ALGORITHMIC IDENTITY PROVING AND INVERSE PROBLEMS

by
YALCIN CAN KILIC

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfilment of
the requirements for the degree of Master of Science

Sabancı University
June 2020

Algorithmic Indentity Proving and Inverse Problems

APPROVED BY:

Assoc. Prof. Dr. Kağan Kurşungöz
(Thesis Supervisor)

Assoc. Prof. Dr. Mohammad Sadek

Asst. Prof. Dr. Özer Öztürk

DATE OF APPROVAL: 17/07/2020

ABSTRACT

ALGORITHMIC IDENTITY PROVING AND INVERSE PROBLEMS

YALCIN CAN KILIC

Mathematics M.A. THESIS, June 2020

Thesis Supervisor: Assoc.Prof. Kagan Kursungoz

Keywords: Hypergeometric Summation, Sister Celine Algorithm, Gosper's Algorithm, Creative Telescoping Algorithm, Algorithm Hyper, Inverse Zeilberger Problem

In their book 'A=B' Marko Petkovsek, Herbert Wilf and Doron Zeilberger talked about computer generated proofs of identities which contains hypergeometric functions and four fundamental algorithms about them: Sister Celine's Algorithm, Gosper's Algorithm, Zeilberger's Algorithm and Algorithm Hyper. Sister Celine's algorithm, given a definite sum with proper hypergeometric summand finds a linear recurrence operator with polynomial coefficients which annihilates the given sum. Gosper's algorithm, given an indefinite sum with hypergeometric summand decides whether this sum can be written as a sum of hypergeometric function and a constant. Zeilberger's algorithm does the exact same job as Sister Celine's algorithm. However, it is much faster. Algorithm Hyper, given a linear recurrence equation with polynomial coefficients checks whether this recurrence has hypergeometric solutions or not. In addition, Petkovsek wrote an article 'Definite Sums as Solutions of Linear Recurrences With Polynomial Coefficients' which tries to solve, so called Inverse Zeilberger Problem: Given a linear recurrence operator with polynomial coefficients, find a sum which is annihilated by the given linear recurrence operator.

In this thesis, these four algorithms are examined in detail, and numerous examples are given. Then, an inverse problem is described, and Petkovsek's recent paper on an instance of this particular problem is explicated. Finally, the algorithms are briefly analyzed in many aspects such as generality, time and space complexity, etc.

ÖZET

ALGORITHMIC IDENTITY PROVING AND INVERSE PROBLEMS

YALÇIN CAN KILIÇ

MATEMATİK YÜKSEK LİSANS TEZİ, MAYIS 2020

Tez Danışmanı: Doç. Dr. Kağan Kurşungöz

Anahtar Kelimeler: Hipergeometrik Toplama, Rahibe Celine Algoritması, Gosper Algoritması, Yaratici Sadeleştirme Algorithması, Algorithm Hiper, Ters Zeilberger Problemi

Marko Petkovsek, Herbert Wilf ve Doron Zeilberger, " A = B " adlı kitaplarında hipergeometrik fonksiyonlar ve bunlar hakkında dört temel algoritma içeren bilgisayar tarafından oluşturulan özdeşlik kanıtlarını anlatıyorlar: Rahibe Celine Algoritması, Gosper Algoritması, Zeilberger Algoritması ve Algoritma Hiper. Rahibe Celine'in algoritması, sonsuz bir hipergeometrik toplam verildiğinde, verilen toplamı sıfırlayan polinom katsayıları olan doğrusal bir yineleme operatörü bulur. Gosper algoritması, sonlu bir hipergeometrik toplam verildiğinde, bu toplamın bir hipergeometrik fonksiyon ve bir sabitin toplamı olarak yazılıp yazılamayacağına karar verir. Zeilberger'in algoritması Rahiber Celine'in algoritmasıyla aynı işi yapıyor. Ancak, çok daha hızlı. Algoritma Hiper ise polinom katsayıları ile doğrusal bir yineleme denklemi verildiğinde, bu yinelemenin hipergeometrik çözümlere sahip olup olmadığını kontrol eder. Buna ek olarak, Petkovsek " Polinom Katsayılı Doğrusal Yinelemelerin Çözümü Olarak Belirli Toplamlar" adlı makalesinde Ters Zeilberger Problemi olarak adlandırılan " Polinom katsayılı doğrusal bir yineleme operatörü verildiğinde, verilen operatör tarafından sıfırlanan bir toplamı bulma" problemini çözmeye calışır.

Bu tezde, bu dört algoritma ayrıntılı olarak incelenmiş ve çok sayıda örnek verilmiştir. Ardından, ters Zeilberger problemi tanımlanmış ve Petkovsek'in bu probleminin özel bir durumu hakkındaki son makalesi açıklanmıştır. Son olarak, algoritmalar genellik, zaman ve mekan karmaşıklığı gibi birçok açıdan kısaca analiz edilmiştir.

ACKNOWLEDGEMENTS

First of all, I would like to thank my thesis advisor, Kağan Kurşungöz. With his kind personality, he is always there when I need help. Also, he motivated me with his smiling face and patience when my moral was down. Furthermore, I want to thank whole Sabanci University Math Department. I know that I will not be at this position without their help. Moreover, all of my friends are always with me throughout this challenging but exciting journey. They sometimes with me as a friend, sometimes as a teacher or a student , or just a listener. I cannot thank them enough. Finally, I would like to thanks my parents. They always care for me and try to help me in whole possible ways, they can. I hope, in the future I can help my children, like they help me today. Last but not least, I would like to thank TUBITAK for their scholarship.

to my mother, father and brother
anneme, babama ve kardesime

TABLE OF CONTENTS

1. INTRODUCTION

Let's start with a theorem right off:

Theorem 1. $1535215412 \times 36373252131 = 55840777256073042972$

Can we publish an article in which Theorem 1 is the main result of the paper? Of course, it is impossible! But, what is the exact reason? Why cannot we publish it? It is most probably an original result. Thus, it is not about originality of the result. The main problem is that, it is *routine*! Let's define the term *routine* in a proper way:

Definition 1. *If there is a well-defined, step-by-step solution to a problem, then this problem is called* ***routine****. Otherwise, it is called* ***non-routine****.*

There is a problem with the above definition, since a problem can be routine to one person and non-routine to another person. As an example, probably, Theorem 1 is routine to elementary school students but it is non-routine for kindergarten students! This means that, we need a more objective way to assess routineness of a problem. Fortunately, with computers it is easy to make this definition more proper:

Definition 2. *If a problem can be solved by a computer, in a reasonable time(whatever that means!), it is called* ***(universally) routine*** *problem.*

However, one may argue that some computers are (much) faster than the others. This means that in Definition 2 *reasonable time* is not well-defined. We can solve this problem with the abstraction: Rather than talking about a special computer we can talk about an algorithm! Hence, our final definition is the following:

Definition 3. *A problem is called* ***(universally) routine*** *if there is a fast(whatever that means!) algorithm to solve this problem.*

Let's look at some famous examples of routine problems:

Example 1.

1. *Finding the reduced row echelon form of a matrix.*
2. *Checking the irreducibility of a polynomial over* $\mathbb{Q}$*.*

3. *Finding the greatest common divisor of two numbers.*

4. *Finding the Jordan canonical form of a matrix.*

5. *Sorting n numbers in an ascending order.*

Let's look at the other side of the medal:

Example 2. 1. *Yuri Matiyasevish proved that Hilbert's 10th problem is non-routine Matiyasevich (1993).*

2. *Richardson proved that some identities are non-routine Richardson (1966).*

We also have a gray area in the middle of these problems. In other words, we do not know whether a problem is routine or non-routine.

Example 3. 1. *Given a graph checking whether the graph has a Hamiltonian cycle or not.*

2. *Given two graphs checking whether they are isomorphic or not.*

Thus, it is not known whether there is a fast algorithm to solve above two problems.

Definition 4. ***Computer algebra*** *is the part of mathematics which tries to* routinese *problems.* ***Computer algebraist*** *is a mathematician who works on computer algebra.*

We will discuss a subpart of computer algebra called **Hypergeometric summation**. Loosely speaking, we will try to evaluate sums of the form $\sum_{k=m}^{n} F(n,k)$ where m and n and integers such that $m < n$ and $F(n,k)$ is a hypergeometric term with respect to n and k (we will define this term in the next chapter). We will proceed as follows:

1. In the second chapter we will look at some definitions.

2. In the third chapter, we will discuss Sister Celine's algorithm. One can argue that her algorithm is the beginning of the hypergeometric summation algorithms.

3. In the fourth chapter, we will discuss Gosper's algorithm. It is the cornerstone of the creative telescoping algorithm.

4. In the fifth chapter, we will discuss creative telescoping algorithm(Zeilberger's algorithm). This is a (much) faster version of Sister Celine's algorithm.

5. In the sixth chapter, algorithm hyper will be discussed. In this chapter we slightly change our perspective and look at the solution of recurrences.

6. Finally, in the last chapter inverse Zeilberger problem will be discussed. Here, we look at the inverse creative telescoping algorithm.

Most of the material in these chapter are coming from Petkovšek, Wilf & Zeilberger (1996).

2. Definitions

In this section we will give some definitions, fix the notation etc.

Definition 5. *A function $f(n)$ is said to be* ***hypergeometric*** *if the term ratio, i.e $\frac{f(n+1)}{f(n)}$, is a rational function of n. Similarly, we can generalize the notion as $f(n,k)$ is said to be* ***hypergeometric*** *in both arguments if $\frac{f(n+1,k)}{f(n,k)}$ and $\frac{f(n,k+1)}{f(n,k)}$ are both rational functions of n and k.*

Remark 1. *We always consider our variables as discrete objects. In other words, $f(n): \mathbb{N} \to \mathbb{R}$ and $f(n,k): \mathbb{N} \times \mathbb{N} \to \mathbb{R}$. Thus, we can see them as sequences as well. From now on, we use the terms function and sequence interchangeably.*

In Definition 5 we see that we can study functions of 2-variables. This can be generalized to any finite number of variables in a straightforward manner. However, for our purposes one-variable and two-variable cases would be sufficient. Let us look at some examples and non-examples:

Example 4.

1. *$F_1(n) = n^2 + 3$ is hypergeometric, since $\frac{F_1(n+1)}{F_1(n)} = \frac{(n+1)^2+3}{n^2+3}$ is a rational function of n.*
2. *It is easy to generalize the first example:*
   (a) *First, if $F(n)$ is a polynomial in n, then, $F(n)$ is a hypergeometric function.*
   (b) *Second, more generally, if $F(n)$ is a rational function of n, then $F(n)$ is a hypergeometric function.*
3. *$F(n,k) = \binom{n}{k}$ is a hypergeometric function in both arguments.*
4. *More generally, $F(n,k) = \binom{an+b}{ck+d}$ is a hypergeometric function in both argu-*

ments. Since,

$$\frac{F(n+1,k)}{F(n,k)} = \frac{\binom{an+a+b}{ck+d}}{\binom{an+b}{ck+d}} =$$

$$\frac{(an+a+b)(an+a+b-1)...(an+b+1)}{(an+a+b-ck-d)(an+a+b-ck-d-1)...(an+b-ck-d+1)}$$

Thus, $F(n,k)$ *is a hypergeometric function with respect to* n*. Similarly,*

$$\frac{F(n,k+1)}{F(n,k)} = \frac{\binom{an+b}{ck+c+d}}{\binom{an+b}{ck+d}} =$$

$$\frac{(an+b-ck-c-d)(an+b-ck-c-d+1)...(an+b-ck-d)}{(ck+d+1)(ck+d+2)...(ck+c+d)}$$

Hence, $F(n,k)$ *is a hypergeometric function with respect to* k*.*

5. *Consider the series of the form* $\sum_k t_k$ *where* t_k *is a hypergeometric function. Such series are called* ***hypergeometric series****. Some examples of hypergeometric series:*

   (a) e^x *and* $ln(x)$*.*

   (b) *Trigonometric functions like* $sin(x)$, $cos(x)$, $tan(x)$ *etc.*

   (c) *Legendre polynomials.*

   (d) *Bessel polynomials etc.*

   One can see that this functions are indeed hypergeometric series by looking at their Taylor expansions. For example, $e^x = \sum\limits_{k\geq 0} \frac{x^k}{k!}$*. Obviously, the summand is a hypergeometric function in* k*.*

