm is correct)*

If g is the Gosper-term to f then $\phi_f(E_1 g - g) = \phi_f(f)$.

Proof

It suffices to prove $E_1 g - g = f$. We have

$$E_1 g - g = (Q_1 g - 1) \cdot g = (Q_1 \frac{r\,y}{p} \cdot Q_1 f - 1)\, \frac{r\,y}{p} \cdot f,$$

and complete the proof by

$$\begin{aligned}(Q_1 \frac{r\,y}{p} \cdot Q_1 f - 1)\, \frac{r\,y}{p} &= E_1 \frac{r\,y}{p} \cdot Q_1 f - \frac{r\,y}{p} = E_1 \frac{r\,y}{p} \cdot \frac{(E_1 p)\, q}{p\,(E_1 r)} - \frac{r\,y}{p} = \\ &= \frac{q E_1 y - r y}{p} = 1.\end{aligned}$$

□

You can see, to find an anti-difference of a hypergeometric term it suffices to compute one by Gosper's algorithm. If the algorithm does not yield a result, then in most cases this necessarily means that the anti-difference of the input-term cannot be expressed in HT^+, see section 5.2.3. That we cannot expect this property for all indefinite sums, is shown by the following example:

5.2.2 (Very) finite summation

Try to find a closed form for the indefinite sum

$$s(a) = \sum_{k=0}^{a} 12 \binom{3}{k} \binom{n}{k}$$

by applying Gosper's algorithm! I did, in Maple V Release 3, and got the following answer:

```
> sum( 36*binomial(3,k)*binomial(n,k),k=0..a);

                                a
                              -----
                               \
                                )   12 binomial(3, k) binomial(n, k)
                               /
                              -----
                              k = 0
```

But check out the hypergeometric term

$$g := n \left(-22 + 49\,k - 20\,k^2 - 12\,n + 3\,k\,n + 3\,k^2\,n - 2\,n^2 + 2\,k\,n^2 - k^2\,n^2\right) \binom{2}{k}.$$

We have that $\Delta_k g = E_k f$ and therefore also $s(a) = g[a] - g[0] + f[0]$. We cannot see this right away, but the computer reveals the facts:

```
In[1]:=
  f[k_] := 12 Binomial[3, k] Binomial[n, k]

In[2]:=
  g[k_] := n (-22 + 49 k - 20 k^2 - 12 n + 3 k n + 3 k^2 n -
            2 n^2 + 2 k n^2 - k^2 n^2) Binomial[2, k]

In[3]:=
  Table[ Factor[f[i+1]] == Factor[ g[i+1] - g[i]], {i, 0, 4}]

Out[3]=
  {True, True, True, True, True}

In[4]:=
  Table[ Factor[Sum[f[i], {i,0,j}]] == Factor[g[j]-g[0]+f[0]],
```

```
        {j, 0, 4}]
Out[4]=
  {True, True, True, True, True}
```

5.2.3 Symbolic sums with better support

If the domain of a hypergeometric term is large enough, then we are able to prove, that Gosper's algorithm finds all anti-differences to that term.

Definition 24 *(Sufficiently large sets)*

A set A is *sufficiently large* iff $\mathbb{C}^{\mu}\backslash A$ is contained by some finite union of $(\nu-1)$-dimensional lines.

Lemma 4 *(Checking equality of rational functions)*

Let A be sufficiently large, such that for fixed ϱ, ς we have $D_m(\varrho), D_m(\varsigma) \subseteq A$, then:

$$(\forall \vec{u} \in A\colon\ \varrho[\vec{u}] = \varsigma[\vec{u}]) \Rightarrow \varrho = \varsigma.$$

Proof

Left to the reader.

Definition 25 *(Good support)*

A term f has *good support* iff $\{\vec{u} \in \mathbb{N} \times D_M(f) \mid \mathrm{eval}(f, \vec{u}) \neq 0\}$ is sufficiently large.

Note that if f has good support, then its m-domain is a set of values that can be used to check equality in $\mathbb{C}(\vec{x})$.

We are looking for a representation $g^+ \in \mathrm{HT}_f^+$ of the anti-difference of the sequence $\phi_f(f)$ in HT_f^+. Let us write g^+ as finite sum of elements of hypergeometric terms, $g^+ = g_1 + g_2 + \cdots + g_m$, where none of $g_1, \ldots, g_m$ has a zero sequence as interpretation. We are only interested in those g^+, where all of the hypergeometric constitutes are m-evaluable over the m-domain of f. If there is an anti-difference using other constitutes, then we are not able to m-evaluate it.

We have an algorithm, that looks for anti-differences of f in HT_f but none that looks in HT_f^+. Therefore we need the following proposition:

Proposition 12 *(Anti-differences are simple)*

If f has good support, and $\phi_f(f) = \phi_f(E_1 g^+ - g^+)$, with $g^+ = g_1 + g_2 + \cdots + g_n$, $g_1, \ldots, g_n \in \mathrm{HT}_f$, then there exist $g \in \mathrm{HT}_f$ and a sufficiently large set D such that

$$\forall \vec{u} \in D\colon \ \text{eval}(f, \vec{u}) = \text{eval}(E_1 g, \vec{u}) - \text{eval}(g, \vec{u}).$$

Proof

Fix $f, g^+, g_1 + g_2 + \cdots + g_n$, as in the proposition. Then we want to prove that there is at least one constitute g_i, for which we have $\text{eval}(E_1 g_i - g_i, \vec{u}) = \text{eval}(f, \vec{u})$ over a domain D which is sufficiently large.

Assume that none of the constitutes is such an anti-difference of f, then we derive a contradiction: Since $\phi_f(f)$ satisfies a recurrence relation of order one, we know that the anti-difference $\phi_f(g^+)$ satisfies such a relation of order two. Let p_0, p_1 and p_2 be such that $\phi_f(p_0 g^+ + p_1 E_1 g^+ + p_2 E_1^2 g^+) = 0$.

Let $D := \mathbb{N} \times D_M(f)$, then we have

$$\forall \vec{u} \in \mathbb{N} \times D_M(f)\colon \ \text{eval}(p_0 g^+ + p_1 E_1 g^+ + p_2 E_1^2 g^+, \vec{u}) = 0.$$

There exist rational functions $\varrho_1, \ldots, \varrho_m$, such that

$$\forall \vec{u} \in \mathbb{N} \times D_M(f)\colon \ \text{eval}(\varrho_1 g_1 + \varrho_2 g_2 + \cdots + \varrho_m g_m, \vec{u}) = 0.$$

If $\varrho_1 = 0$ then we have that g_1 satisfies the recurrence relation for the anti-differences of f, i.e. g_1 is an anti-difference of f. We conclude that $\varrho_1 \neq 0$. From that we find

$$\forall \vec{u} \in D'\colon \ \text{eval}(g_1, \vec{u}) = \text{eval}(\frac{\varrho_2}{\varrho_1} g_2 + \cdots + \frac{\varrho_m}{\varrho_1} g_m, \vec{u}),$$

where $D' := \mathbb{N} \times D_M(f) \cap D_m(\frac{1}{\varrho_1})$. As you see, we can find g_1^+ by replacing g_1 by a linear combination of all but the first constitutes.

By iteratively applying the same argument we find a g in HT with the property that $\text{eval}(E_1 g - g, \vec{u}) = \text{eval}(f, \vec{u})$ over D'', which is still sufficiently large.

□

I have to mention that the basic idea to this argument can be found in [Pet92]. There is a big difference, though, since there sequences are considered to be equal as soon as almost all sequence elements are equal. As we showed above the idea carries over to our refined sequence interpretation.

Proposition 13 *(Gosper's algorithm finds all anti-differences)*

If f has good support and Gosper's equation to f has no polynomial solution, then there are no $g_1, \ldots g_m \in \text{HT}_f$, such that $\phi_f(E_1 g^+ - g^+) = \phi_f(E_1 f)$, where $g^+ := g_1 + \cdots + g_m$.

Proof

Assume that $\phi_f(E_1 g^+ - g^+) = \phi_f(f)$ then, by proposition 12, we have a g and a domain D, such that

$$\forall \vec{u} \in D\colon\ \text{eval}(E_1 g - g, \vec{u}) = \text{eval}(f, \vec{u}),$$

where D is sufficiently large.

Since f has good support the intersection D' of D and the support of f is still sufficiently large.

For a sufficiently large D'', we have

$$\begin{aligned}&\forall \vec{u} \in D''\colon\\ &\qquad \text{eval}(E_1 g - g, \vec{u}) = \text{eval}(f, \vec{u}) \neq 0,\\ &\qquad \text{eval}(E_1(E_1 g - g), \vec{u}) = \text{eval}(E_1 f, \vec{u}) \neq 0,\end{aligned}$$

where equality in $\mathbb{C}(\vec{x})$ can still be decided over D''. By Property 1 and 2 of the evaluation model, compare proposition 7, we have

$$\forall \vec{u} \in D''\colon\ \text{eval}(Q_1(E_1 g - g), \vec{u}] = \text{eval}(Q_1 f, \vec{u}).$$

Since D'' is sufficiently large, we have $Q_1(E_1 g - g) = Q_1 f$ or equivalently

$$Q_1 g \cdot Q_1(Q_1 g - 1) = Q_1 f.$$

Let $(y, \bar{q}, \bar{r})$ be the GP-representation of $Q_1 g$, then we have

$$\frac{(E_1 y)\,\bar{q}}{y\,(E_1 \bar{r})} Q_1 \frac{(E_1 y)\,\bar{q} - y\,(E_1 \bar{r})}{y\,(E_1 \bar{r})} = Q_1 f.$$

After some simplification we find

$$\frac{\bar{q}}{E_1^2 \bar{r}} Q_1(\bar{q}(E_1 y) - (E_1 \bar{r}) y) = Q_1 f.$$

Observe, that $\gcd(\bar{q},\ \bar{q} E_1 y - \bar{r} y) = \gcd(\bar{r},\ \bar{q} E_1 y - \bar{r} y) = 1$. Therefore the left-hand side is already written in GP-form, and we find with c free of x_1 and (p, q, r) as GP-representation of $Q_1 f$, that

$$E_1 \bar{r} = r, \bar{q} = q, q E_1 y - r y = c\, p.$$

This implies that Gosper's equation has a polynomial solution.