6. *Let's look at some functions which are not hypergeometric:*

   (a) *Trigonometric functions such as* $sin(x)$, $cos(x)$, $tan(x)$ *are not hypergeometric since* $\frac{sin(x+1)}{sin(x)}$, $\frac{cos(x+1)}{cos(x)}$ *and* $\frac{tan(x+1)}{tan(x)}$ *are not rational functions of* x*. One can prove this fact by showing that all of the above fractions have infinitely many roots. If they are rational functions, this cannot be the case.*

   (b) $\sqrt{x}$ *is not hypergeometric.*

   (c) $ln(x)$ *is not hypergeometric.*

Remark 2. 1. *There is an equivalent definition of hypergeometric functions as* *well: A function $f(n)$ is hypergeometric if it satisfies a first-order homogeneous linear recurrence with polynomial coefficient, i.e. there exists polynomials $p(n)$ and $q(n)$ such that*

$$p(n)f(n) - q(n)f(n+1) = 0.$$

2. *We can generalize this as $F(n,k)$ is a hypergeometric function on both arguments if and only if it satisfies two first-order homogeneous linear recurrence with polynomial coefficients, i.e there exists polynomials $p_1(n,k)$, $q_1(n,k)$, $p_2(n,k)$ and $q_2(n,k)$ such that*

$$p_1(n,k)F(n,k) - q_1(n,k)F(n,k+1) = 0,$$
$$p_2(n,k)F(n,k) - q_2(n,k)F(n+1,k) = 0.$$

3. *The good thing about this definition is that, another generalization is also possible: The function $F(n)$ is called **P-recursive** if it satisfies a homogeneous recurrence with polynomial coefficients. In other words, $F(n)$ is P-recursive if there exists polynomials $p_0(n)$, $p_1(n)$,...$p_r(n)$ such that*

$$p_0(n)F(n) + p_1(n)F(n+1) + ..p_r(n)F(n+r) = 0.$$

Remark 3. *Now we have a class of functions called hypergeometric functions. It is good to check whether they are closed under two basic operations: Multiplication and addition. Suppose $F(n)$ and $G(n)$ are hypergeometric functions. It is clear that $F(n)G(n)$ is hypergeometric. However, for $F(n)+G(n)$ the answer is: it depends! In other words,*

1. *If we choose $F(n) = n$ and $G(n) = n^3$, then $F(n)+G(n) = n^3+n$ is indeed a hypergeometric function.*

2. *However, choosing $F(n) = 2^n$ and $G(n) = 1$ shows that $F(n)+G(n) = 2^n+1$ which is not a hypergeometric function!*

Remark 4. *Although we define the hypergeometric functions above, with similar spirit we can define hypergeometric sequence as well. A sequence $\langle a_n \rangle_{n=0}^{\infty}$ is called a **hypergeometric sequence** if there is a rational function $R(n)$ such that $a_{n+1} = R(n)a_n$.*

Some motivation to study hypergeometric functions or sum of hypergeometric functions:

1. Their theory is well-studied. In other words, if you know that the function

under consideration is hypergeometric you can use vast amount of transformations, algorithms etc.

2. We want to study functions which are easy to work with(in other words, *nice*) but general as well. We can study constant functions since they are very easy to work with! However, they are not general at all. In other words, lots of the functions we study are not constant. Thus, hypergeometric functions are a balance of ease and generality.

Remark 5. *There is a difference between the one-variable case and the two-variable case. In one-variable case, for any choice of rational function $r(n)$, we can find a sequence $\{a_n\}_{n=0}^{\infty}$ such that, $r(n) = \frac{a(n+1)}{a(n)}$. In two-variable case, it is not true. More precisely, there exist rational functions $R_1(n,k)$ and $R_2(n,k)$ such that, we cannot find a sequence $\{a_{n,k}\}_{n,k\geq 0}$ where $R_1(n,k) = \frac{a_{n+1,k}}{a_{n,k}}$ and $R_2(n,k) = \frac{a_{n,k+1}}{a_{n,k}}$. To sum up: Not every rational function can be a term ratio in two variable case. The reason is the following: Suppose $F(n,k)$ is a hypergeometric term in both arguments, in other words $R_1(n,k) = \frac{F(n+1,k)}{F(n,k)}$ and $R_2(n,k) = \frac{F(n,k+1)}{F(n,k)}$ are both rational functions of n and k. Then, the critical point is that, the shift in n followed by shift in k must be same as shift in k followed by shift in n. This means that the following equation must hold:*

$$\frac{R_1(n,k+1)}{R_1(n,k)} = \frac{R_2(n+1,k)}{R_2(n,k)}.$$

Remark 6. *The problem of deciding whether a function is hypergeometric is completely algorithmic. The following Maple code gives us whether a particular function is hypergeometric or not: IsHypergeometricTerm(F,n). More precisely, this means that, we have an efficient algorithm to decide whether a function is a hypergeometric function or not. Thus, for example,*

1. *$IsHypergeometricTerm(sin(x),x)$ returns false. In other words, $sin(x)$ is not a hypergeometric function with respect to x.*

2. *$IsHypergeometricTerm(2^n,n)$ returns true in Maple. This means that 2^n is a hypergeometric function with respect to n.*

Remark 7. *Suppose we have a two-variable hypergeometric sequence, i.e $\{a_{n,k}\}_{n,k=0}^{\infty}$ such that $a_{n,k}$ is hypergeometric in both arguments. Then, it can be the case that our sequence can be expressible using gamma function, or it may not be the case. Thus, we will distinguish the ones which can be expressed using the gamma function. Intuitively, a function can be expressed using the gamma function means that, it can be written as a ratio of factorial product. See below for the definition and concrete examples.*

Definition 6. *A function F(n,k) is called an* ***proper hypergeometric term*** *if we can write* $F(n,k)$ *as*
$F(n,k) = P(n,k)\frac{\prod_{i=1}^{m}(a_i n+b_i k+c_i)!}{\prod_{i=1}^{t}(u_i n+v_i k+w_i)!}$
where
1- P is a polynomial of n and k,
2- a,b,u,v are fixed integers, in other words they do not contain any parameters,
3- Also m and t are specific non-negative integers,

Example 5.

1. *If* $F(n,k)$ *is any polynomial of n and k, then obviously,* $F(n,k)$ *is a proper hypergeometric term. Since, we can choose* $P(n,k) = F(n,k)$ *in the definition and take* $m = 0$ *,* $t = 0$*.*

2. $\frac{1}{n^2+k^2+1}$ *is not a proper hypergeometric term. Note that it is a hypergeometric function.*

3. *All functions of the form* $F(n,k) = \binom{a_1n+b_1}{c_1k+d_1}\binom{a_2n+b_2}{c_2k+d_2}\cdots\binom{a_rn+b_r}{c_rk+d_r}$ *are proper hypergeometric functions where* a_i *,* b_i *,* c_i *and* d_i *are fixed integers.*

Remark 8.

1. *All proper hypergeometric functions are hypergeometric, but the converse is not true as shown in Example 5.*

2. *Checking whether a particular function is proper hypergeometric or not is completely algorithmic, in Maple, the function IsProperHypergeometricTerm(F(n,k),n,k) completely solves the problem. In other words, IsProperHypergeometricTerm(F(n,k),n,k) returns true, if* $F(n,k)$ *is a proper hypergeometric term with respect to* n *and* k*. Otherwise, it returns false. Thus,*

   (a) $IsProperHypergeometricTerm(\frac{1}{n^2+k^2+1}, n, k)$ *returns false, meaning that* $\frac{1}{n^2+k^2+1}$ *is not a proper hypergeometric function.*

   (b) $IsProperHypergeometricTerm(\frac{\binom{3n+4}{2k+3}2^{n+k}k!^3}{k^3+5k^2+3k+87}, n, k)$ *returns true, meaning that* $\frac{\binom{3n+4}{2k+3}2^{n+k}k!^3}{k^3+5k^2+3k+87}$ *is a proper hypergeometric function.*

3. *Also, it is very important to understand whether* $F(n,k)$ *is proper hypergeometric or not. Since, if it is proper hypergeometric, then Zeilberger's algorithm must terminate. For details, see Theorem 2 and Theorem 6 in Chapter 3 and Chapter 5, respectively.*

When we are talking about infinite sums, we should be careful about the convergence issues. One way to solve this issue is that, having a sum $\sum_k F(n,k)$ such that the summand, $F(n,k)$ is zero for all but finitely many k. We will give a special name to such functions:

Definition 7. *A function $F(n,k)$ is said to have a* ***compact support*** *if for all n, $F(n,k)$ is 0 for all values of k but finitely many k. In other words, for each fixed n, $|\{k \in \mathbb{Z} : F(n,k) \neq 0\}|$ is finite.*

Let's look at examples and non-examples:

Example 6.

1. *$F(n,k) = \binom{n}{k}$ has a compact support since $\binom{n}{k} = 0$ for $k > n$ or $k < 0$. To be more precise, $|\{k \in \mathbb{Z} : \binom{n}{k} \neq 0\}| = n+1$ for each n.*
2. *$F(n,k) = n^2 k$ does not have a compact support since, for instance, when $n = 1$ $F(n,k) \neq 0$ for infinitely many k.*
3. *Generalizing the first example : $F(n,k) = \binom{3n+5}{2k-4}$ has a compact support since $\binom{3n+5}{2k-4} = 0$ for $2k-4 > 3n+5$ or $2k-4 < 0$.*
4. *We can generalize the above example further as follows: If*

$$F(n,k) = \binom{an+b}{ck+d}$$

   . Then, $F(n,k)$ has a compact support.
5. *Even more generalization is possible: If $F(n,k) = \binom{an+b}{ck+d} G(n,k)$ where $G(n,k)$ is a hypergeometric function in both arguments. Then, $F(n,k)$ is a hypergeometric function in both arguments and $F(n,k)$ has a compact support.*

Moral of the Story: The binomial coefficients are our main actors to guarantee compact support. From now on, we will (almost) always considers summand with binomial coefficients. So that we do not need to worry about converge at all!

Let's look at more examples in the following table:

More Examples			
	Hypergeometric	Proper Hypergeometric	Compact Support
$\binom{n}{k}$	true	true	true
$(2n+k+3)!$	true	true	false
$\frac{1}{n^2+k^2}\binom{n}{k}$	true	false	true
$\frac{1}{n^2+k^2}$	true	false	false
Impossible	false	true	true
Impossible	false	true	false
$\begin{cases} e^x & \|x\| \leq 1 \\ 0 & Otherwise \end{cases}$	false	false	true
$cot(x)$	false	false	false

From now on, we will usually consider the functions of the form

$$F(n,k) = \binom{an+b}{ck+d} G(n,k)$$

where $G(n,k)$ is hypergeometric in both arguments. The reason is that we want to have a hypergeometric function with compact support. Above we observed that multiplication of two hypergeometric function is again hypergeometric. Also we see that the sum of hypergeometric functions may or may not be hypergeometric:

Example 7. 1. *If $F(n) = 2^n$ and $G(n) = 1$. Then, obviously $F(n)$ and $G(n)$ are hypergeometric functions but $F(n)+G(n) = 2^n+1$ is not a hypergeometric function.*

2. *If $F(n) = n!$ and $G(n) = (n-1)!\cdot(n+5)$, then both $F(n)$ and $G(n)$ are hypergeometric functions. Also, $F(n)+G(n)$ is a hypergeometric function.*

Thus, it makes sense to distinguish the ones whose sum is a hypergeometric function.

Definition 8. *Two hypergeometric functions $F(n)$ and $G(n)$ are said to be **similar** if their sum is also a hypergeometric function.*

Before looking at examples and non-examples of similar hypergeometric functions, we need to answer: "Okay, hypergeometric functions are easy to work with. They are well studied. Why we want to consider their sums, as well?" The answer is obvious once we look at the function in the above example: 2^n+1. Even though it is not hypergeometric, it is a well-behaved and easy-to-work-with function. Also, it is a sum of hypergeometric functions, 2^n and 1, as well.

Moral of the Story: Usually, sum of hypergeometric functions behave nicely as well. Therefore, we want to study sums of hypergeometric functions. For example sums of the form $\sum_k F(n,k)$ where $F(n,k)$ is hypergeometric in both arguments and $F(n,k)$ has a compact support so that we skip convergence issues.

It is a good idea to distinguish between the sums which can be written as sum of hypergeometric functions. To be more precise, given a definite sum $f(n) = \sum_{k=0}^{n} t_k$ where t_k is a hypergeometric term. We want to understand whether $f(n)$ can be written as a sum of hypergeometric functions or not. Actually, it is already the sum of hypergeometric functions! The critical question is that whether we can write $F(n)$ as a sum of *fixed number of* hypergeometric functions(We will make this concept more precise). We start with the easiest case: Can we write $f(n)$ as a sum of a hypergeometric function plus a constant - e.g., $2^n + 1$ - ? Again, both answers to our question is possible. Thus, we will distinguish the sums $f(n) = \sum_{k=0}^{n} t_k$ such that $f(n)$ can be written as a hypergeometric term plus a constant.

Definition 9. *Given a sum $f(n) = \sum_{k=0}^{n} t_k$, the summand t_k is called* ***Gosper summable****, if there exists a hypergeometric function d_n such that*

$$t_n = d_{n+1} - d_n. \tag{2.1}$$

The following remark shows that if a summand t_k is Gosper summable, then we really answer the question of whether $f(n)$ can be written as a sum of a hypergeometric term and a constant or not:

Remark 9. *Suppose t_k is Gosper Summable, then there exists a hypergeometric function d_n such that $d_{n+1} - d_n = t_n$. Thus,*

$$\sum_{k=0}^{n} t_k = \sum_{k=0}^{n} d_{k+1} - d_k = d_{n+1} - d_0. \tag{2.2}$$

Obviously, d_{n+1} is a hypergeometric function and d_0 is a constant. Thus, we write $\sum_{k=0}^{n} t_k$ as a sum of hypergeometric term and a constant. This also means that, once we find d_n we do evaluate the sum as well!

Example 8. *1. $k \cdot k!$ is Gosper summable since $\sum_{k=0}^{n} k \cdot k! = (k+1)! - 1$. Thus, we can choose $d_n = n!$ in the definition.*

2. However, $k!$ is not Gosper summable in other words there does not exist a hypergeometric function $d(n)$ such that $n! = d_{n+1} - d_n$ holds. We will prove *this in Chapter 4.*

3. $\frac{1}{k}$ *is not Gosper summable as well. We will show this in Chapter 4.*

4. *All polynomials are Gosper summable. We will prove this in Chapter 4.*

5. *As shown above, not all rational functions are Gosper summable. Malm and Subramaniam, in Malm & Subramaniam (1995), gives an algorithm to check whether a particular rational function is Gosper summable or not.*

Remark 10. *In Chapter 4, we will discuss an algorithm, called Gosper's Algorithm which completely solves the following problem: Given a definite sum, can we find* $d(n)$ *as in* (2.1)*? In other words, if such a* $d(n)$ *exists Gosper's algorithm finds it. Otherwise, it shows(proves!) that such a* $d(n)$ *does not exist.*

It is reasonable to ask the relationship between hypergeometricity of the function and Gosper summability. In other words,

1. Suppose t_k is Gosper summable. Can we conclude that t_k is a hypergeometric function?

2. Suppose t_k is hypergeometric. Can we conclude that t_k is Gosper summable?

We already answered the second question as negative. For instance, $k!$ is a hypergeometric function but it is not Gosper summable. However, as the next remark shows the answer of the first question is positive.

Remark 11. *Suppose* t_k *is Gosper summable, then we want to show that* t_k *is a hypergeometric function. Since* t_k *is Gosper summable, there exists a hypergeometric function* d_n *such that* $t_n = d_{n+1} - d_n$ *. Then,*

$$\frac{t_{n+1}}{t_n} = \frac{d_{n+2} - d_{n+1}}{d_{n+1} - d_n} = \frac{\frac{d_{n+2}}{d_{n+1}} - 1}{\frac{d_{n+1}}{d_n} - 1}$$

Thus, t_n *is a hypergeometric function.*

Remark 12. *Let's look at some analogous problems from different settings:*

1. *Given a function* $H(x) = \int_a^x f(t)dt$*, can we find a function* $F(x)$ *such that* $H(x) = F(x) - F(a)$*?*

2. *Let* F *be a field. Given a polynomial* $P(x) \in F[x]$*, can we find a polynomial* $T(x) \in F[X]$ *such that* $P(x) = T(x+1) - T(x)$*?*

3. *We can generalize above question to the rational functions: Let* F *be a field. Given a rational function* $R(x) \in F(x)$*, can we find a rational function* $Q(x) \in$

$F(x)$ such that $R(x) = Q(x+1) - Q(x)$?

4. *Thus, we can create instances of the same problem via picking a property P in the following statement: Given a function $F(x)$ with property P can we find function $Q(x)$ with property P such that $F(x) = Q(x+1) - Q(x)$?*

Now, we will introduce the concept of canonical form of rational functions. It is important for Gosper's algorithm and Zeilberger's algorithm which will be described in Chapter 4 and Chapter 5.

Definition 10. *Let $f(n)$ be a rational function. Then, a* ***canonical form*** *of $f(n)$ is*

$$f(n) = \frac{a(n)}{b(n)} \frac{c(n+1)}{c(n)} \tag{2.3}$$

where $a(n)$, $b(n)$ and $c(n)$ are polynomials that satisfies:

1. *$a(n)$ and $c(n)$ are relatively prime*
2. *$b(n)$ and $c(n+1)$ are relatively prime*
3. *$gcd(a(n), b(n+h)) = 1$ for every nonnegative integer h.*

Example 9.

1. *Let $f(n) = \frac{6n+3}{(2n^2+1)(2n^2+4n+3)}$. Then, we can choose $a(n) = \frac{3}{2}(n+\frac{1}{2})$, $b(n) = \frac{(2n^2+1)(2n^2+4n+3)}{4}$ and $c(n) = 1$.*
2. *Let $g(n) = \frac{2(n^3-6n-6)(n^3)}{(n+2)^3(n^3-3n^2-3n-1)}$. Then, $a(n) = 2n^3$, $b(n) = (n+2)^3$ and $c(n) = n^3 - 3n^2 - 3n - 1$ satisfies (2.3).*
3. *Let $h(n) = (n+k)(n+k)!$ where k is a formal parameter. Then, we can choose $a(n) = n+k+1$, $b(n) = 1$ and $c(n) = n+k$.*

Remark 13. *At first glance it not obvious whether the canonical form of a rational function always exists or not. In Chapter 4, we will prove that such a canonical form always exists. Moreover, we will discuss an algorithm to find the canonical form of a rational function.*

Remark 14. *Note that the third condition is a stronger property than being relatively prime. For example, if $a(n) = n+4$ and $b(n) = n+1$. Then, $gcd(a(n), b(n)) = 1$. So they are relatively prime. However, $gcd(a(n), b(n+3)) = n+4$. Thus, our third condition is not satisfied.*

Remark 15. *There are other canonical forms of rational functions exists as well. We will not discuss them. See Abramov, Le & Petkovšek (2003).*

Remark 16. *Given a rational function $f(n)$, we can find its canonical form using Maple as follows: PolynomialNormalForm(f,n).*

1. *Given* $f(n) = \frac{(n+2)(n-3)(n+5)}{(n-5)(n+4)(n+6)(n+3)}$, *PolynomialNormalForm(f,n) gives* 1, $n+2$, $(n+6)(n+3)$, $(n+4)(n-4)(n-5)$ *means that we can choose* $a(n) = (n+2)$, $b(n) = (n+6)(n+3)$ *and* $c(n) = (n+4)(n-4)(n-5)$.

Definition 11. *A function N is called the **forward shift operator** in n. Similarly, K is called the **forward shift operator** in k. In other words, $Nf(n,k) = f(n+1,k)$ and $Kg(n,k) = g(n,k+1)$.*

Example 10.

1. *Let* $f(n,k) = \binom{n}{k}$. *Then,* $Nf(n,k) = f(n+1,k) = \binom{n+1}{k}$. *Similarly,* $Kf(n,k) = f(n,k+1) = \binom{n}{k+1}$.
2. *Let* $g(n,k) = (n+1)^2 + nk + k^3$. *Then,* $Ng(n,k) = g(n+1,k) = (n+2)^2 + (n+1)k + k^3$. *Similarly,* $Kg(n,k) = g(n,k+1) = (n+1)^2 + n(k+1) + (k+1)^3$.

Definition 12. *Let L be an operator on $\mathbb{K}^{\mathbb{N}}$. Then, L is called a **linear recurrence operator** if*

$$L = \sum_{i=0}^{r} a_i N^i \tag{2.4}$$

*where N is the shift operator on $\mathbb{K}^{\mathbb{N}}$. The **order** of L is r if $a_r \neq 0$.*

Example 11.

1. $L = (n+3)N^2 + n(n+2)N + nN^0$ *is a linear recurrence operator of order 2.*
2. $M = (n^3+2n+6)N^4 + N^2 + (n+4)(n+5)(n-3)N + n^3N^0$ *is a linear recurrence operator of order 4.*

The next example shows the effect of linear recurrence operators on some functions.

Example 12.

1. *Let* $L = (n+1)N^2 + (n+3)(n+4)N + (n+5)$. *Then,*
   (a) *Let* $f(n) = n^3 + 2n^2 + 3n + 6$. *Then,* $Lf = (n+1)f(n+2) + (n+3)(n+4)f(n+1) + (n+5)f(n) = n^5 + 14n^4 + 73n^3 + 186n^2 + 276n + 202$.
   (b) *Let* $g(n) = n^2 + 5n + 3$. *Then,* $Lg = (n+1)g(n+2) + (n+3)(n+4)g(n+1) + (n+5)g(n) = n^4 + 16n^3 + 90n^2 + 201n + 140$.
2. *Let* $M = N^3 + (n^3 + 4n^2 + 3n + 9)N$. *Then,*
   (a) $Mf = n^6 + 9n^5 + 33n^4 + 77n^3 + 134n^2 + 168n + 168$ *where f is defined as above.*
   (b) $Mg = n^5 + 11n^4 + 40n^3 + 67n^2 + 101n + 108$, *$g$ is defined as above.*

Notation: Throughout the thesis, we will use the following notation:

1. K : a field of characteristic 0
2. $K^{\mathbb{N}}$: the set of all sequences whose terms are coming from K. In other words, $K^{\mathbb{N}} = \{(a_n)_{n=1}^{\infty} : a_n \in K\}$.

3. Sister Celine Algorithm

In this chapter, the "Sister Celine's Method" will be discussed.

Sister Celine's Algorithm In Nutshell:
Input: A sum with a *nice*(discussed in detailed below , see **Theorem2**) summand
Output: A recurrence satisfied by the summand

Let's clarify the meaining of the recurrence in our context.

Definition 13. *Let $F(n,k)$ be a hypergeometric function in both arguments,i.e $F(n,k)$ is hypergeometric with respect to n and k. We say that $F(n,k)$ satisfies a* ***recurrence with polynomial coefficients*** *if there exist non-negative integers I,J and polynomials $a_{i,j}$'s such that*

$$\sum_{i=0}^{I}\sum_{j=0}^{J} a_{i,j}(n)F(n+j,k+i) = 0. \tag{3.1}$$

Note that the coefficients, $a_{i,j}$'s do not depend on k. Similarly, let $f(n)$ be a hypergeometric function. Then, f(n) is said to satisfy a ***recurrence with polynomial coefficients*** *if there exists a non-negative integer J such that*

$$\sum_{j=0}^{J} a_j(n)f(n+j) = 0. \tag{3.2}$$

Before discussing the details of the input, output and steps of the algorithm, let's do two examples:

Example 13. *Evaluate the sum $\sum_k \binom{n}{k}$.*

Obviously, this sum can be evaluated without using the Sister Celine's algorithm. Even, one may argue that Sister Celine's algorithm is one of the longest way to solve the question. However, we want to see the main points of the algorithm in a clear setting. Also, do not forget that this process is completely algorithmic. In other words, we do not need to think at all to evaluate the sum!

Before looking at the solution, one obvious question is that: Why do we choose this sum? In particular,

1. Why does the summation index run through all integers rather than from 0 to n?

2. Why do we choose $\binom{n}{k}$ rather than, for instance, $n \cdot sin(k)$?

Answers to both questions would be clear once we look at the solution.