□

5.2.4 Evaluation domain of Gosper-terms

We have an algorithm that finds (all) anti-differences for f over HT_f^+. So far, we did not check, whether the Gosper-term computed is really an element of HT_f^+. In other words is the anti-difference found by Gosper's algorithm a sequence of functions, which have domains larger than $D_M(f)$? The next proposition is born to answer this subtle question.

Proposition 14 *(Gosper-terms are evaluable)*

Let g be the Gosper-term to the given f, let y be the corresponding solution of Gosper's equation to f and (p, q, r) the GP-representation of $Q_1 f$. Then

$$D_M(f) \cap D_M(y) \subseteq D_M(g).$$

Proof

Assume that $\vec{w} \in (D_M(f) \cap D_M(y))$. We know, that $g = \frac{r\,y}{p} \cdot f$. Because p is monic, we can choose an n such that $(n, \vec{w}) \in D_m(\frac{1}{p})$ and thus $(n, \vec{w}) \in D_m(g)$.

With n as a basis we use induction to prove $\forall i:\ (i, \vec{w}) \in D_m(g)$. The induction leads from n in both directions towards 0 and ∞:

1. The forward step:

   Let n be such that $(n, \vec{w}) \in D_m(g)$. Then we have $(n, \vec{w}) \in D_m(g + f)$ and since $E_1 g = g + f$, we have $(n, \vec{w}) \in D_m(E_1 g)$. By property 5 in proposition 7 we have $(n + 1, \vec{w}) \in D_m(g)$.

2. Stepping backwards:

   Let $n > 0$ be such that $(y, \vec{w}) \in D_m(g)$. Then we have $(n - 1, \vec{w}) \in D_m(f)$. By property 5 in proposition 7 we also have $(n - 1, \vec{w}) \in D_m(E_1 g)$ and since $g = E_1 g - f$, we have $(n - 1, \vec{w}) \in D_m(g)$.

This completes induction and proof.

□

Actually, an answer with an exception $(D_M(f) \cap D_M(y)$ instead of $D_M(f)$ is no solution to the anti-difference problem for a given f over HT_f^+. It is a solution for the problem using an even smaller domain. This domain is not known before the execution of the algorithm, therefore we may be tempted to say, that this is too easy if a decision algorithm itself sets up the domain where its decisions are valid.

Is there another algorithm, that finds more solutions by setting up its domain of validity? The answer is no, as long as these domain restrictions do not destroy the good support property. If you want to find an algorithm that computes anti-differences, where Gosper's algorithm fails, then you have to make domain restrictions such that the terms loose the good support property, compare section 5.2.2. Note that the restrictions set up by Gosper's algorithm do not spoil the good support property.

5.3 Examples

5.3.1 Harmonic Numbers

We try to find a closed form for the harmonic numbers:

$$H_m := \sum_{k=0}^{m} \frac{1}{k+1} =?$$

We try to solve the related indefinite problem by Gosper's algorithm and find no solution.

If we use a variation of the problem:

$$S(m,n) := \sum_{k=0}^{m} \frac{k!}{(k+n)!},$$

we can use Gosper's algorithm!

```
In[39]:= Gosper[1/(k+1), {k,0,m}]
Out[39]= {}

In[40]:= Gosper[k!/(k+n)!, {k,0,m}]
If 'm' is a natural number, then:

              (1 - n) k!                      n      (1 + m) m!
Out[40]= {SUM[----------, {k, 0, m}] == -(--) + ----------}
               (k + n)!                       n!      (m + n)!
```

So for $n \neq 1$, we have that

$$S(m,n) = \frac{-(m+1)!(n-1)! + (n+m)!}{(n-1)(n-1)!(n+m)!}.$$

Note that the closed form cannot be evaluated for $n = 1$. Of course, the reason for this is, that the solution to Gosper's equation in this example is given by $-\frac{1}{n-1}$. Our theory suggests that this is an evaluation point where things can go wrong and therefore in the Paule/Schorn implementation $(1-n)$ is multiplied to the summand.

We go further by observing that $S(m, n)$ is continuous in n and therefore we have

$$H_m = S(m, 1) = \lim_{n \to 1} \frac{-(m+1)!(n-1)! + (n+m)!}{(n-1)(n-1)!(n+m)!}.$$

This limit can be converted:

```
In[42]:= Limit[(-(m+1)!(n-1)!+(n+m)!) / ((n-1)(n-1)!(n+m)!), n-> 1]

Out[42]= EulerGamma + PolyGamma[0, 2 + m]
```

We have proven a result that can be found for instance in [SLL89]. The result can be found more conveniently by a rational summation algorithm, which for instance is implemented in Maple:

```
> sum(1/(k+1), k=0..m);
                              Psi(2 + m) + gamma
```

5.3.2 Applying the simplifier Ψ

To find a closed form for

$$s(m, n) = \sum_{k=0}^{m} \frac{-k(n+1) + n^2 + 2(n+1)}{(k-1)!} \binom{k}{n},$$

we apply Gosper's algorithm:

```
> sum( (-k*(n+1)+n^2+2*(n+1))*binomial(k,n)/(k-1)!, k);
                binomial(k, n) (k - n) (- 1 + k n + k)
                --------------------------------------
                          (k - 1) (k - 2)! k
```

Surprisingly Maple does not give a rational multiple of f as an answer. Since in this example the GP-form of the shift quotient is given by $(k(-k(n+1) + n^2 + 2(n+1)), 1, k-n)$ and the solution to the Gosper-equation is $(-1 + kn + k)$ I expected

$$g := \frac{(k-n)(kn+k-1)}{k(k-1)!}\binom{k}{n}.$$

Anyways we solved the summation problem:

$$s(m,n) = \text{eval}(g, m+1, n) - \text{eval}(g, 0, n).$$

Since we do not know the specific values for m and n, we are not able to compute the evaluations. Therefore we try to use classic evaluation instead. If $g[m+1,n]$ is defined, then it is equal to $\text{eval}(g,(m+1,n))$, anyhow. In proposition 14 we proved that $(m+1,n)$ belongs to the domain of eval. And indeed in this example we have $g[m+1,n] = \frac{(m-n+1)(m+mn+n)}{(m+1)m!}\binom{m+1}{n}$, so also $\text{eval}(g,n+1,n) = \frac{(m-n+1)(m+mn+n)}{(m+1)m!}\binom{m+1}{n}$. With $\text{eval}(g,(0,n))$, the same trick does not work, because $g[0,n]$ is not defined.

It is time to use the simplifier Ψ to construct an evaluable term. We know that the original summand f is c- and m-evaluable at $(0,n)$ and g, a rational multiple of f, is m-evaluable at $(0,n)$. Therefore we know that $\Psi(g)$ is c- and m-evaluable at $(0,n)$! So lets compute it.

```
In[2]:=
SimplifierPsi[(k-n)(k n + k -1)/k Binomial[k, n]/ (k-1)!, k]
Out[2]=
   (-k + n) (-1 + k + k n) Binomial[k, n]
 -(--------------------------------------)
                     k!
```

Now, denoting the result above as g', we find that

$$\text{eval}(g,0,n) = \text{eval}(g',0,n) = g'[0,n] = n\binom{0}{n} = 0.$$

So we have

$$S(m,n) = \frac{(m-n+1)(m+mn+n)}{(m+1)m!}\binom{m+1}{n} = \frac{(m+mn+n)}{m!}\binom{m}{n}.$$

The Paule/Schorn implementation does most of this automatically:

```
In[4]:= Gosper[(-k(n+1)+n^2+2(n+1)) Binomial[k,n]/(k-1)!, {k,0,n}]
If 'm' is a natural number, then:
```

```
                              (m + n + m n) Binomial[m, n]
Out[4]= {-(n Binomial[0, n]) + ----------------------------}
                                          m!
```

Also Maple is capable of producing a similar result automatically:

```
> sum((-k*(n+1)+n^2+2*(n+1))*binomial(k,n)/(k-1)!,k=0..m);
binomial(m + 1, n) (m + 1 - n) ((m + 1) n + m)            1
---------------------------------------------- + --------------------
              m (m - 1)! (m + 1)                 n GAMMA(n) GAMMA(-n)
```

5.4 Summary for real life

Given f, Gosper's algorithm finds a representation in HT^+ of the indefinite sum over f, if it exists. There are two foot-traps:

1. This "closed form" for the indefinite sum must be interpreted as sequence of functions with domain $D_M(f) \cap D_M(y)$, where y is the solution of Gosper's equation to f. The user believes that the result is valid on $D_C(f)$.
2. The evaluation of the anti-difference (the Gosper-term) is guaranteed to work only by means of eval.