Solution:

1. Let's define f(n) := $\sum_k \binom{n}{k}$ and $F(n,k) := \binom{n}{k}$. Our goal is to find a recurrence relation satisfied by $f(n)$ as in (3.2). To do this, first , we will find a recurrence satisfied by $F(n,k)$ as in (3.1). Then, we will use this recurrence to find a recurrence satisfied by $f(n)$. An obvious question is the following: "Suppose we find such a recurrence. Does this mean that we really evaluate the sum?" The answer is no. To be more precise, not yet. Of course, the answer depends on whether we can solve the recurrence or not. We will solve this problem algorithmically in Chapter 6. In other words, in Chapter 6 we will answer the following question: Given a recurrence of the form (3.2), can we find $f(n)$? Let's go back to our solution:

2. Assume that we can find a recurrence of the form (in other words, we take $I = 1$ and $J = 1$ in (3.1))

   $$(3.3)\quad a(n)F(n,k)+b(n)F(n+1,k)+c(n)F(n,k+1)+d(n)F(n+1,k+1)=0.$$

   where the coefficients of $a(n), b(n), c(n), d(n)$ depend only on n, not on k. The reason for this will become clear during the solution. To find these coefficients, we will divide (3.3) by $F(n,k)$ to get

   $$(3.4)\qquad a(n)+b(n)\frac{F(n+1,k)}{F(n,k)}+c(n)\frac{F(n,k+1)}{F(n,k)}+d(n)\frac{F(n+1,k+1)}{F(n,k)}=0.$$

   Now, we can substitute $\binom{n}{k}$ for $F(n,k)$. So, we get

   $$(3.5)\qquad a(n)+b(n)\frac{n+1}{n+1-k}+c(n)\frac{n-k}{k+1}+d(n)\frac{n+1}{k+1}=0$$

   Thus, we have rational functions of n and k as coefficients of $a(n)$, $b(n)$, $c(n)$ and $d(n)$. Note that this not a coincidence. It is a direct consequence of the fact that $F(n,k) = \binom{n}{k}$ is a hypergeometric function with respect to the both arguments. This makes it clear why we don't choose, say, $n \cdot sin(k)$ as a summand.

3. Now, we need to clear the denominators. Because we want to get polynomials rather than rational functions. Multiplying (3.5) by $(n+1-k)(k+1)$ gives (3.6)
$a(n)(n+1-k)(k+1)+b(n)(n+1)(k+1)+c(n)(n-k)(n+1-k)+d(n)(n+1)(n-k+1) = 0$

   Collecting powers of k in the equation (3.6) gives us

$$k^2[-a(n)+c(n)]+$$
$$k^1[na(n)+nb(n)+b(n)-2nc(n)-c(n)-nd(n)-d(n)]+$$
$$k^0[na(n)+a(n)+nb(n)+b(n)+n^2c(n)+nc(n)+n^2d(n)+2nd(n)+d(n)] = 0$$

4. Since, our recurrence must be true for all values of n and k, each power of k must vanish. This gives us system of linear equations as follows:
$$\begin{bmatrix} -1 & 0 & 1 & 0 \\ n & n+1 & -2n-1 & -n-1 \\ n+1 & n+1 & n^2+n & n^2+2n+1 \end{bmatrix} \begin{bmatrix} a(n) \\ b(n) \\ c(n) \\ d(n) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$
   Note that, we have more unknowns than the number of equations. Also, the system is a homogeneous system. Combining these facts, we are guaranteed to find a nontrivial solution!

5. When we solve the system, we get the following as a solution
$\begin{bmatrix} a(n) & b(n) & c(n) & d(n) \end{bmatrix} = d(n) \begin{bmatrix} -1 & 0 & -1 & 1 \end{bmatrix}$

6. This means that, our recurrence is the following

$$-d(n)F(n,k) - d(n)F(n,k+1) + d(n)F(n+1,k+1) = 0 \tag{3.7}$$

   where $d(n) \neq 0$. Now, we need to switch from $F(n,k)$ to $f(n)$ somehow. The trick is to sum (3.7) over k . This trick makes it clear why we choose $a(n)$, $b(n)$, $c(n)$, $d(n)$ independent of k. Also note that, $\sum_k F(n,k+j) = \sum_k F(n,k)$ for any integer j. Therefore, it is advantageous to choose summation range as all integers. Hence, we get $-d(n)f(n) - d(n)f(n) + d(n)f(n+1) = 0$, i.e we get $f(n+1) = 2f(n)$. Also, $f(0) = 1$. Thus, $f(n) = 2^n$. So, we answered the question.

Example 13 shows some characteristic and non-characteristic properties of the method:

1. First, we find a homogeneous system using (3.6). In the system we have more equations than number of unknowns. We will prove that this is always the case for suitable choises of I and J in Definition 13. In other words, maybe some values of I and J gives us systems that does not have non-trivial solutions. However, we can always find a system which has a non-trivial solution. For details see Theorem 2

2. Second, we assume that a recurrence of the form (3.3) exists. It will not be the case always. See Example 14.

3. Third, we are able to solve the problem in hand. It will not be the case always. Obviously, we can solve any problem by hand, since we can mimic the computer's steps. However, it will not be practical at all. See Example 14.

Example 14. *Evaluate the sum* $\sum_k \binom{n}{k}^2$.

Solution:

1. Again, let's define $f(n) := \sum_k \binom{n}{k}^2$ and $F(n,k) := \binom{n}{k}^2$.

   Assume that we have a recurrence of the form (3.3)

   $$a(n)F(n,k)+b(n)F(n+1,k)+c(n)F(n,k+1)+d(n)F(n+1,k+1)=0. \tag{3.8}$$

   Dividing both sides of (3.8) by $F(n,k)$ gives us

   $$a(n)+b(n)\frac{F(n+1,k)}{F(n,k)}+c(n)\frac{F(n,k+1)}{F(n,k)}+d(n)\frac{F(n+1,k+1)}{F(n,k)}=0. \tag{3.9}$$

   Again substitute $F(n,k)=\binom{n}{k}^2$ in (3.9) to get

   $$a(n)+b(n)\frac{(n+1)^2}{(n+1-k)^2}+c(n)\frac{(n-k)^2}{(k+1)^2}+d(n)\frac{(n+1)^2}{(k+1)^2}=0. \tag{3.10}$$

2. Clearing the denominators and collecting terms with respect to power of k

gives the following:

$$\begin{aligned}
k^4\{a(n)+c(n)\}+\\
k^3\{-2na(n)-4nc(n)-2c(n)\}+\\
k^2\{n^2a(n)-2na(n)-2a(n)+n^2b(n)+\\
2nb(n)+b(n)+6n^2c(n)+6nc(n)+\\
c(n)+n^2d(n)+2nd(n)+d(n)\}+\\
k^1\{2n^2a(n)+2na(n)+2n^2b(n)+4nb(n)+\\
2b(n)-4n^3c(n)-6n^2c(n)-2nc(n)-\\
2n^3d(n)-6n^2d(n)-6nd(n)-2d(n)\}+\\
k^0\{n^2a(n)+2na(n)+a(n)+n^2b(n)+\\
2nb(n)+b(n)+n^4c(n)+2n^3c(n)+\\
n^2c(n)+n^4d(n)+4n^3d(n)+\\
6n^2d(n)+4nd(n)+d(n)\}=0
\end{aligned}$$

3. As in the above example, we have a homogeneous system of linear equations:
$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ -2n & 0 & -4n-2 & 0 \\ n^2-2n-2 & n^2+2n+1 & 6n^2+6n1 & n^2+2n+1 \\ 2n^2+2n & 2n^2+4n+2 & -4n^3-6n^2-2n & -2n^3-6n^2-6n-2 \\ n^2+2n+1 & n^2+2n+1 & n^4+2n^3+n^2 & n^4+4n^3+6n^2+4n+1 \end{bmatrix} \begin{bmatrix} a(n) \\ b(n) \\ c(n) \\ d(n) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

4. Solving this linear system gives us $\begin{bmatrix} a & b & c & d \end{bmatrix} = d\begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$ In other words, there is no non-trivial solution for this system! Hence, Sister Celine's algorithm cannot find a first order recurrence satisfied by $F(n,k)$. What went wrong? The problem is that in our homogeneous system the number of equations does **not** exceed the number of unknowns. In our case, due to this fact we do not have any non-trivial solutions. What is the remedy? Trying a recurrence with larger order than (3.8). In other words, we will update our assumption as(i.e we take $I = 2$ and $J = 2$):

(3.11)
$$a(n)F(n,k)+b(n)F(n+1,k)+c(n)F(n,k+1)+d(n)F(n+1,k+1)+e(n)F(n+2,k)+$$
$$g(n)F(n+2,k+1)+h(n)F(n+2,k+2)+l(n)F(n,k+2)+m(n)F(n+1,k+2)=0.$$

We need to repeat steps (1),(2) and (3) until we get a system which has a nontrivial solution. Obviously, this becomes extremely messy to do by hand. Thus, we can use computer to do these computations for us. The following Maple code shows that we cannot find a recurrence of the form (3.8) : $celine((n,k)->\frac{n!^2}{(k)!^2\cdot(n-k)!^2},1,1)$. Maple outputs: $''0=0''$. In other words, it cannot find a nontrivial recurrence when $I=J=1$. Hence, we should try to find a bigger recurrence:

$celine((n,k)->\frac{n!^2}{(k)!^2\cdot(n-k)!^2},2,2)$.
Then, Maple find, the following recurrence:

$$(3.12)\quad (n-1)F(n-2,k-2)+(2-2n)F(n-2,k-1)+nF(n,k)$$
$$(1-2n)F(n-1,k-1)+(n-1)F(n-2,k)+(1-2n)F(n-1,k)=0$$

5. Summing (3.12) over all integers k gives us

$$(2-4n)f(n-1)+nf(n)=0.$$

Thus, we have

$$\frac{f(n)}{f(n-1)}\frac{f(n-1)}{f(n-2)}\cdots\frac{f(1)}{f(0)}=\frac{(2n-1)(2n-2)}{n(n-1)}\frac{(2n-3)(2n-4)}{(n-1)(n-2)}\cdots\frac{2}{1}=\frac{2n!}{n!^2}.$$

6. Hence, we get the well-known $\sum_k \binom{n}{k}^2=\binom{2n}{n}$.

Obviously, we can try Sister Celine's algorithm for $\binom{n}{k}^3$ or $\binom{n}{k}^4$ etc. In principle, the method is very much the same. However, in practice these are almost impossible to do by hand.

Remark 17. *In general, when we want to use Sister Celine's algorithm to find the recurrence satisfied by $\sum_k F(n,k)$ in Maple, we write the following code: celine(F(n,k) , a , b) where $F(n,k)$ is our summand , a is a positive integer represents I in Definition 13 and b is a positive integer which represent J in Definition 13.*

Remark 18. *In example 14, we see that Sister Celine's algorithm cannot find a recurrence of order 1. Thus a reasonable question to ask is: Suppose Sister Celine's algorithm cannot find a recurrence of order r satisfied by a sum $f(n)=\sum_k F(n,k)$. Can we conclude that $f(n)$ does not satisfy a recurrence of order r or less? The answer is no in general. As a result, we cannot use Sister Celine's algorithm to prove that a sum does not satisfy a recurrence of order r.*

The following questions are in order:

1. What happens if we cannot find a recurrence as assumed?

2. Suppose we can find a recurrence for $F(n,k)$. Thus, we can find a recurrence for $f(n)$ as well. How can we find $f(n)$ using these recurrence?

3. How fast is this algorithm?

4. Which properties must be satisfied by $F(n,k)$ to execute the algorithm?

5. What happens if we have definite sum rather than indefinite sum?

Answers:

1. This is not a possibility as shown in the Theorem 2 , actually that is one of the advantages of the Sister Celine's algorithm. To be more precise, for some values of I and J in Definition 13 maybe we cannot find a recurrence. However, as Theorem 2 shows there exists suitable values of I and J such that, a recurrence of the form Definition 13 exists.(Actually, we are kind of lying here. The summand should be a proper hypergeometric term to conclude that we can find suitable I and J such that Sister Celine's algorithm finds a recurrence satisfied by the sum.)

2. This question is addressed in Chapter 6.

3. Unfortunately, it is very slow.(We will not discuss exact complexity of the algorithm). Fortunately, it turns out that we have another algorithm, called Zeilberger's algorithm, which does exactly the same job as Sister Celine and much faster. For more information, see Chapter 5.

4. $F(n,k)$ must be a proper hypergeometric term. See, Definition 6. To be more precise, if $F(n,k)$ is a proper hypergeometric term, then we can find a recurrence satisfied by $f(n) = \sum_k F(n,k)$, i.e it is a sufficient condition not a necessary one!

5. Then, Sister Celine is not enough. However, in Chapter 4 we will see that another algorithm , Gosper's Algorithm exists precisely for this job.

Remark 19. *1. Let's examine the first question more carefully: "How can we guarantee that there is a solution for the recurrence relation?" Obvious answer is, there must exist a recurrence since we can always choose coefficients of $F(n+j,k+i)'s$ equal to 0 in (3.1). Of course it is a useless recurrence. So the main question is that:"Can we always find a nontrivial recurrence?" If we look at Example 13 and Example 14, we can find nontrivial solutions because the number of unknowns exceed the number of equations. Thus, if we can guarantee that this always must be the case, we are done. In fact, by the following theorem, we are guaranteed to find a nontrivial solution to our homogeneous* *system. To sum up, it is possible that for some choices of I and J in Definition 13, no non-trivial recurrence exists for $F(n,k)$. According to Theorem 2 we can choose I and J in a way that a non-trivial recurrence exists.*

2. *A note for second question is also needed: "How do we know that $\sum_k F(n,k) = \sum_k F(n,k+j)$ for any natural number j?" A possible answer(the one we will use as well) is compact support. In other words, if we know that $F(n,k) = 0$ for all but finitely many values of k, then it is indeed true. Thus, compact support is also needed for us. The good thing is that if we have a binomial coefficient of a certain type(see Chapter2), then $F(n,k)$ has compact support. For more information see Definition 7.*

3. *Sister Celine's Algorithm is very slow and there is another algorithm which does the exact same job. Thus, why do we care about this algorithm? The answer is two-fold: First, Sister Celine is the first person who showed that we can attack the problem algorithmically in her paper Fasenmyer (1949). Thus, this algorithm has an historical importance. Second, the algorithm works, is easy to understand , is easy to describe and Theorem 2 is essential for creative telescoping algorithm to work. See, Chapter 5.*

If $F(n,k)$ does not have a compact support, then we are in trouble because of the following reasons:

1. First, there is a problem about convergence of the sum. In other words, the sum may diverge. For example, $\sum_k k$ does not make sense.
2. Second, there may not exist a recurrence for $F(n,k)$ at all. For instance, if $F(n,k) = \frac{1}{n^2+k^2+1}$, then there does not exist a recurrence satisfied by $F(n,k)$ which has polynomial coefficients. where rf and ff are short-hand for raising factorials and falling factorials, respectively

Now, we come to the theorem which makes Sister Celine's Algorithm completely settle the summation problem, at least in theory.

Theorem 2. *Let F(n,k) be a proper hypergeometric term. Then, F satisfies a k-free recurrence relation. In other words, there exits positive integers I and J and polynomials $a_{i,j}(n)$ for $i = 0,1,..I$ and $j = 0,1,..J$, not all zero such that the recurrence*

$\sum_{i=0}^{I} \sum_{j=0}^{J} a_{i,j}(n) F(n+j,k+i) = 0$

holds at every point (n,k) at which $F(n,k) \neq 0$ and all of the values of F that occur in above equation are well defined.

Before beginning the proof, we will look at the behaviour of translates of a proper hypergeometric term:
(1) Suppose $f(n) = (3n+2)!$. We want to find $\frac{f(n-2)}{f(n)}$. Obviously,
$\frac{f(n-2)}{f(n)} = \frac{1}{(3n+2)(3n+1)(3n)(3n-1)(3n-2)(3n-3)}$.

(2) Suppose $f(n) = (2-3n)!$. Then,
$\frac{f(n-2)}{f(n)} = (8-3n)(7-3n)(6-3n)(5-3n)(4-3n)(3-3n)$.

We can generalize these examples as follows: Let $f(n) = (an+b)!$, we want to compute $\frac{f(n-j)}{f(n)}$ where $j \geq 0$. Then, we have 2 cases:
Case 1: If $a \leq 0$ then $\frac{f(n-j)}{f(n)}$ is a polynomial in n.
Case 2 : If $a > 0$ then $\frac{f(n-j)}{f(n)}$ is a reciprocal of a polynomial in n.

Similarly, let's look at the translates of proper hypergeometric terms with 2 variables. In other words, we will consider functions like $f(n,k) = (3n+2k+1)!$.

1. Suppose $f(n,k) = (-3n+2k+1)!$. Then,
   (a) $\frac{f(n+2,k+1)}{f(n,k)} = \frac{1}{(2k-3n-2)(2k-3n-1)(2k-3n)(2k-3n+1)}$.
   (b) $f(n+1,k+2) = (2k-3n+2)$.
2. Suppose $g(n,k) = (5n-4k-3)!$. Then,
   (a) $\frac{g(n+4,k+4)}{g(n,k)} = \frac{(5n-4k-2)(5n-4k-1)(5n-4k)(5n-4k+1)}{1}$.
   (b) $\frac{g(n+1,k+2)}{g(n,k)} = \frac{1}{(5n-4k-5)(5n-4k-4)(5n-4k-3)}$.

Generalizing these examples like above gives us the following: Let $F(n,k) = (an+bk+c)!$. Then,
Case1: If $aj+bi \geq 0$ $\frac{F(n+j,k+i)}{F(n,k)} = (an+aj+bk+bi+c)(an+aj+bk+bi+c-1)..(an+bk+c+1)$
Case2: If $aj+bi \leq 0$ $\frac{F(n+j,k+i)}{F(n,k)} = \frac{1}{(an+bk+c)(an+bk+c-1)..(an+bk+c-aj-bi+1)}$

Lastly, we introduce two notations to simplify our task in the proof. Let $rf(e,p) := \prod_{j=1}^{e}(p+j)$ and $ff(e,p) := \prod_{j=1}^{e-1}(p-j)$ where rf and ff are short-hand for raising factorials and falling factorials, respectively. Using our new notation, case1 and case2 above become

Case1 : If $aj+bi \geq 0$ $\frac{F(n+j,k+i)}{F(n,k)} = rf(aj+bi, an+bk+c)$.
Case2 : If $aj+bi \leq 0$ $\frac{F(n+j,k+i)}{F(n,k)} = \frac{1}{ff(|aj+bi|, an+bk+c)}$

Sketch of the Proof. *The idea of the proof is the following:*

1. *We start as in the examples, i.e assuming that there exists a recurrence of the form* (3.1)*. The only difference is that I and J will not be fixed this time. In the examples, we start as initializing $I = J = 1$.*

2. *Then, we divide the* (3.1) *by $F(n,k)$ as in the examples. We should get rational functions of n and k as coefficients of $a_{i,j}$'s. Using the fact that $F(n,k)$ is a hypergeometric term allows us to show that it is indeed the case.*

3. *Following steps of the examples, we collect everything on a common denominator. Then, clear the denominators. Since $F(n,k)$ is a proper hypergeometric term we can precisely determine the form of the equation.*

4. *The important point is that: We should be able to solve the existing linear system for suitable choices of I and J. In other words, we need to show that the number of equations exceeds number of unknowns. To do this, we will write the number of equations and number of unknowns in terms of I and J. Then, we will show that the former grows faster than latter. In other words, there exists I and J such that the number of equations exceeds the number of unknowns.*

Let's look at the detailed proof.

Proof. 1. Suppose $F(n,k)$ is a proper hypergeometric term. In other words, $F(n,k)$ can be written as

$$F(n,k) = P(n,k)\frac{\prod_{s=1}^{m}(a_s n + b_s k + c_s)!}{\prod_{s=1}^{t}(u_s n + v_s k + w_s)!}.$$

Assume that we have a recurrence of the form (3.1). Let's look at the ratio $R(n,k) = \frac{F(n+j,k+i)}{F(n,k)}$. From the observations above, R must be a rational function of n and k, say $R(n,k) = \frac{R1(n,k)}{R2(n,k)}$.

2. More precisely, we get

$$(3.13)\quad R1(n,k) = P(n+j,k+i) \prod_{\substack{s=1 \\ a_sj+b_si\geq 0}}^{m} rf(a_sj+b_si, a_sn+b_sk+c_s) \prod_{\substack{s=1 \\ u_sj+v_si\leq 0}}^{t} ff(|u_sj+v_si|, u_sn+v_sk+w_s)$$

Similarly,

$$(3.14)\quad R2(n,k) = P(n,k) \prod_{\substack{s=1 \\ a_sn+b_sk+c\leq 0}}^{m} ff(|a_sj+b_si|, a_sn+b_sk+c_s) \prod_{\substack{s=1 \\ u_sj+v_sk+w_s\geq 0}}^{t} rf(u_sj+v_si, u_sn+v_sk+w_s)$$

Dividing the assumed recurrence by $F(n,k)$ gives us

$$(3.15)\quad \sum_{\substack{0\leq i\leq I \\ 0\leq j\leq J}} a_{i,j}(n) \frac{R1_{i,j}(n,k)}{R2_{i,j}(n,k)}.$$

where $a_{i,j}$'s are polynomials of n we are looking for, $R1(n,k)$ and $R2(n,k)$ are of the form (3.13) and (3.14), respectively.

3. The next step is to collect all terms in (3.15) over a common denominator. Looking at (3.14) we know that, $P(n,k)$ always appears in the denominator. This means that our common denominator for (3.15) must contain $P(n,k)$. Since we know the exact form of the denominator, i.e $R2(n,k)$, we can find the exact form of the common denominator. Obviously, to simplify our task, we want to find the least common multiple of all $R2_{i,j}(n,k)$. Then, for each s, we want to find falling factorials whose first argument is the largest. Similarly, for the rising factorials. Hence, we have

$$\prod_{s=1}^{m} ff((a_s^+ J + (b_s^+ I, a_sn+b_sk+c_s) \prod_{s=1}^{t} rf((-u_s)^+ J + (-v_s)^+ I, u_sn+v_sk+w_s$$

as our least common denominator where $a^+ := max(a,0)$ for any real number

a. Now, our equation becomes

$$\sum_{\substack{0\le i\le I\\0\le j\le J}} a_{i,j}\frac{R1_{i,j}(n,k)}{R2i,j(n,k)}CD(n,k), \tag{3.16}$$

where CD(n,k) represents the common denominator of (3.15)

4. To finish the proof we need to show that we have a suitable choice of I and J that gives us a linear system which has a non-trivial solution. Equivalently, we need to show that number of equations exceeds the number of unknowns in (3.16) if we choose I and J big enough. It is obvious that we have $(J+1)(I+1)$ unknowns. Number of powers of k is $c_1J+c_2I+c_3$ for constants c_1 , c_2 and c_3. In other words, number of unknowns grows like IJ whereas number of powers of k grows like $I+J$. It means that, the number of unknowns grows quadraticly and number of powers of k grows linearly. Thus, we are done.

□

Let's try Sister Celine's algorithm on another example.

Example 15. *Evaluate the sum* $\sum\limits_{k}\binom{n}{k}\frac{1}{n^2+k^2+1}$.

Solution:
(i) Let $F(n,k)=\binom{n}{k}\frac{1}{n^2+k^2+1}$. Assume that we can find a recurrence of the form (3.1) satisfied by $F(n,k)$. In other words, we have

$$\sum_{i=0}^{I}\sum_{j=0}^{J}\frac{a_{i,j}(n)\binom{n+i}{k+j}}{(1+(n+i)^2+(k+j)^2)\binom{n}{k}}=0 \tag{3.17}$$

(ii) Thus, at the left hand side of (3.17), we have a rational function with respect to k. Any rational function can be uniquely determined by finite set of points. Also not all $a_{i,j}$'s are identically zero. Suppose we take one of the complex poles of one of the summands, say $k=\alpha$. Then all other summands are finite at $k=\alpha$. Thus, when $k=\alpha$, we get a contradiction, since on the left hand side we have ∞ and on the right hand side we have 0. Hence, (3.17) cannot be valid.

Moral of the Story: Even when we have a hypergeometric summand with compact support, it is not enough to conclude that we can find a recurrence satisfied by the summand. So, proper hypergeometric term is essential for us.

From theoretical perspective, there is no need to find values of I and J beforehand. Since by Theorem 2 we know that we can find suitable I and J values. However, from the practical perspective, it is important to be able to determine values of I and J. Another practically important question is that: In the $5th$ step of Sister Celine's Algorithm we increase I and J by 1. Is it possible to increase only one of them and try to find a smaller recurrence satisfied by $F(n,k)$? We will not answer these question. For more information see Koepf (1998).