Since you cannot ask every user of Gosper's algorithm to be aware of this, it is your task to make your implementation safe.

Half of the first problem can be avoided, if you multiply the denominator of y to the summand as this was done in the first example above. Then we get a solution that is valid on $D_M(f)$.

The second half is more difficult, it has to take care of the values in $D_C(f) \backslash D_M(f)$. As we know, this set of values contains these choices of parameters, where the upper entries of a binomial become a negative integer (and similar restrictions on falling and raising factorials). If the sum in question is continuous in some variable that appears in the upper parameter of a binomial, then we can approach the values ruled out from $D_M(f)$ by this binomial by a limit. Continuity of the sum can be determined automatically and therefore the algorithm can output the restriction for the remaining binomials.

The second problem can be attacked by applying the simplifier Ψ to the Gosper-term. We have to express $L_m g$ and $E_1 g$ such that the user can c-evaluate these terms over $\mathbb{N} \times D_C(f)$. If we find a representation g' for g that evaluates over

$\mathbb{N} \times D_{C,M}(f)$, then we can express $L_m g$ by $L_c g'$ which is nothing but $g'[(0, \vec{x_2})]$. Since g is a rational multiple of f which is c- and m-evaluable over $\mathbb{N} \times D_{C,M}(f)$ and $\mathbb{N} \times D_{C,M}(f) \subseteq D_m(g)$ such a g' can be computed by $\Psi(g)$. Summarizing, for $\vec{u} \in \mathbb{N} \times D_{C,M}(f)$ we have

$$\mathrm{eval}(L_m g, \vec{u}) = \mathrm{eval}(L_m \Psi(g), \vec{u}) = (\Psi(g)[0, \vec{x_2}])[\vec{u}].$$

Therefore we represent $L_m g$ by $\Psi(g)[0, \vec{x_2}]$.

Similarly we have

$$\mathrm{eval}(E_1 g, \vec{u}) = \mathrm{eval}(\Psi(E_1 g), \vec{u}) = (\Psi(E_1 g)[\vec{x}])[\vec{u}],$$

and therefore represent $E_1 g$ by $\Psi(E_1 g)[\vec{x}]$. These representations are at least evaluable over $\mathbb{N} \times D_{C,M}(f)$.

Remark For f with Gosper-term $\frac{r\,y}{p} f$ we have

$$E_1 g = E_1 \frac{r\,y}{p} \cdot E_1 f = E_1 \frac{r\,y}{p} \cdot \frac{(E_1 p)\,q}{p\,(E_1 r)} \cdot f = \frac{q\,E_1 y}{p} \cdot f.$$

Chapter 6

Definite sums of hypergeometric terms

In the previous chapters we discussed indefinite sums in every detail. These sums have the nice property that the summation bounds are independent of the summands. This is already different looking at the Binomial Theorem

$$\sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k} = (x+y)^n,$$

where the n occurs both in the summand and the summation bound. Thus, the Δ-operator applied on this sum does not yield the shifted summand.

We could go for an interpretation in the sequence world, again. The following example though, suggests to avoid this:

$$\sum_{k=0}^{n} (n-k)!(n-k) = (n+1)! - 1.$$

You can see, that the summand is undefined as soon as k is larger than n. However, due to the upper summation bound we have a guarantee, that the sum can be evaluated for any choice of parameters. In other words, we have to introduce a summation range in dependence of the parameters. We do this in a straight forward manner, rather than in a general setting of sequences.

6.1 Basic notions of definite summation

We give a definition of definite summation:

Definition 26 *(Definite hypergeometric sum)*

If $b_l, b_u \in \mathrm{IL}$, free of x_1, then the tuple (f, b_l, b_u) denotes a *definite (hypergeometric) sum*; instead of (f, b_l, b_u) we also write $\sum_{x_1=b_l}^{b_u} f$.

Note, that indefinite sums are just a special subclass of the definite sums. As an example of a definite sum we have:

$$\left((-1)^{x_1}\binom{x_2}{x_1}, 0, x_2\right), \text{ or more familiar } \sum_{x_1=0}^{x_2}(-1)^{x_1}\binom{x_2}{x_1}.$$

If we choose x_2 to be $1/2$, then this problem does not make sense. The difference between the upper and lower bound has to be a natural number. Furthermore, the summand must be evaluable throughout the range of summation; thus in this case we have $x_2 \in \mathbb{N}$. We will now define the domain of a definite summation problem. For a fixed value out of the domain the definite sum reduces to a finite sum over complex numbers.

Definition 27 *(Interpretation of definite sums)*

1. For the summation-problem $P = (f, b_l, b_u)$ we define a *domain* for both evaluation methods:

$$\begin{aligned} D_C(P) &:= \left\{ \vec{w} \;\middle|\; \begin{array}{l} b_u[\vec{w}] - b_l[\vec{w}] \in \mathbb{N}, \\ \forall u \in \{b_l[\vec{w}], \ldots, b_u[\vec{w}]\}\colon\ (u, \vec{w}) \in D_c(f) \end{array} \right\}, \\ D_M(P) &:= \left\{ \vec{w} \;\middle|\; \begin{array}{l} b_u[\vec{w}] - b_l[\vec{w}] \in \mathbb{N}, \\ \forall u \in \{b_l[\vec{w}], \ldots, b_u[\vec{w}]\}\colon\ (u, \vec{w}) \in D_m(f) \end{array} \right\}. \end{aligned}$$

2. The *evaluation* of a definite sum $P = (f, b_l, b_u)$ *at* $\vec{w}$ is defined by:

$$\forall \vec{w} \in D_C(P)\colon \qquad P[\vec{w}] := \sum_{k=b_l[\vec{w}]}^{b_u[\vec{w}]} f[k, \vec{w}],$$

$$\forall \vec{w} \in D_M(P)\colon \qquad \mathrm{eval}(P, \vec{w}) := \sum_{k=b_l[\vec{w}]}^{b_u[\vec{w}]} \mathrm{eval}(f, k, \vec{w}).$$

3. For the definite sum $P = (f, b_l, b_u)$ we define two summation *ranges* at $\vec{w}$:

$$\mathrm{range}(P, \vec{w}) := \{b_l[\vec{w}], \ldots, b_u[\vec{w}]\}.$$

   The *max-range* at $\vec{w}$ is defined only if $\vec{w} \in D_M(P)$ and is the range extended in both directions as long as f remains m-evaluable.

In the example, $P = \sum_{x_1=0}^{x_2}(-1)^{x_1}\binom{x_2}{x_1}$, we find that $D_M(P) = D_C(P) = \mathbb{N}$. Therefore the evaluation of P at 5 is defined and it is given by

$$\sum_{k=0}^{5}(-1)^k\binom{5}{k} = 0.$$

The ranges are:

$$\begin{aligned}\text{range}(P,5) &= \{0,1,2,3,4,5\}\\ \text{max-range}(P,5) &= \mathbb{N}.\end{aligned}$$

Assume we have a definite sum P. We take an evaluation point out of $D_C(P)$ and find the evaluation $P[\vec{w}]$. Now a representation of the definite sum must have the same evaluation at $\vec{w}$. At the best this is true for all evaluation points of $D_C(P)$. In the following we will also consider representations that evaluate correctly only over a subset of $D_C(P)$.

Definition 28 *(Representation of definite sums)*

g^+ is a *representation* of the definite sum $P = (f, b_l, b_u)$ over $D \subseteq D_C(P)$, iff for any $\vec{w} \in D$, the following holds:

- The representation is evaluable:

$$\text{range}(P,\vec{w}) \times \{\vec{w}\} \subseteq D_C(g^+).$$

- The c-evaluation of P at $\vec{w}$ is equal to the c-evaluation of the representation:

$$P[\vec{w}] = g^+[\vec{w}].$$

6.2 Using indefinite summation

Prove the Binomial Theorem

$$\sum_{k=0}^{n}\binom{n}{k}x^k y^{n-k} = (x+y)^n,$$

by showing that $(x+y)^n$ is a representation of the definite sum to the left! We can try do do this by finding a representation of the corresponding indefinite problem by means of Gosper's algorithm. It is given by

$$\sum_{k=0}^{a} \binom{n}{k} x^k y^{n-k} = ?$$

As a first drawback we find out that in opposition to the definite sum, the indefinite one does not have a hypergeometric representation. This is not very surprising, as the indefinite problem is much more general and the nice representation for a definite sum may depend on the additional information stored in the summation bounds. In other words, we will need further methods to compute representations of definite sums, see section 6.3.