There is also generalization of Sister Celine's Algorithm to multivarite and q-cases. In other words, there exists a more generalized version of Theorem 2 which works for sums over several summation indices, and to $q-$ and multi-q-sums. For more information see **?**

Steps of Sister Celine's Algorithm:

1. Fix $I = 1$ and $J = 1$ where I and J are same as Theorem 2
2. Assume we have a recurrence of the form

$$a(n)F(n,k)+b(n)F(n+1,k)+c(n)F(n,k+1)+d(n)F(n+1,k+1)=0.$$

3. Divide the both sides of the recurrence by $F(n,k)$, simplify the ratios of binomial coefficients, factorials etc. until we have only rational functions of n and k left.
4. Clear the denominator, then collect terms of the resulting expression with respect to powers of k.
5. Solve the resulting linear system of equations by equating the coefficient of each power of k to 0. If this is not possible, then increase I and J until we can find a recurrence.
6. Once we find the coefficients $a_{i,j}(n)$'s in Theorem 2, we can sum the recurrence over k , to find a recurrence satisfied by $f(n)$.

As noted above, due to Theorem 2 this algorithm must stop. Now, let's look at the definite summation problem.

4. Gosper's Algorithm

In this chapter, the "Gosper's Algorithm" will be discussed.

Gosper's Algorithm In Nutshell:
Input: A **definite** sum with a *nice* summand
Output: Evaluation of that sum or informing us that this sum cannot be evaluated by **Gosper's algorithm**

Before discussing the Gosper's algorithm, it makes sense to compare this algorithm with Sister Celine's algorithm:

Sister Celine's Algorithm vs Gosper's Algorithm		
	Input	Output
Sister Celine	$f(n) = \sum\limits_k F(n,k)$ where $F(n,k)$ is hypergeometric	Recurrence with polynomial coefficients satisfied by f(n)
Gosper	$g(n) = \sum\limits_{k=0}^{n} t_k$ where t_k is hypergeometric	Evaluation of that sum , if $g(n)$ is Gosper summable. Otherwise, proving that t_k is not Gosper summable.

To sum up, Sister Celine's algorithm is used for indefinite sums, Gosper's algorithm is used for definite sums. Another difference is that, Sister Celine's algorithm does not evaluate the sum, it gives a recurrence satisfied by the sum. Using algorithm hyper in Chapter 6 we can solve the recurrence. On the other hand, Gosper's algorithm evaluates the sum if the sum is Gosper summable, otherwise it shows(proves!) that the sum is not Gosper summable.

As we can see from Remark 3 and Example 7 in Chapter 2, sum of hypergeometric functions may or may not be hypergeometric. The aim of Gosper's algorithm is to distinguish the sums which can be written as hypergeometric closed form from the others. Before discussing the details of the input, output and steps of the algorithm, let's look at some examples:

Example 16. *Evaluate the sum* $S_n = \sum\limits_{k=0}^{n} k \cdot \frac{k!}{(2k)!}$.

Solution:

1. First, let's call the summand t_n, i.e

$$t_n = n \cdot \frac{n!}{(2n)!}.$$

2. Second, we need to compute the term ratio, call it $r(n)$, i.e

$$r(n) = \frac{t_{n+1}}{t_n} = \frac{(n+1) \cdot \frac{(n+1)!}{(2n+2)!}}{n \cdot \frac{n!}{(2n)!}} = \frac{n+1}{2n(2n+1)}.$$

3. Third, we need to find polynomials $a(n)$, $b(n)$ and $c(n)$ such that $r_n = \frac{a(n)}{b(n)} \cdot \frac{c(n+1)}{c(n)}$ and they satisfy properties of Theorem 4. In other words, we want to find the canonical form for the rational function $r(n)$, see Definition 10 in Chapter 2. Obviously, we can choose $a(n) = 1$, $b(n) = 2(2n+1)$ and $c(n) = n$. (The mechanical(algorithmic) way to tackle this problem will be discussed below.)

4. Now consider the equation

$$a(n)x(n+1) - b(n-1)x(n) = c(n). \tag{4.1}$$

Thus, (4.1) becomes:

$$x(n+1) - 2(2n-1)x(n) = n.$$

Since on both sides of the equations we have a polynomial in n degrees must match. In other words, $deg(x(n)) = 0$ if such a $x(n)$ exists. Writing $x(n) = c$ gives no solution. Thus, this sum cannot be written as a sum of hypergeometric term plus a constant. Equivalently, the summand $k\frac{k!}{(2k)!}$ is not Gosper summable.

There are obvious questions come from Example 16:

1. Can we always write the term ratio, r_n in the above, as $\frac{a(n)}{b(n)} \cdot \frac{c(n+1)}{c(n)}$? In other words, can we always find the canonical form of the rational function r_n? See,

Definition 10 in Chapter 2.

2. Suppose we find $x(n)$ above. So what?
3. We wrote $deg(x(n))$ in the above solution. So we are hinting that $x(n)$ is a polynomial. How do we know?
4. In the above example we could not find $x(n)$ that satisfies (4.1) . Does this really mean that this sum cannot be written as a sum of hypergeometric term plus a constant?
5. How can we search for an $x(n)$ that satisfies (4.1) in a systematic manner? In other words, if we can determine degree of $x(n)$, then we just need to solve a system of linear equations. Can we determine the degree of $x(n)$?
6. Which properties of the summand is needed guarantee that Gosper's algorithm will work?

Before answering this questions. Let's look at another example.

Example 17. *Evaluate the sum* $\sum_{k=0}^{n-1} k^4$.

Solution:
(i) First, let

$$t_n = n^4.$$

Let's compute term ratio:

$$r(n) = \frac{t_{n+1}}{t_n} = \frac{(n+1)^4}{n^4}.$$

(ii) Second, we try to write term ratio as $r(n) = \frac{a(n)}{b(n)} \cdot \frac{c(n+1)}{c(n)}$. Obviously, we can choose $a(n) = 1$, $b(n) = 1$ and $c(n) = n^4$.
(iii) Now, we try to find $x(n)$ such that (4.1) holds. Substituting $a(n)$, $b(n)$ and $c(n)$ gives us

$$x(n+1) - x(n) = n^4. \tag{4.2}$$

Degree of right hand side must be same as degree of left hand side in (4.2). Since leading terms of $x(n+1)$ and $x(n)$ would be same, they cancel each other. That means that, $deg(x(n)) = deg(n^4) + 1 = 5$. So, we can substitute a general degree 5 polynomial instead of $x(n)$. Say, $x(n) = y_5 n^5 + y_4 n^4 + y_3 n^3 + y_2 n^2 + y_1 n + y_0$.

(iv) Let's substitute this in our equality, to get

$$
\begin{aligned}
&n^4(5y_5)+\\
&n^3(4y_4+10y_5)+\\
&n^2(3y_3+6y_4+10y_5])+\\
&n^1(2y_2+3y_3+4y_4+5y_5)+\\
&4n^0(y_1+y_2+y_3+y_4+y_5)=n^4
\end{aligned}
\tag{4.3}
$$

Thus, if we equate the coefficients of same power of n in both sides we get a linear system. Solving this linear system gives us $[y_5,y_4,y_3,y_2,y_1,y_0]=[\frac{1}{5},\frac{-1}{2},\frac{1}{3},0,\frac{-1}{30},0]$. In other words, $x(n)=\frac{n^5}{5}-\frac{n^4}{2}+\frac{n^3}{3}-\frac{n}{30}$.
(v) Then computing, $\frac{b(n-1)x(n)}{c(n)}t_n$ gives us the evaluation of the sum $\frac{n^5}{5}-\frac{n^4}{2}+\frac{n^3}{3}-\frac{n}{30}$. Thus,we are done.

Remark 20. *Note that in Chapter 2 we promised to show that if t_k is a polynomial, then t_k is Gosper summable, i.e we need to show that there exists a hypergeometric function d_n such that $d_{n+1}-d_n=t_n$. In fact, we will show that a polynomial d_n always exists! Actually, an easy generalization of the previous example shows(proves!) that every polynomial in one variable is Gosper summable. Observe that:*

1. *If t_n is a polynomial of degree d, then the term ratio is a ratio of two polynomials of degree d.*
2. *When we write $r(n)=\frac{t_{n+1}}{t_n}$ in canonical form, we find that $a=b=1$ and $c(n)=t_n$.*
3. *Thus, we get a equation of the form*

$$x(n+1)-x(n)=c(n)$$

   as in the previous example. This means that $deg(x)=deg(c)+1=d+1$.

4. *When we plug in a generic polynomial of degree $d+1$ instead of $x(n)$ and try to solve the corresponding system of equations, then obviously it will always have a solution due to lower-triangular form of the equations as in (4.3). Thus, a solution $x(n)$ exists. Hence, t_k is Gosper summable.*

Remark 21. *Note that the above remark gives us an interesting way to evaluate $\sum_{k=0}^{n} k^4$ as well! Due to above arguments we know that the answer would be a degree 5 polynomial of n. Thus, we can do the following:*

1. *Find value of the sum in $5+1=6$ different points say $0,1,2,3,4,5$.*

2. *Find the(yes it is unique!) polynomial of degree 5 which goes through that points.*

In action, we have

$$\sum_{k=0}^{0} k^4 = 0,$$
$$\sum_{k=0}^{1} k^4 = 1,$$
$$\sum_{k=0}^{2} k^4 = 17,$$
$$\sum_{k=0}^{3} k^4 = 98,$$
$$\sum_{k=0}^{4} k^4 = 354,$$
$$\sum_{k=0}^{5} k^4 = 979.$$

Now let $\sum_{k=0}^{n} k^4 = a_5n^5 + a_4n^4 + a_3n^3 + a_2n^2 + a_1n^1 + a_0$. *Thus, we need to find the polynomial which goes through* $(0,0),(1,1),(2,17),(3,98),(4,337)$ *and* $(5,962)$. *In other words, we need to solve*

$$\begin{aligned} a_0 &= 0 \\ a_5 + a_4 + a_3 + a_2 + a_1 + a_0 &= 1 \\ 32a_5 + 16a_4 + 8a_3 + 4a_2 + 2a_1 + a_0 &= 17 \\ 243a_5 + 81a_4 + 27a_3 + 9a_2 + 3a_1 + a_0 &= 98 \\ 1024a_5 + 256a_4 + 64a_3 + 16a_2 + 4a_1 + a_0 &= 354 \\ 3125a_5 + 625a_4 + 125a_3 + 25a_2 + 5a_1 + a_0 &= 979. \end{aligned}$$

Solving the system, surely, gives $[a_0, a_1, a_2, a_3, a_4, a_5] = [0, \frac{-1}{30}, 0, \frac{1}{3}, \frac{1}{2}, \frac{1}{5}]$.

Example 18. *Let's try another example, which is not easy to evaluate by hand. Evaluate* $\sum_{k=0}^{n} \frac{k^4 \cdot 4^k}{\binom{2k}{k}}$.

(i) Again, we start by defining the summand and calculating the term ratio:

$$t_n = \frac{n^4 \cdot 4^n}{\binom{2n}{n}},$$

and

$$r(n) = \frac{t_{n+1}}{t_n} = \frac{2(n+1)^5}{n^4 \cdot (2n+1)}.$$

(ii) The next step is to write $r(n)$ as $r(n) = \frac{a(n)}{b(n)} \cdot \frac{c(n+1)}{c(n)}$. Observe that choosing $a(n) = 2(n+1)$, $b(n) = 2n+1$ and $c(n) = n^4$ gives us the desired canonical form.
(iii) Thus, we need to find $x(n)$ in (4.1) i.e

$$2(n+1)x(n+1) - (2n-1)x(n) = n^4.$$

Since degree of both sides must match, $deg(x(n)) = deg(c(n)) - deg(a(n)) + 1 = 4$. Hence, we need to substitute a generic degree 4 polynomial instead of $x(n)$.
(iv) As a result we get

$$\begin{aligned} n^4(11x_4)+ \\ n^3(9x_3+20x_4)+ \\ n^2(7x_2+12x_3+20x_4)+ \\ n^1(5x_1+6x_2+8x_3+10x_4)+ \\ n^0(3x_0+2x_1+2x_2+2x_3+2x_4) = n^4. \end{aligned}$$

Solving these gives, $[x_4, x_3, x_2, x_1, x_0] = [\frac{1}{11}, \frac{-20}{99}, \frac{20}{231}, \frac{26}{693}, \frac{-2}{231},$ i.e $x(n) = \frac{1}{11}n^4 - \frac{20}{99}n^3 + \frac{20}{231}n^2 + \frac{26}{693}n - \frac{2}{231}$.
(v) Writing this $x(n)$ in $\frac{b(n-1)\cdot x(n)}{c(n)}t_n$ gives us $\frac{(2n-1)\cdot(63n^4-140n^3+60n^2+26n-6)4^n}{693\binom{2n}{n}} - \frac{2}{231}$ as our answer.