6.2.1 The starting condition of a double sum

In the computer proof of

$$\sum_{i=0}^{n} \sum_{j=0}^{m} (-1)^{i+j} \binom{n}{i} \binom{m}{j} \binom{i+j}{j} = \delta_{m,n}$$

at the initial value $n = 0$ one has to check a single sum:

$$\sum_{j=0}^{m} (-1)^j \binom{m}{j} = ?$$

We try to solve the related indefinite problem by Gosper's algorithm and (using Maple V) we find:

```
> sum((-1)^j*binomial(m,j), j=0..n);
                                                    n
                       binomial(m, n + 1) (n + 1) (-1)
                       --------------------------------
                                      m
```

Now setting $n := m$ we find that

$$\forall m \neq 0\colon \sum_{j=0}^{m} (-1)^j \binom{m}{j} = 0.$$

Maple can do this in a single step, too,

```
> sum((-1)^j*binomial(m,j), j=0..m);
                                      0
```

but the answer is wrong! The right hand side of the sum is given by $(m = 0)$. Let us investigate this example further and try the implementation of Peter Paule and Markus Schorn.

```
In[1]:=
<< fastZeil.m
Fast Zeilberger by Peter Paule and Markus Schorn. (V 2.0)
Systembreaker = ENullspace

In[2]:=
Gosper[(-1)^j Binomial[m,j], {j, 0, m}]
If `m' is a natural number, then:

Out[2]=
           j
{SUM[-((-1)  m Binomial[m, j]), {j, 0, m}] == 0}
```

Here you can see that the summand was multiplied by m, thus also for $m = 0$ we have a correct evaluation. So it can be discovered automatically that something is different in the case $m = 0$. Of course, the reason is, that in this example the solution of the Gosper-equation is given by $y := -\frac{1}{m}$ and thus $\{0\} = D_M((-1)^j \binom{m}{j}) \backslash D_M(y)$.

6.2.2 Evaluation domain of Gosper-terms

In section 5.2.4 we proved that in general we can evaluate Gosper-terms. The proof depends on the fact that the summation-range (of an indefinite sum) may be extended arbitrarily. For definite sums, this is definitely not the case, therefore proposition 14 must be slightly modified:

Proposition 15 *(Evaluation domain of Gosper-terms)*

Let $P = (f, b_l, b_u)$ be a definite sum, $g = \frac{r\,y}{p} \cdot f$ is the Gosper-term to f then we have

$$D_M(f) \cap D_M(y) \cap \{\vec{w} \mid \deg_{x_1}(p) < |\text{max-range}(P, \vec{w})|\} \subseteq D_M(g).$$

You see, that we had to add another restriction, that guarantees that p evaluates non zero at least once. That this condition is really necessary is shown by the following example.

A constructed example

I constructed an example where the polynomial p causes a problem. We try to find a closed form of the following sum:

$$S_{a,b} = \sum_{k=a}^{b} \frac{(-1)^k k!}{k-1}$$

Here is what the algorithm computes:

```
> sum((-1)^k*k!/(k-1), k=a..b);

                         b                    (a - 1)
                 b! (-1)     (a - 1)! (-1)
                 -------- - --------------------
                     b                a - 1
```

The degree of p is one and the max-range of the sum at $\vec{w} = (0,0)$ is of length one. Therefore theory does not guarantee that we can evaluate the result; and indeed $S_{0,0}$ can by no means be calculated by the formula above. Transformations do not help! Theory has predicted (or at least not ruled out) this breakdown.

Note, that the max-range of the sum $(\frac{(-1)^k k!}{k-1}, \{k, 2, b\})$ at any $\vec{w} := (a_0, b_0)$, where $a_0, b_0 \in \mathbb{N}, a_0 \geq 2, b_0 \geq a_0$ is given by $\mathbb{N} \backslash \{0, 1\}$. Thus the formula found by Gosper's algorithm can be evaluated, for instance, for the choice $a = 2, b \geq a$.

6.2.3 Expressing the result by terms

Given the definite sum $P = (f, b_l, b_u)$ with g as Gosper-term to f, we have that over a suitable domain

$$\text{eval}(P, \vec{w}) = \text{eval}(E_1 g, (b_u[\vec{w}], \vec{w})) - \text{eval}(g, (b_l[\vec{w}], \vec{w})).$$

This follows from the properties of the evaluation-model together with the fact that $E_1 g - g = f$. Now the question is, how can we express $\text{eval}(g, (b_u[\vec{w}] + 1, \vec{w}))$ as $\text{eval}(g', \vec{w})$? Somehow this is a partial evaluation of a hypergeometric term. In other words, you can say that g' is the composition of g with $b_u + 1$ in x_1. We do not have a method to compute such a composition. Indeed it would be possible to define one. (Here is a warning: It is not totally straight forward!) But in fact we want to give a link back to classic evaluation anyhow, so we can avoid introducing this method.

We look for a representation of an indefinite sum over the intersection of $D_{C,M}(P)$ and the restrictions of proposition 15, then over that domain we have:

$$\begin{aligned} P[\vec{w}] &= \text{eval}(P, \vec{w}) = \text{eval}(E_1 g, b_u[\vec{w}], \vec{w}) - \text{eval}(g, b_l[\vec{w}], \vec{w}) = \\ &= \text{eval}(\Psi(E_1 g), b_u[\vec{w}], \vec{w}) - \text{eval}(\Psi(g), b_l[\vec{w}], \vec{w}) = \\ &= \Psi(E_1 g)[b_u[\vec{w}], \vec{w}] - \Psi(g)[b_l[\vec{w}], \vec{w}]. \end{aligned}$$

Now, we can compute the composition of $\Psi(E_1 g)$ and b_u in x_1, since c-evaluation is defined by evaluating sub-term by sub-term, and there the substitution lemma known from logics (see [EFT84]) tells us that composition of functions can be computed by replacement. So we have

$$P[\vec{w}] = ((\Psi(E_1 g) \circ b_u) - \Psi(g) \circ b_l)[\vec{w}].$$

We summarize in a proposition:

Proposition 16 *(Representation of definite sums)*

Given the indefinite sum $P = (f, b_l, b_u)$, where $g = \frac{r\,y}{p} \cdot f$ is the Gosper-term to f then for

$$P' := (\text{den}(y) \cdot f, b_l, b_u),\ g' := \text{den}(y) \cdot g,$$

we have that $\Psi(E_1 g') \circ b_u - \Psi(g') \circ b_l$ is a solution to P' over $D_{C,M}(P') \cap \{\vec{w} \mid \deg_{x_1}(p) < |\text{max-range}(P', \vec{w})|\}$.

Note that again the domain of the solution can usually be enlarged to $D_C \cap \ldots$ by automatically applying the limit properties.

6.3 Finding more representations of definite sums

Instead of constructing a hypergeometric term as a representation of a definite sum, D. Zeilberger had the idea to automatically look for recurrences for these sums. A recurrence relation together with some initial values is a perfect representation for proving identities: Find a recurrence satisfied by both sides of an identity and check enough initial values!

Using Marko Petkovšek's algorithm HYPER (see [Pet92]) you can find all hypergeometric solutions to a recurrence relation. Both methods together form an algorithm to find hypergeometric representations for definite sums!

To talk about Zeilberger's creative telescoping algorithm we need a good notation for recurrences.

6.3.1 Operators and recurrences

Definition 29 *(Set of operators)*

We define the multiplicative group of operator monomials in $E_1, \ldots, E_m$:

$$\Theta(E_1, \ldots, E_m) := \langle \{E_1, \ldots, E_m\} \rangle.$$

Upon this we define the set of operators in $E_1, \ldots, E_m$ with rational coefficients:

$$\Theta(x_1, \ldots, x_n; E_1, \ldots, E_m) :=$$
$$\left\{ \sum_{X \in \Theta(E_1, \ldots, E_m)} p_X X \mid p_X \in \mathbb{C}(x_1, \ldots, x_n), p_X = 0 \text{ for almost all } X, \right\}.$$

$\Theta(E_1, \ldots, E_\nu)$ is the largest set of shift operator monomials for hypergeometric terms and $\Theta(x_1, \ldots, x_\mu; E_1, \ldots, E_\nu)$ the largest set of rational shift operators on HT.

An operator monomial can act on a hypergeometric term ($E_1@f := E_1 f$). It is just the regular action of the operators $E_1, \ldots, E_\nu$. Note that with proposition 4 we have that $E_i@(E_j@f) = E_j@(E_i@f)$ and therefore the action of an operator monomial $(E_1E_2)@f$ is well defined.

On the set of operators, we introduce two operations (operations on operators!) $+$ and $\cdot$, yielding an Ore-ring of operators, see, e.g., [BP93].

Definition 30 *(Operations on operators)*

With $\sum_X p_X X, \sum_X q_X X \in \Theta(x_1, \ldots, x_\mu; E_1, \ldots, E_\nu)$ we define:

$$\sum_X p_X X + \sum_X q_X X := \sum_X (p_X + q_X) X.$$
$$\left(\sum_Y p_Y Y\right)\left(\sum_Z q_Z Z\right) := \sum_X \left(\sum_{YZ=X} p_Y (Y@q_Z)\right) X.$$

We define the order of an operator $O = \sum_{i=0}^{n} p_i X_i \in \Theta(\_\_;\_\_)$ to be n. Write $\mathrm{ord}(O) = n$.

These operators may be applied on hypergeometric terms.

Definition 31 *Application of operators*

For $\sum_i p_i X_i \in \Theta(x_1, \ldots, x_\mu; E_1, \ldots, E_\nu)$ we define

$$\left(\sum_i p_i X_i\right) @f := \sum_i p_i (X_i @f).$$

The proof of

$$\forall A, B \in \Theta(x_1, \ldots, x_\mu; E_1, \ldots, E_\nu), f \in \mathrm{HT}\colon\ (AB)f = A(Bf)$$

is left to the reader.

Very often one uses $E_1 f$ instead of $E_1 @ f$, which is fine. At the same time the abbreviation O_1O_2 for $O_1 \cdot O_2$ is very common. Now if an operator acts on a rational function the notation is ambiguous. For instance $E_1x_1 = x_1+1$ or $E_1x_1 = (x_1+1)E_1$. In such cases you have to write $E_1 @ x_1 = x_1 + 1$ and $E_1 \cdot x_1 = (x_1 + 1)E_1$.

We have operators that act on hypergeometric terms. To express the impact of such an operator on the functions represented by the terms we need to define operators acting on functions. The functions have to be shifted in specific arguments. To do this we use the operators $E_{\#1}, E_{\#2}, \ldots$, shifting the first, second, and so forth argument. If a function is given by some term, lets say $a(k, y) := sin(ky)$, then we will also use $E_k := E_{\#1}$. As one more remark, such a shift operator also shifts the domain of a function. Analogous to above we build up an Ore-ring of operators.

Definition 32 *(Application of operators on functions)*

With $D \subseteq \mathbb{C}^\mu$, $a : D \to \mathbb{C}$, and $\sum_X p_X X \in \Theta(x_1, \ldots, x_\mu; E_{\#1}, \ldots, E_{\#\nu})$ we define

$$\begin{aligned}
&E_{\#1}D := \{(u-1, \vec{w}) | (u, \vec{w}) \in D\},\\
&E_{\#1}a := b,\\
&\qquad b : E_{\#1}D \to \mathbb{C},\\
&\qquad\qquad b(u, \vec{w}) := a(u+1, \vec{w});\\
&\left(\sum_X p_X X\right) a := b,\\
&\qquad b : D' \to \mathbb{C},\\
&\qquad\qquad b(\vec{u}) := \sum_X p_X[\vec{u}](Xa)(\vec{u}),\\
&\qquad\qquad D' := \bigcap_X (XD \cap D_c(p_X)).
\end{aligned}$$

An operator on hypergeometric terms, $A \in \Theta(x_1, \ldots, x_\mu; E_1, \ldots, E_\nu)$ has an obvious interpretation $\phi(A) \in \Theta(x_1, \ldots, x_\mu; E_{\#1}, \ldots, E_{\#\nu})$. Due to the homomorphic properties of the evaluation model on a proper domain, we have

$$\phi(Af) = \phi(A)\phi(f).$$

If you have a p-finite sequence a, then you can express the recurrence by means of an operator, $A \in \Theta(x_1, \ldots, x_\mu; E_{\#1}, \ldots, E_{\#\nu})$ with the simple equation $Aa = 0$.

To find recurrences for sums, we have to analyze the impact of an operator on a definite sum. We have to be careful, as not only the summand is shifted but also the summation bounds. Let $P = (f, b_l, b_u)$, then using the definition of c-evaluation, P defines a function $\phi(P)$ with $\mu - 1$ arguments. Take an arbitrary $A = \sum p_i A_i \in \Theta(x_2, \ldots, x_\mu; E_{\#2}, \ldots E_{\#\nu})$, then we have

$$\begin{aligned} A\phi(P) &= \sum_i p_i A_i \sum_{k=b_l}^{b_u} \phi(f) = \sum_i \sum_{k=A_i b_l}^{A_i b_u} p_i A_i \phi(f) = \\ &= \sum_{k=m_l}^{m_u} A\phi(f) + \sum_i \sum_{k=A_i b_l}^{m_l - 1} p_i A_i \phi(f) + \sum_i \sum_{k=m_u+1}^{A_i b_u} p_i A_i \phi(f). \end{aligned} \tag{6.1}$$

Above we use the functions $m_l(\vec{w}) = \max_i((A_i b_l)[\vec{w}])$ and $m_u(\vec{w}) = \min_i((A_i b_u)[\vec{w}])$. Since b_l and b_u are integer linear in $x_2, \ldots, x_m$, we can determine closed forms $m_l, m_u \in \mathrm{IL}$ for them.

Note that the two double sums, correcting the shift in the bound, are finite. We call them correcting terms. Each of the two correcting terms can be expressed by one hypergeometric term, which can be simplified by means of Ψ to a c-evaluable term (on the intersection of the c- and the m-domain). Therefore, using Equation (6.1) backwards, it is enough to find a sum quantifier free representation for $\sum_{k=m_l}^{m_u} A\phi(f)$ to come up with a sum quantifier free representation of $A\phi(P)$. In other words, to find a recurrence relation for the definite sum it suffices to find an operator $A \in \Theta(x_2, \ldots, x_\mu; E_{\#2}, \ldots E_{\#\nu})$, such that we can find a sum quantifier free representation of $\sum_{k=m_l}^{m_u} A\phi(f)$.

It also suffices to find an $A' \in \Theta(x_2, \ldots, x_\mu; E_2, \ldots E_\nu)$ such that we can find a closed form for the indefinite sum $\sum_{x_1} A' f$. Of course we have to take care of the domain, on which then the interpretation is defined. Note, that it is possible that for a $\vec{w} \in D_M(P)$ we have $m_l[\vec{w}] > m_u[\vec{w}]$ and thus $\vec{w} \notin D_M(\{A'f, m_l, m_u\})$.

6.3.2 Zeilberger's method of creative telescoping

Given f, then the theory of holonomic functions makes it possible – via the k-free recurrence – to find an operator A and a closed form g such that $\sum Af = g$, see [WZ92]. In practice, you can only compute very few examples with this method, since you have to solve huge linear systems, which usually cause your computer to run out of memory.

Zeilberger's method of creative telescoping is a more efficient algorithm, that finds smaller recurrences than the general theory. The algorithm is described for instance in [Zei91]. It is based on Gosper's algorithm and finds the operator A and the closed form for $\sum Af$ at the same time. You can then use Equation (6.1) to find the inhomogeneous part of $A \sum f$.

Given f, then we are looking for $C \in \Theta(x_2, \ldots, x_\mu; E_2)$ and $g \in \text{HT}$, such that $\Delta g = Cf$. Assume that C is given by $\sum_{i=0}^{n} c_i E_2^i$ then

$$Cf = \Big(\sum_{i=0}^{n} c_i \prod_{j=1}^{i} E_2^{j-1} Q_2 f\Big) \cdot f.$$

This means that we can compute two polynomials b_0 and b_1 such that

$$Cf = \frac{b_0(c_0, \ldots, c_n)}{b_1} \cdot f.$$

The polynomial b_0 depends on the coefficients of C, in a linear manner. Note, that Cf is a rational multiple of $1/b_1 \cdot f$ and therefore with the help of proposition 11 also g. Let's say $g = \frac{\varsigma}{b_1} \cdot f$. Expressing everything as multiple of f the requirement $\Delta g = Cf$ reads as

$$Q_1\left(\frac{1}{b_1} \cdot f\right) E_1 \varsigma - \varsigma = b_0(c_0, \ldots, c_n).$$

This is a rational difference equation. With the representation $a_0/a_1 := Q_1(\frac{1}{b_1} \cdot f)$, where a_0 and a_1 are polynomials, we make the coefficients of the difference equation polynomial:

$$a_0\, E_1 \varsigma - a_1 \varsigma = a_1 b_0(c_0, \ldots, c_n).$$

In [Abr89] it is discussed that the denominator of ς has to be bordered by $E_1^{-1} a_0$ and a_1 and it is my claim that it is enough to take p as the largest possible denominator if (p, q, r) is the GP-representation of $\frac{a_0}{a_1}$. Peter Paule established a proof for that, see [Pau95]. Therefore we can transform the equation to a polynomial difference equation ($\varsigma' = p\varsigma$):

$$q\, E_1 \varsigma' - E_1 r \varsigma' = E_1 r\, b_0(c_0, \ldots, c_n).$$

Since $E_1 r$ does not divide q, we read off, that it divides $E_1 \varsigma'$ and with $\varsigma'' r = \varsigma'$ we finally have

$$q\, E_1 \varsigma'' - r\, \varsigma'' = b_0(c_0, \ldots, c_n).$$

Now we have various methods to solve this polynomial difference equation with the extra parameters $c_0, \ldots, c_n$. The easiest one is by computing a degree-bound and comparing coefficients, [Gos78, Zei91], another possibility is [ABP95]. If y solves the equation above, then a representation for the sum is given by

$$g = \frac{r\, y}{p\, b_1} \cdot f.$$

6.3.3 Validity of recurrences found by Zeilberger's algorithm

If Zeilberger's algorithm has found C and g such that $\Delta g = Cf$, then we are close to applying Equation (6.1). What we need is, that $\sum_{k=m_l}^{m_u} \phi(C)\phi(f)$ can be expressed sum quantifier free. In the following we investigate on which domain this interpretation works.