Now, we can answer the questions we ask just after solving example 16:

1. We can always write a rational function $f(n)$ in canonical form. In other words, we can find polynomials $a(n)$, $b(n)$ and $c(n)$ such that conditions of Definition 10 in Chapter 2 is satisfied.

2. If we can find $x(n)$ that satisfies (4.1), then the sum is Gosper summable. Thus, we can easily evaluate it , see Remark 9. To show that the sum is Gosper summable, it suffices to find a hypergeometric function z_n such that $t_n = z_{n+1} - z_n$. We claim that $z_n = \frac{b(n-1)x(n)}{c(n)}t_n$.

Proof of the claim:

$$z_{n+1}-z_n = \frac{b(n)x(n+1)t_{n+1}}{c(n+1)} - \frac{b(n-1)x(n)t_n}{c(n)}$$
$$= t_n(\frac{b(n)x(n+1)t_{n+1}}{c(n+1)t_n} - \frac{b(n-1)x(n)}{c(n)})$$
$$= t_n(\frac{b(n)x(n+1)a(n)c(n+1)}{c(n+1)b(n)c(n)} - \frac{b(n-1)x(n)}{c(n)})$$
$$= t_n(\frac{x(n+1)a(n)-b(n-1)x(n)}{c(n)})$$
$$= t_n.$$

3. See Theorem 3.
4. Yes, see Theorem 5.
5. See, Remark 24.
6. The summand must be a hypergeometric function. However, it is only a necessary condition, not a sufficient one as shown in Examples 19 and 20.

In Chapter 2 we claim that neither $\sum\limits_{k=0}^{n} k!$ nor $\sum\limits_{k=0}^{n} \frac{1}{k}$ are Gosper summable. Now, we will prove these claims.

Example 19. *Check whether* $\sum\limits_{k=0}^{n} k!$ *is Gosper summable or not.*

Solution:

1. Let $t_n = n!$. Then, the term ratio becomes $r(n) = \frac{t_{n+1}}{t_n} = \frac{(n+1)!}{n!} = n+1$.
2. Then, obviously, we can choose $a(n) = n+1$, $b(n) = 1$ and $c(n) = 1$.
3. Our equation becomes $(n+1)x(n+1) - x(n) = 1$. Obviously, there is no polynomial solution for equation. Thus, it is not Gosper summable. Thus, we cannot write $k!$ as hypergeometric term plus a constant.

Example 20. *Check whether* $\sum_{k=0}^{n} \frac{1}{k}$ *is Gosper summable or not.*

Solution:

1. Let $t_n = \frac{1}{n}$. Then, the term ratio becomes $r(n) = \frac{t_{n+1}}{t_n} = \frac{n}{n+1}$.
2. It is clear that we can choose $a(n) = n$, $b(n) = n+1$ and $c(n) = 1$.
3. Then, our equation becomes $nx(n+1) - nx(n) = 1$. Thus, degree of $x(n)$ is 0. However, plugging $x(n) = c$ gives us $1 = 0$. Thus, this equation has no

polynomial solution. As a result, it is not Gosper summable. Hence, we cannot write $\frac{1}{k}$ as hypergeometric term plus a constant.

Remark 22. *A useful way to check the correctness of our evaluation of a sum using Gosper's algorithm is the following: We will give the hypergeometric function d_n in the Definition 9 as a proof. Thus, one just need to check whether $t_n = d_{n+1} - d_n$ or not.*

Before looking at the steps of the algorithm and the proofs that shows the correctness of this algorithm, let's try to understand the problem in abstract terms. In other words, we are in the following situation: We are given a sum of the form $\sum\limits_{k=m}^{n} t_k$ where t_k is a hypergeometric function of k and m is fixed. We want to evaluate the sum. Let's suppose we can find d_n such that $d_{n+1} - d_n = t_n$ and $d(n)$ is a hypergeometric function. Then,

$$\sum_{k=m}^{n} t_k = \sum_{k=m}^{n} d_{k+1} - d_k = [d_{m+1} - d_m] + [d_{m+2} - d_{m+1}] + ...[d_{n+1} - d_n] = d_{n+1} - d_m$$

Then, we do write the sum as a sum of a hypergeometric term and a constant. Since, d_m is a constant and d_n is a hypergeometric term. Now, we can write the aim of Gosper's Algorithm in precise terms: Given a sum of the form $\sum\limits_{k=m}^{n} t_k$. Gosper's algorithm finds d_n as above if such $d(n)$ exists. Else, it shows that the summand cannot be written as sum of a hypergeometric term and a constant. Equivalently, Gosper's algorithm completely solves the problem: Whether a definite sum with hypergeometric summand is Gosper summable or not.

Theorem 3. *Let $a(n)$, $b(n)$ and $c(n)$ be polynomials in Definition 10. If $x(n)$ is a rational function satisfying the equation* (4.1)*, then $x(n)$ is a polynomial in n.*

Proof.

1. Let $x(n)$ be a rational function of n , i.e $x(n) = \frac{r_1(n)}{r_2(n)}$ where $r_1(n)$ and $r_2(n)$ are polynomials such that $gcd(r_1(n), r_2(n)) = 1$. We need to show that $r_2(n)$ is a constant polynomial. In other words, $x(n)$ is a polynomial.

2. Substituting $x(n) = \frac{r_1(n)}{r_2(n)}$ in equation (4.1) gives

$$a(n)r_1(n+1)r_2(n) - b(n-1)r_1(n)r_2(n+1) = c(n)r_2(n)r_2(n+1). \tag{4.4}$$

For the sake of contradiction, assume that $x(n)$ is not a polynomial, i.e $r_2(n)$ is not a constant polynomial. Let N be the largest integer such that $gcd(r_2(n), r_2(n+N))$ is a non-constant polynomial. Note that, first, $N \geq 0$ since $gcd(r_2(n), r_2(n))$ is a non-constant polynomial by our assumption. Secondly, N is well-defined since polynomials have bounded degree. Let $d(n)$ be an irreducible common divisor of $r_2(n)$ and $r_2(n+N)$. Thus, $d(n)$ divides both $r_2(n)$ and $r_2(n+N)$. It follows that $d(n-N)$ divides $r_2(n)$ as well.

3. As a result $d(n-N)$, divides right hand side of (4.4), also it divides the first term ($a(n)r_1(n+1)r_2(n)$) on the left hand side. Thus, it must divide the second term in the left hand side. Hence, $d(n-N)$ divides $b(n-1)r_1(n)r_2(n+1)$.

4. Note that $d(n-N)$ does not divide $r_1(n)$ since by our assumption $r_1(n)$ and $r_2(n)$ are relatively prime. Similarly, $d(n-N)$ does not divide $r_2(n+1)$. Otherwise, $d(n)$ divides $r_2(n+N+1)$ which contradicts the maximality of N. Combining both of these facts we have $d(n-N)$ divides $b(n-1)$, equivalently $d(n+1)$ divides $b(n+N)$.

5. With a similar reasoning, $d(n+1)$ must divide $a(n)r_1(n+1)r_2(n)$. It is clear that, $d(n+1)$ does not divide $r_1(n+1)$ since this implies that $d(n)$ divides $r_1(n)$ which contradict relatively primeness of $r_1(n)$ and $r_2(n)$. Similarly , $d(n+1)$ does not divide $r_2(n)$ as , otherwise maximality of N leads to a contradiction. As a result, $d(n+1)$ divides $a(n)$.

6. However, $d(n+1)$ divides $a(n)$ and $d(n+1)$ divides $b(n+N)$ contradicts our assumption that $gcd(a(n),b(n+h))=1$ for all positive integers h. This shows that $r_2(n)$ is indeed a constant polynomial. Therefore $x(n)$ is a polynomial, as well.

□

Theorem 4. *Let K be a field of characteristic zero and $r(n) \in K(n)$ be a nonzero rational function. Then, there exists polynomials $a(n)$, $b(n)$ and $c(n)$ in $K[n]$ such that*
(i) b and c are monic polynomials,
(ii) $r(n) = \frac{a(n)}{b(n)}\frac{c(n+1)}{c(n)}$,
(iii) $gcd(a(n),b(n+h))=1$ for every non-negative integer h,
(iv) $gcd(a(n),c(n))=1$,
(v) $gcd(b(n),c(n+1))=1$.
Moreover, such polynomials are constructed by "Step2" of Gosper's Algorithm.

Proof. 1. Let $r(n)$ be a rational function. Thus,

$$r(n) = \frac{f(n)}{g(n)}$$

where $f(n)$ and $g(n)$ are polynomials such that $gcd(f(n), g(n)) = 1$. We want to find $a(n)$, $b(n)$ and $c(n)$ such that the conditions (i)-(v) holds. We have two cases to consider.

2. Case1: $gcd(f(n), g(n+h)) = 1$ for all nonnegative integers h. Then, obviously, we can take $a(n) = f(n)$, $b(n) = g(n)$ and $c(n) = 1$. However, b should be a monic polynomial. Thus we just divide both $a(n)$ and $b(n)$ by the leading coefficient of $b(n)$. Hence, we did find the desired factorization.

3. Case2: There exists a nonnegative integer $\tilde{h}$ such that $gcd(f(n), g(n+\tilde{h})) = u(n)$ where $u(n)$ is a non-constant polynomial. Then, we can take this $u(n)$ factor out from $f(n)$ and $g(n)$. More precisely,

$$\tilde{f}(n) = \frac{f(n)}{u(n)},$$

and

$$\tilde{g}(n) = \frac{g(n)}{u(n-h)}.$$

Then,

$$r(n) = \frac{f(n)}{g(n)} = \frac{\tilde{f}(n)}{\tilde{g}(n)} \frac{u(n)}{u(n-h)}.$$

We can write the ratio $\frac{u(n)}{u(n-h)}$ as follows:

$$\frac{u(n)}{u(n-h)} = \frac{u(n)u(n-1)...u(n-h+1)}{u(n-1)u(n-2)...u(n-h)}.$$

Why do we write it like this? If we look at all the factors seperately, we can see that they are all of the form $\frac{c(n+1)}{c(n)}$. Now, we will repeat the same procedure for $\tilde{f}(n)$ and $\tilde{g}(n)$, instead of $f(n)$ and $g(n)$, respectively. After finite number of steps we are done, since all polynomials have finite degree. Thus, they have finitely many factors as well.

□

Remark 23. *More precise explanation of the algorithm used in the proof is given in Algorithm1.*

Example 21. *Let's try to follow the procedure in an example. Find the canonical form of* $\frac{(n+1)^2(n^5+6n^4+4n^3+5n+1)}{n(n+3)^2}$.

1. *Let* $f(n) = (n+1)^2(n^5+6n^4+4n^3+5n+1)$ *and* $g(n) = n(n+3)^2$. *Then, clearly,* $gcd(f(n), g(n+1)) = (n+1)$. *Thus, let* $\tilde{f}(n) = \frac{(n+1)^2(n^5+6n^4+4n^3+5n+1)}{n+1}$

and $\tilde{g}(n) = \frac{n(n+3)^2}{n}$. *Hence,*

$$\tilde{f}(n) = (n+1)(n^5+6n^4+4n^3+5n+1),$$

and

$$\tilde{g}(n) = (n+3)^2.$$

2. *Now we repeat the same procedure for* $\tilde{f}(n)$ *and* $\tilde{g}(n)$. *Since* $gcd(\tilde{f}(n), \tilde{g}(n+h) = 1$ *for all nonnegative integers* h, *we are done. As a result we have*

$$\frac{(n+1)^2(n^5+6n^4+4n^3+5n+1)}{n(n+3)^2} = \frac{(n+1)(n^5+6n^4+4n^3+5n+1)}{(n+3)^2}\frac{n+1}{n}.$$

Thus choosing $a(n) = (n+1)(n^5+6n^4+4n^3+5n+1)$, $b(n) = (n+3)^2$ *and* $c(n) = n$ *gives us desired factorization.*

Also let's see what happens if we did not use the canonical form of the rational functions:

Example 22. *Evaluate* $\sum_{k=0}^{n-1}(k+1)(k+3)$

Solution:

1. As usual, let $t_n = (n+1)(n+2)$.

$$r(n) = \frac{t_{n+1}}{t_n} = \frac{(n+2)(n+4)}{(n+1)(n+3)}.$$

2. Now, normally, we need to write canonical form of $r(n)$. However, suppose we use $a(n) = (n+4)(n+2)$, $b(n) = (n+3)(n+1)$ and $c(n) = 1$ instead of the correct choices $a(n) = 1$, $b(n) = 1$ and $c(n) = (n+3)(n+1)$.