First, notice that for any operator $A \in \Theta(x_2, \dots, x_\mu; E_2)$ the interpretation $\phi(A)$ exists. We have

$$D_m(Af) \subseteq \bigcap_{i=0}^{\mathrm{ord}(A)} D_m(E_2^i f).$$

Proceeding as suggested by proposition 16 we find an analog:

Proposition 17 *(Validity of sum recurrences)*

Given the definite sum $P = (f, b_l, b_u)$, where $g = \frac{r\,y}{p\,b_1} \cdot f$ as above, and C is the output of Zeilberger's algorithm, with input f and order n, then with

$$C' := \mathrm{den}(y) \cdot C, \; g' := \mathrm{den}(y) \cdot g,$$

we have:

$$\forall \vec{w} \in \left(\bigcap_{i=0}^{n} D_{C,M}(E_2^i P) \cap \{\vec{w} \mid \deg_{x_1}(p) < |\text{max-range}(E_2^i P, \vec{w})|\} \right) :$$
$$(\phi(C')\phi(P))(\vec{w}) = \big(\Psi(E_1 g') \circ m_u - \Psi(g') \circ m_l - \text{ correcting Terms}\big)\,[\vec{w}].$$

6.3.4 Bounds for the order of recurrence relations of sums over hypergeometric terms

By the argument above we have a sound and correct method to find recurrences for definite sums. A weak point is, that we had to fix the order of the recurrence before we constructed it. Therefore, looking for a recurrence we have to try Zeilberger's algorithm for recurrences of order 1 then 2, and so on.

For the large class of *proper hypergeometric* terms – these are hypergeometric terms, where the denominator of the rational part can be split into linear factors – it is proven that there exists such a recurrence and furthermore a bound for the order is given, see [WZ92]. The claim is, that the order of the minimal recurrence of

$$\sum_{x_1} \varrho \frac{\prod_{i=1}^{n}(a_i x_2 + b_i x_1 + c_i)!}{\prod_{i=1}^{m}(u_i x_2 + v_i x_1 + w_i)!},$$

found by Zeilberger's algorithm, is bound by

$$\sum_i |b_i| + \sum_i |v_i|.$$

This is not the best bound that can be given. In [Wil91] you find the bound

$$\sum_i \max(b_i, 0) + \sum_i \max(-v_i, 0) + \max\Big(0,\ \sum_i v_i - \sum_i b_i\Big).$$

I suggest to rewrite this bound in a representation that displays the symmetry:

$$\max\Big(\sum_i \max(b_i, 0) + \sum_i \max(-v_i, 0),\ \sum_i \max(-b_i, 0) + \sum_i \max(v_i, 0)\Big).$$

In the latter form you can also see the connection to the GP-representation (p, q, r) of the summand. The bound can be written as $\max(\deg q, \deg r)$.

6.3.5 Examples

SIAM-Problem 95-1

The following example was given as a problem in [XRTM95]:

$$T(m, n) := \sum_{k=0}^{m} \underbrace{\binom{2n - m - k - 1}{n - k}\binom{m + k}{k}}_{t(k,m,n)}.$$

For $m, n \in \mathbb{N}$ we want to find a closed form for T.

Using the Paule/Schorn implementation We try to come up with the recurrence for the sum by simply applying Zeilberger's algorithm:

```
In[1]:=
<< fastZeil.m
Fast Zeilberger by Peter Paule and Markus Schorn. (V 2.0)
Systembreaker = ENullspace

In[2]:=
t[k_,m_,n_] := Binomial[2n-m-k-1, n-k] Binomial[m+k,k];
```

```
In[3]:=
Zb[ t[k,m,n], {k,0,m}, m, 1]
If ‘m’ is a natural number
and ‘-3 - 2 m + 2 n’ is no negative integer, then:
Out[3]=
{SUM[m] - SUM[1 + m] == 0}
```

The second restriction out-putted is due to the fact that otherwise the upper parameter of the binomial changes its sign, which as we know from the evaluation model causes problems. So for $0 \leq m \leq n-1$ we have $T(m,n) = T(0,n) = \binom{2n-1}{n}$.

To examine the case $m \geq n$, I split $T(m,n)$ into two sums, such that in both sums the sign of the upper parameter of the first binomial does not change.

$$\begin{aligned} T_1(m,n) &:= \sum_{k=0}^{2n-m-2} \binom{2n-m-k-1}{n-k}\binom{m+k}{k}, \\ T_2(m,n) &:= \sum_{k=2n-m-1}^{m} (-1)^{n-k}\binom{-n+m}{n-k}\binom{m+k}{k}. \end{aligned}$$

Obviously we have, $T(m,n) = T_1(m,n) + T_2(m,n)$. First we analyze T_1 by applying Zeilberger's algorithm, and we find:

```
In[4]:=
Zb[ t[k,m,n], {k, 0, 2n-m-2}, m, 1]
If ‘-3 - m + 2 n’ is a natural number
and ‘2 (-1 + n)’ is no negative integer, then:
Out[4]=
{SUM[m] - SUM[1 + m] ==

   ((-1 + 2 n) Binomial[1, 2 + m - n]

      Binomial[-2 + 2 n, -2 - m + 2 n]) / (1 + m)}
```

We found an inhomogeneous recurrence. But note, if $m \geq n$ then the binomial $\binom{1}{2+m-n}$ is zero and the recurrence relation turns into a homogeneous one. Therefore for $0 < n \leq m \leq 2n-3$ we know that $T_1(m,n) = T_1(n,n)$. Now $T_1(n,n) = 0$, $T_1(m,0) = 0$, and for $m \leq 2n-3$ even, all summands are zero. If $m \geq n$ we have $T(m,n) = T_2(m,n)$.

We analyze T_2:

```
In[5]:=
u[k_,m_,n_]:= (-1)^(n-k) Binomial[-n+m,n-k] Binomial[m+k,k];
```

```
In[6]:=
Zb[ u[k,m,n],{k, 2n-m-1, m}, m, 1]
If '1 + 2 m - 2 n' is a natural number and none of
{2 m, m - n}
is a negative integer, then:
Out[6]=
{SUM[m] - SUM[1 + m] == 0}
```

For $m \geq n$ we have that $T_2(m,n) = T_2(n,n)$ and with the previous result also $T(m,n) = T_2(n,n)$, for $m \geq n$.

We complete the answer by computing $T_2(n,n)$:

$$T_2(n,n) = \sum_{k=n-1}^{n} (-1)^{n-k} \binom{0}{n-k} \binom{n+k}{k} = \binom{2n}{n}.$$

Summarizing both cases we have

$$T(m,n) = \begin{cases} \binom{2n-1}{n} & \text{, if } m \leq n-1 \\ \binom{2n}{n} & \text{, if } m \geq n \end{cases} .$$

We check the result with some values:

```
In[7]:=
Table[ Sum[ t[k,m,n], {k,0,m}], {n,0,5}, {m,0,6}]˘
Out[7]=
{{1, 1, 1, 1, 1, 1, 1}, {1, 2, 2, 2, 2, 2, 2},
  {3, 3, 6, 6, 6, 6, 6}, {10, 10, 10, 20, 20, 20, 20},
  {35, 35, 35, 35, 70, 70, 70},
  {126, 126, 126, 126, 126, 252, 252}}
```

```
In[8]:=
Table[ If[ m <= n-1, Binomial[2n-1,n], Binomial[2n, n]],
         {n,0,5}, {m,0,6}]
Out[8]=
{{1, 1, 1, 1, 1, 1, 1}, {1, 2, 2, 2, 2, 2, 2},
  {3, 3, 6, 6, 6, 6, 6}, {10, 10, 10, 20, 20, 20, 20},
  {35, 35, 35, 35, 70, 70, 70},
  {126, 126, 126, 126, 126, 252, 252}}
```

A shorter variant of the former proof To prove the result of above for the case $m \geq n$ we use the fact that then $T(m,n) = \sum_{k=0}^{n} t(k,m,n)$. By replacing the upper bound we found a representation of $T(m,n)$ which immediately tells us that it is continuous in m (also the Paule/Schorn implementation knows that!). Therefore the values of $T(m,n)$ can be approached by a limit.

```
In[9]:=
Zb[ t[k,m,n], {k,0,n}, m, 1]
If ‘n’ is a natural number, then:
Out[9]=
{SUM[m] - SUM[1 + m] == 0}
```

Again, the example is completed by computing $T(n,n)$.

Chapter 7

A new method attacks double sums

In the previous chapters we guaranteed the correctness of algorithmically computed recurrences for $\sum_k f$ by the evaluation model. A different approach is naturally given by starting with several small recurrences for the summand, for instance:

$$A(k, n; E_k)f = 0, \; B(k, n; E_n)f = 0.$$

If we compute any *polynomial* linear combination of these two operators, then we again get a correct recurrence relation for f.

$$\Big(C(k, n; E_k, E_n)A + D(k, n; E_k, E_n)B\Big)f = 0. \tag{7.1}$$

Inspired by the inventors of Gosper's and Zeilberger's algorithm I leave the details of when and over what domain such recurrences are valid to the advanced reader. He will observe that for the required correctness proofs, the evaluation model will be helpful, again. Also following famous paragons in the following we assume standard boundary conditions for f, which means that f is non-zero over a finite range for the summation variables only, and f is well defined just outside this range. Now we can conclude that $F(n; E_n)$ annihilates $\sum_k f$ from the fact that $F + (E_k - 1)R$ annihilates f.

If we cleverly choose C and D in Equation (7.1), we will eventually be able to come up with a recurrence $F(n; E_n)$ for the sum over f. All what we need is an operator that can be represented as $CA + DB$ (then it is annihilating) and as $F + (E_k - 1)R$ (then it is summable) at the same time. So we have to find C, D, F and R, such that

$$C(k, n; E_k, E_n)A + D(k, n; E_k, E_n)B = F(n; E_n) + (E_k - 1)R(k, n; E_k, E_n).$$

In the multi-sum case we start with the annihilating operators $A_1(k_1, \ldots, k_r, n, E_{k_1})$, $\ldots$, $A_r(k_1, \ldots, k_r, n, E_{k_r})$, $B(k_1, \ldots, k_r, n; E_n)$ and look for a left linear combination of these operators representable as

$$F(n; E_n) + (E_{k_1} - 1)R_1 + \cdots + (E_{k_r} - 1)R_r.$$

If we solve the problem for $r = 1$, then we have a solution to the general problem: Starting with the operators $A_1, \ldots, A_r$ we can compute the following operators, that are all left linear combinations of the starting operators, by means of an algorithm for the case $r = 1$:

Figure 7.1: Iterative elimination

Here we present an algorithm that in most cases solves the problem for $r = 1$ but sometimes is not able to produce a solution. Nevertheless, on this algorithm we will base an algorithm, algorithm MS, for the general multi-sum case.

As a first surprise we find out that it is enough to look for operators C and D, free of E_k: Assume that for given $A(k, n; E_k)$ and $B(k, n; E_n)$ we have $C(k, n; E_k, E_n)$, $D(k, n; E_k, E_n)$, $Q(k, n; E_k, E_n)$ and $F(n; E_n)$ such that $CA + DB = F + (E_k - 1)Q$. Then we can write $C = C_R(k, n; E_n) + (E_k - 1)C_Q(k, n; E_k, E_n)$ and similar $D = D_R(k, n; E_n) + (E_k - 1)D_Q(k, n; E_k, E_n)$. Now we have

$$C_R A + D_R B = F + (E_k - 1)(Q + C_Q A + D_Q B).$$

So there exist also C_R and D_R, free of E_k solving the problem.

7.1 The algorithm MS

1. As *input* to MS we provide two operators, $A(k, n; E_k)$ and $B(k, n; E_n)$, none of which is free of k.

2. Let $\beta := \deg_k B$, $\alpha := \deg_k A$ and $A = a_0 + E_k a_1 + \cdots + E_k^m a_m$. In the following we use the notation $C(k+j) := (E_k)^j @ C$ – which is different to $(E_k)^j \cdot C$ – and use the fact that $E_k^j C(k-j) = C(k) E_k^j$.

3. We analyze the impact of an arbitrary C on A:

$$\begin{aligned} CA &= C(k)a_0 + E_k C(k-1)a_1 + \cdots + E_k^\alpha C(k-\alpha)a_\alpha = \\ &= C(k)a_0 + \cdots + C(k-\alpha)a_\alpha + \\ &\quad + \underbrace{(E_k-1)C(k-1)a_1 + \cdots + (E_k^\alpha - 1)C(k-\alpha)a_\alpha}_{(E_k-1)Q}. \end{aligned}$$

   So for any C, there exists a Q, such that

$$CA + DB = \underbrace{C(k)a_0 + \cdots + C(k-\alpha)a_\alpha + DB}_{\text{free of } E_k} + (E_k - 1)Q.$$

4. The operator $C(k)a_0 + \cdots + C(k-\alpha)a_\alpha + DB$ is the one we have to make free of k. Remember that C and D are still arbitrary and with the additional unknown F we have to solve the operator equation

$$C(k)a_0 + \cdots + C(k-\alpha)a_\alpha + DB - F = 0. \tag{7.2}$$

5. We make an Ansatz for $C(k,n;E_n)$ and $D(k,n;E_n)$ by fixing their degree in k:

$$\begin{aligned} C &:= \sum_{i=0}^{\beta-1} C_i(n;E_n)k^i, \\ D &:= \sum_{i=0}^{\alpha-1} D_i(n;E_n)k^i. \end{aligned}$$

   Now we can compare the coefficients of k in Equation (7.2), leading us to a system of linear and homogeneous equations in $C_0, \ldots, C_{\beta-1}$, $D_0, \ldots, D_{\alpha-1}$ and F. All these unknowns are operators in n and E_n. Therefore, this is a linear system over an Ore-algebra, see [BP93]. All the unknowns in the set of equations appear with right coefficients. We can compute its null-space by the Gaussian algorithm using right multiplications on the equations.

   Since $\deg_k(CA), \deg_k(DB) \leq \alpha+\beta-1$, we have at most $\alpha+\beta$ linear equations. The number of unknowns is given by $\alpha+\beta+1$ and exceeds the number of equations by one, so for sure, we find a non-trivial null-space vector.

6. It is possible that $F = 0$ in which case we did not solve the problem.

7.2 MS-proofs for double-sum identities

We have a Mathematica implementation of MS looking for recurrence relations of multi-sums over hypergeometric terms with standard boundary conditions and use it to prove two binomial identities of V. Strehl (see [Str93]). The three sums in question have standard boundary conditions.

We want to prove

$$\sum_{k=0}^{n}\sum_{j=0}^{k}\binom{n}{k}\binom{n+k}{k}\binom{k}{j}^3 =$$
$$=\sum_{k=0}^{n}\sum_{j=0}^{k}\binom{n}{k}\binom{n+k}{k}\binom{k}{j}^2\binom{2j}{k}=\sum_{k=0}^{n}\binom{n}{k}^2\binom{n+k}{k}^2.$$

We simply apply MS:

```
In[2]:=
MS[ Binomial[n,k] Binomial[ n+k, k] Binomial[k,j]^3, {j,k},n]
If none of {1 + n, 2 + n} is zero, then
Out[2]=
       3                                 2                3  2
-(1 + n)   +  (3 + 2 n) (39 + 51 n + 17 n ) N  +  -(2 + n)  N

In[3]:=
MS[ Binomial[n,k] Binomial[ n+k, k] Binomial[k,j]^2 *
    Binomial[2j, k], {j,k},n]
If none of {1 + n, 2 + n} is zero, then
Out[3]=
       3                                 2                3  2
-(1 + n)   +  (3 + 2 n) (39 + 51 n + 17 n ) N  +  -(2 + n)  N

In[4]:=
MS[ Binomial[n,k]^2 Binomial[n+k, k]^2, {k}, n]
Out[4]=
      3                                   2                3  2
(1 + n)   +  -((3 + 2 n) (39 + 51 n + 17 n )) N  +  (2 + n)  N
```

Since all of the three sums satisfy the same recurrence relation, we only have to check two initial values (the leading coefficient of the recurrence does not have integer roots in n). So we do that:

```
In[5]:=
Table[
   {Sum[ Binomial[n,k] Binomial[ n+k, k] Binomial[k,j]^3,
         {k,0,n},{j,0,k}],
    Sum[ Binomial[n,k] Binomial[ n+k, k] Binomial[k,j]^2 *
         Binomial[2j, k], {k,0,n}, {j,0,k}],
    Sum[ Binomial[n,k]^2 Binomial[k+n,k]^2, {k,0,n}]}, {n,0,1}]
Out[5]=
{{1, 1, 1}, {5, 5, 5}}
```

The amazing thing is, that we are not only able to compute recurrences for double sums, we can do it very fast! The recurrences for the double-sums above have been computed in less than 20 seconds. We list further double-sums that have been proven with MS and give a list of timings afterwards.

- Trinomial Identity (see [GKP88])

$$\sum_{k=0}^{n}\sum_{i=0}^{k}\binom{n}{k}\binom{k}{i}x^{n-k}y^{i}z^{k-i}=(x+y+z)^{n}. \tag{7.3}$$

- Vandermonde for 2 variables (see [GKP88]

$$\sum_{i=0}^{r}\sum_{j=0}^{s}\binom{r}{i}\binom{s}{j}\binom{t}{n-i-j}=\binom{r+s+t}{n}. \tag{7.4}$$

- Strehl: Binomial Identities (see [Str93])

$$\sum_{k=0}^{n}\sum_{j=0}^{k}\binom{n}{k}\binom{k}{j}^{2}\binom{k}{j+a}=\sum_{k=0}^{n}\binom{n}{k}^{2}\binom{2k}{k+a}. \tag{7.5}$$

$$\sum_{k=0}^{n}\sum_{j=0}^{k}\binom{n}{k}\binom{n+k}{k}^{2}\binom{k}{j}^{3}=\sum_{k=0}^{n}\binom{n}{k}^{2}\binom{n+k}{k}^{2}. \tag{7.6}$$

$$\sum_{k=0}^{n}\sum_{j=0}^{k}\binom{n}{k}\binom{n+k}{k}\binom{k}{j}^{2}\binom{2j}{k}=\sum_{k=0}^{n}\binom{n}{k}^{2}\binom{n+k}{k}^{2}. \tag{7.7}$$

- Stechkin Identity in simple form (see [Ego84]

$$\sum_{i=0}^{n}\sum_{j=0}^{r}\binom{n}{i}\binom{r-n}{j-i}\binom{m}{q-i}\binom{s-m}{q-i+j-l}=$$

$$=\binom{n+m}{q}\binom{r+s-n-m}{r+q-n-l}. \tag{7.8}$$

- Carlitz Identities (see [Car68], [Car64])

$$\sum_{i=0}^{n}\sum_{j=0}^{n}\binom{i+j}{i}\binom{n-i}{j}\binom{n-j}{i}=\sum_{k=0}^{n}\binom{2k}{k} \tag{7.9}$$

$$\sum_{i=0}^{m}\sum_{j=0}^{n}\binom{i+j}{i}\binom{j+m-i}{j}\binom{m+n-i-j}{m-i}\binom{n-j+i}{i}=$$
$$=\frac{(m+n+1)!}{m!n!}\sum_{k=0}^{n}\frac{1}{2k+1}\binom{m}{k}\binom{n}{k}. \tag{7.10}$$

For each identity above you can compute recurrence relations in n for the left and right-hand side. We give a table that lists the order of the recurrence relation found for the left-hand side and the computation time needed using Mathematica 2.2 on an SGI-workstation with the following hardware configuration:

1 100 MHZ IP20 Processor, FPU: MIPS R4010 Floating Point
CPU: MIPS R4000 Processor, Main memory size: 96 Mbytes

Identity	Computation time	Order of computed recurrence
7.3	1 sec.	1
7.4	2 sec.	1
7.5	16 sec.	3
7.6	6 sec.	2
7.7	7 sec.	2
7.8	13 sec.	1
7.9	76 sec.	6
7.10	52 sec.	4

7.3 MS as an alternative to Zeilberger's algorithm of creative telescoping

Each example, where you can find a recurrence relation for a sum by means of Zeilberger's algorithm, may be inputted to MS. It turns out that MS is a little slower than his companion, but note you do not have to supply the order of the recurrence it looks for! Of course, we cannot neglect the unfortunate property that for some examples MS does not find recurrences.

7.3.1 Three easy examples and a conjecture

1. An easy example, where MS delivers just the same recurrence as Zeilberger's algorithm:

```
In[2]:=
Zb[(-1)^k Binomial[n,k] Binomial[3k,n],k,n,1]
Try higher order!
Out[2]=
{}

In[3]:=
Zb[(-1)^k Binomial[n,k] Binomial[3k,n],k,n,2]
Out[3]=
{9 (1 + n) (2 + n) SUM[n] + 3 (2 + n) (7 + 5 n) SUM[1 + n] +
 2 (2 + n) (3 + 2 n) SUM[2 + n] == 0}

In[4]:=
MS[(-1)^k Binomial[n,k] Binomial[3k,n],{k},n]
If none of {n, 1 + n, 2 + n} is zero, then
Out[4]=
                                                 2
-9 (1 + n)  +  -3 (7 + 5 n) N  +  -2 (3 + 2 n) N
```

2. An example where MS finds an operator of higher order:

```
In[5]:=
Zb[(n+k) Binomial[n,k],k,n,1]
Out[5]=
{-2 (1 + n) SUM[n] + n SUM[1 + n] == 0}

In[6]:=
MS[ (n+k) Binomial[n,k], {k}, n]
If none of {1 + n, 2 + n} is zero, then
Out[6]=
2 (1 + n) (2 + n)  +  (2 + n) (12 + 7 n) N  +
                         2
  -2 (1 + n) (3 + 2 n) N
```

3. An example where MS fails:

```
In[7]:=
Zb[1/(n+k) Binomial[n,k], k, n, 2]

Out[7]=
{-2 n (1 + n) (5 + 3 n) SUM[n] -
                               2
      (1 + n) (16 + 44 n + 21 n ) SUM[1 + n] +
      2 (2 + n) (3 + 2 n) (2 + 3 n) SUM[2 + n] == 0}
```

```
In[8]:=
MS[ 1/(n+k) Binomial[n,k], {k}, n]
Method fails at level 1:1
Out[8]=
$Aborted
```

Examining more examples we find a pattern and a conjecture: If we have a polynomial factor in the summand, then the order of the recurrence found by MS is higher than the one found by Zeilberger's algorithm. If we have a polynomial divisor in the summand, then MS fails, else both algorithms deliver the same recurrence.

Actually it is only easy to prove the negative result, namely that MS does not find recurrences, if the summand contains a polynomial divisor p: Then p is a polynomial right-hand factor of A and B and by the operator Equation (7.2) we find that p right-divides F. Since F is constant, it immediately follows that F is zero.

Whether the recurrences found by MS for sums over pure hypergeometric terms are the same as those found by Zeilberger's algorithm is just a conjecture.

7.3.2 A slight improvement and an open problem

If f is a hypergeometric summand containing a polynomial factor p, then we can write $f = p\,f'$. Now f' is a hypergeometric term, too, and with $Af' = 0$, $Bf' = 0$ we have $A\frac{1}{p}f = 0$ and $B\frac{1}{p}f = 0$. Therefore we can look for C, D, F and Q such that

$$CA\frac{1}{p} + DB\frac{1}{p} = F + (K-1)Q\frac{1}{p}.$$

Solving this modified operator equation for the second example above, we find the operator of order 1.

```
In[9]:=
MSpoly[ (n+k) Binomial[n,k], {k}, n]
Out[9]=
{{-2 (1 + n)  +  n N, True}}
```

Since the algorithm was formulated using only operators, we have to ask, how can the algorithm detect polynomial factors in the summand by just looking at the operators. For operators of order 1, you can look at the hypergeometric terms defined by these operators, the generalization to operators of higher order is an open problem to me.

Appendix A

Acknowledgements

To write this thesis I had some help and I thank you →

- Bruno Buchberger, for making RISC possible.
- Peter Paule, the center of energy.
- Istvan Nemes, for all the discussions and all the right papers you handed to me.
- Kurt Wegschaider, for collecting the double-sum identities.

Bibliography

[ABP95] Sergei A. Abramov, Manuel Bronstein, and Marko Petkovšek. On polynomial solutions of linear operator equations. *Journal of Symbolic Computation*, 1995. to appear.

[Abr89] Sergei A. Abramov. Rational solutions of linear differential and difference equations with polynomial coefficients. *USSR Computational Mathematics and Mathematical Physics*, 29(11):1611–1620, 1989. Translation of the Zhurnal Vychislitel'noi matematiki i matematichesckoi fiziki.

[BP93] Michael Bronstein and Marko Petkovšek. On ore rings, linear operators and factorisation. Technical Report 200, Department Informatik, ETH, 1993.

[Car64] Leonard Carlitz. A binomial identity arising from a sorting problem. *SIAM Review*, 6:20–30, 1964.

[Car68] Leonard Carlitz. Summations of products of binomial coefficients. *The American Mathematical Monthly*, 75:906–908, 1968.

[EFT84] Heinz-Dieter Ebbinghaus, Jörg Flum, and Wolfgang Thomas. *Mathematical Logics*, chapter 3, page 52. Springer Verlag, New York, 1984.

[Ego84] Georgy P. Egorychev. *Integral Representation and the Computation of Combinatorial Sums*, volume 59, chapter 2, page 54. American Mathematical Society, 1984.

[GKP88] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Math.* Addison-Wesley Publishing Company, 1988.

[Gos78] R. William Gosper. Decision procedure for indefinite hypergeometric summation. *Proc. Natl. Acad. Sci. USA*, 75:40 –42, 1978.

[Koo92] Tom H. Koornwinder. On zeilberger's algorithm and its q-analogue: a rigorous description. *Journal of Computational and Applied Math.*, 1992.

[LPS93] Petr Lisonek, Peter Paule, and Volker Strehl. Improvement of the Degree Setting in Gosper's Algorithm. *Journal of Symbolic Computation*, 16(3):243–258, September 1993.

[Pau93] Peter Paule. Greatest factorial factorization and symbolic summation I. RISC-Linz Report Series 93-02, J. Kepler University, Linz, 1993.

[Pau95] Peter Paule. Greatest factorial factorization and symbolic summation. *Journal of Symbolic Computation*, 1995. to appear.

[Pet92] Marko Petkovšek. Hypergeometric solutions of linear recurrences with polynomial coefficients. *Journal of Symbolic Computation*, 14:243–264, 1992.

[PN95] Peter Paule and Istvan Nemes. A canonical form guide to symbolic summation. to appear in tasso monography, 1995.

[PS95] Peter Paule and Markus Schorn. A Mathematica version of Zeilberger's algorithm for proving binomial coefficient identities. *Journal of Symbolic Computation*, 1995. to appear.

[SLL89] Dominic Y. Savio, Edmund A. Lamagna, and Shing-Min Liu. Summation of harmonic numbers. In Stephen M. Watt Erich Kaltofen, editor, *Computer and Mathematics*, page 18. Springer-Verlag New York, 1989.

[Str93] Volker Strehl. Binomial identities - combinatorial and algorithmic aspects. Technical Report 93-1, Institut für Mathematische Maschinen und Datenverarbeitung der Universität Erlangen, January 1993.

[Wil91] Herbert S. Wilf. Sums of closed form functions satisfy recurrence relations. University of Pennsylvania, Philadelphia, 1991.

[WZ92] Herbert S. Wilf and Doron Zeilberger. An algorithmic proof theory for hypergeometric (ordinary and "q") multisum / integral identities. *Inventiones Mathematicæ*, 108:575–633, 1992.

[XRTM95] Ma Xin-Rong and Wang Tian-Ming. Problem 95-1. *SIAM Review*, 37(1):98, March 1995.

[Zei90] Doron Zeilberger. A fast algorithm for proving terminating hypergeometric identities. *Discrete Math.*, 80:207–211, 1990.

[Zei91] Doron Zeilberger. The method of creative telescoping. *Journal of Symbolic Computation*, 11:195–204, 1991.