3. Then, we construct the equation

$$a(n)x(n+1) - b(n-1)x(n) = c(n) \rightarrow (n+4)(n+2)x(n+1) - n(n+2)x(n) = 1$$

Obviously, it does not have a polynomial solution. Thus, the sum is not Gosper summable!

4. Let's proceed with the correct choices of $a(n)$, $b(n)$ and $c(n)$:

$$a(n)x(n+1) - b(n-1)x(n) = c(n) \rightarrow x(n+1) - x(n) = (n+3)(n+1)$$

Then, $degx(n) = 3$. Substitute $x(n) = c_3n^3 + c^2n^2 + c_1n + c_0$ and solving the corresponding equation gives $x(n) = \frac{n^3}{3} + \frac{3n^2}{2} + \frac{7n}{6}$.

5. Finally, computing

$$\frac{b(n-1)x(n)}{c(n)} t_n = \frac{n^3}{3} + \frac{3n^2}{2} + \frac{7n}{6}.$$

   In other words, the sum is Gosper summable as it should be! Because the summand is a polynomial.

Moral of the Story: The canonical form of rational function is necessary for Gosper's algorithm to work correctly. Note that in the above example the wrong choices of $a(n)$ and $b(n)$ satisfy the condition $gcd(a(n), b(n)) = 1$. Thus, as shown in the example, this condition is not enough!

Remark 24. *Looking at the above examples, we can find the degree of $x(n)$ easily. Let's try to generalize and automate the process. We have two cases to consider:*

1. *Case1: The leading terms of equation* (4.1) *do not cancel each other. In other words, we have $deg(a(n)) \neq deg(b(n))$ or $lc(a(n)) \neq lc(b(n))$. Thus, the degree on the left hand-side is $d + max\{deg(a(n)) + deg(b(n))\}$. Equating this to degree of the right hand-side gives $d + max\{deg(a(n)) + deg(b(n))\} = deg(c(n))$. Thus, $d = deg(c(n)) - max\{deg(a(n)) + deg(b(n))\}$.*

2. *Case2: The leading terms of equation* (4.1) *cancel each other. Then, we have two cases to analyze:*

   (a) *Case2a : On the left hand-side the second-highest degree terms do not cancel each other. Then, on the left hand-side the degree is $d + max\{deg(a(n)) + deg(b(n))\} - 1$. Equating this to the right hand-side's degree, we get : $d + max\{deg(a(n)) + deg(b(n))\} - 1 = deg(c(n))$. As a result, $d = deg(c(n)) - max\{deg(a(n)) + deg(b(n))\} + 1$*

   (b) *Case2b : On the left hand-side, the second-highest degree terms cancel each other. Then, let*

$$\begin{aligned} a(n) &= Ln^e + An^{e-1} + O(n^{e-2}), \\ b(n-1) &= Ln^e + Bn^{e-1} + O(n^{e-2}), \\ x(n) &= Cn^d + Dn^{d-1} + O(n^{d-2}), \\ x(n+1) &= Cn^d + (Cd + D)n^{d-1} + O(n^{d-2}). \end{aligned}$$

Now we can substitute this into (4.1)*. Then, we have*

$$a(n)x(n+1) = LCn^{e+d} + (LCd + LD + AC)n^{e+d-1} + O(n^{e+d-2}),$$
$$b(n-1)x(n) = LC^{e+d} + (LD + CB)n^{e+d-1} + O(n^{e+d-2}),$$
$$a(n)x(n+1) - b(n-1)x(n) = C(Ld + A - B)n^{e+d-1} + O(n^{e+d-2}).$$

By our assumption, second-highest degree terms cancel out, i.e $C(Ld + A - B) = 0$*. Since,* $deg(x(n)) = d$*,* $C \neq 0$*. This means, that* $Ld + A = B$*. Thus,* $d = \frac{B-A}{L}$*. Since both case2a and case2b can occur, we can take maximum of this upper bound for* $deg(x(n))$*. In other words, in case2 ,*

$$deg(x(n)) = max\{deg(c(n)) - max\{deg(a(n)) + deg(b(n))\} + 1, \frac{B-A}{L}\}$$

Example 23. *Let's look at different examples where we see all of above cases.*

1. *Suppose* $a(n) = n^2$ *,* $b(n-1) = n+1$ *and* $c(n) = (n+2)^3$*. In this case, we are in case1, since* $dega(n) \neq degb(n)$*. Thus,* $degx(n) = degc(n) - max\{dega(n) + degb(n)\}$*. In other words,* $degx(n) = 3 - 2 = 1$*.*

2. *Suppose* $a(n) = n$ *,* $b(n-1) = n+3$ *and* $c(n) = 1$*. In this case, we are in case2. For case2a, we find* $d = 0$ *and in case2b, we find* $d = 3$*. It is ok and normal since both* $x(n) = \frac{-1}{3}$ *and* $x(n) = \alpha n^3 + 3\alpha n^2 + 2\alpha n - \frac{1}{3}$ *, for all* α *, are solutions to* (4.1)*.*

Now, we can look at the steps of the algorithm:

Steps of the Algorithm

(i) Let the summand be t_n and calculate the term ratio $r(n) = \frac{t_{n+1}}{t_n}$.
(ii) Write $r(n)$ in canonical form. In other words, find polynomials $a(n)$,$b(n)$ and $c(n)$ such that $r(n) = \frac{a(n)}{b(n)}\frac{c(n+1)}{c(n)}$.
(iii) Form $a(n)x(n+1) - b(n-1)x(n) = c(n)$. Using this equation find $d = deg(x(n))$. Substitute a generic polynomial of degree d in place of $x(n)$.
(iv) Construct a linear system via equating the coefficient of each power of n to 0.
(v) Try to solve the linear system. If there does not exist a solution, then the input sum cannot be written as a sum of hypergeometric term plus constant. If there exists a solution, then $\frac{b(n-1)x(n)}{c(n)}t_n$ is the evaluation of the sum.

Now, we will explain how to find canonical forms of rational functions algorithmically:

Algorithm 1: Finding the Canonical Form of Rational Functions

Input: A rational function $C\frac{f(n)}{g(n)}$, where f and g are monic polynomials and C is a constant.

Output: Polynomials $a(n)$, $b(n)$ and $c(n)$ such that conditions on Defition 10 holds.

Let $R(h) := Resultant_n(f(n), g(n+h))$;
Let $S := \{h_1, h_2, ..., h_m\}$ be the set of nonnegative integer zeros of $R(h)$
Let $p_0(n) := f(n)$; $q_0(n) := g(n)$;
for j=1,2..m do
  $s_j(n) := gcd(p_{j-1}(n), q_{j-1}(n+h_j))$;
  $p_j(n) := \frac{p_{j-1}(n)}{s_j}$;
  $q_j(n) := \frac{q_{j-1}(n)}{s_j(n-h_j)}$;
$a(n) := Cp_m(n)$;
$b(n) := q_m(n)$;
$c(n) := \prod_{i=1}^{m}\prod_{j=1}^{h_i} s_i(n-j)$;

Example 24. *Given* $d(n) = \frac{(n-3)(6-n)}{(n+7)(-n+2)(n-5)}$*, find* $a(n)$ *,* $b(n)$ *and* $c(n)$ *such that the conditions in Definition 10 holds.*

Solution:

1. First, let's clarify our input: We have $C = 1$, $f(n) = (n-3)(6-n)$ and $g(n) = (n+7)(-n+2)(n-5)$.

2. Second, we need to find the resultant of $f(n)$ and $g(n+h)$: Using maple, we find that $R(h) = (10+h)(-1-h)^2(-2+h)(-13-h)(4+h)$.

3. Third, we need to find nonnegative integer zeros of $R(h)$. Obviously, we have $S = \{2\}$.

4. Fourth, $p_0(n) = f(n)$, in other words, $p_0(n) = (n-3)(6-n)$ and $q_0(n) = g(n)$, i.e , $q_0(n) = (n+7)(-n+2)(n-5)$.

5. Since, S has 1 element, $m = 1$. In the first iteration of the loop $j = 1$. We have, $s_1(j) = gcd(p_0(n), q_0(n+2)) = gcd((n-3)(6-n), (n+9)(-n+4)(n-3) = (n-3)$. Similarly, $p_1(n) = (6-n)$ and $q_1(n) = (n+7)(n-2)$.

6. Finally, $a(n) = Cp_1(n) = (6-n)$, $b(n) = q_1(n) = (n+7)(n-2)$ and $c(n) := \prod_{i=1}^{1}\prod_{j=1}^{h_i} s_i(n-j) = (n-4)(n-5)$.

As a result, we can write $\frac{(n-3)(6-n)}{(n+7)(-n+2)(n-5)} = \frac{(6-n)}{(n+7)(n-2)}\frac{(n-3)(n-4)}{(n-4)(n-5)}$.

Let's look at the Gosper's algorithm from a broader perspective: Given a sum we want to understand whether this sum can be written as a hypergeometric term plus a constant. However, it may be the case that the sum can be written as sum of two hypergeometric terms or sum of three hypergeometric terms etc. Thus, a more general question is the following: Given a sum of the form $\sum_{k=0}^{n} t_k$ can we find hypergeometric functions $s_1(n), s_2(n), ..s_r(n)$, r is a fixed integer, such that this sum can be written as a linear combination of $s_i(n)$'s? The following theorem answers our question:

Theorem 5. *If Gosper's algorithm does not succeed then the given sum cannot be expressed as a linear combination of a fixed number of hypergeometric terms.*

Let's try to understand why Theorem 5 holds:

1. Suppose we have a sum $\sum_{k=0}^{n-1} t_k$. Also, assume that

$$\sum_{k=0}^{n-1} t_k = h_n^{(1)} + h_n^{(2)} + .. + h_n^{(r)}$$

   Then, obviously we have

$$t_n = (h_{n+1}^{(1)} - h_n^{(1)}) + (h_{n+1}^{(2)} - h_n^{(2)}) + ... + (h_{n+1}^{(r)} - h_n^{(r)}) \tag{4.5}$$

   We claim that:

   (a) First, we can assume that $h_n^{(1)}, h_n^{(2)}, ...h_n^{(r)}$ are pairwise dissimilar.

   (b) Second, $r \leq 2$.

   (c) Third, if $r = 2$, then $h_n^{(1)}$ or $h_n^{(2)}$ is a constant.

2. Proof of (a): Suppose we find hypergeometric terms $j_n^{(1)}, j_n^{(2)}, ...j_n^{(m)}$ such that $\sum_{k=0}^{n-1} t_k = j_n^{(1)} + j_n^{(2)} + .. + j_n^{(m)}$ and $j_n^{(d)}$ is similar with $j_n^{(e)}$ for some $1 \leq d \leq e \leq m$. Then, we can write $s_n = j_n^{(d)} + j_n^{(e)}$. By Definition 8 s_n is a hypergeometric term. We can apply the same procedure until we get pairwise dissimilar terms. Thus, we can assume that $h_n^{(1)}$, $h_n^{(2)}$,...$h_n^{(r)}$ are pairwise dissimilar.

3. Proof of (b) and (c): Obviously, on the right hand side of (4.5), the terms are pairwise dissimilar. Then, their sum cannot be a hypergeometric function unless

   (a) $r = 1$ and $t_n = h_{n+1}^{(i)} - h_n^{(i)}$ for some i or

   (b) $r = 2$ and $t_n = h_{n+1}^{(j)} - h_n^{(j)} + CONST$ for some j and $CONST = h_{n+1}^{(k)} - h_n^{(k)}$ for some k.

Note that in both cases, Gosper's algorithm will give a positive answer. In other words, Theorem 5 holds.

Moral of the Story: Consider $\sum_{k=0}^{n-1} t_k$. If Gosper's algorithm proves that we cannot find a hypergeometric function $d(n)$ such that $d_{n+1} - d_n = t_n$, then our sum cannot be written as a linear combination of fixed number of hypergeometric functions!

Now we will go back to the indefinite summation problem. However, this time we will use creative telescoping algorithm rather than Sister Celine's algorithm.

5. Creative Telescoping Algorithm

In this section, Creative Telescoping algorithm is discussed. Basically, this algorithm does the exact same job as Sister Celine's Algorithm but in a great faster fashion. As usual, we start by looking at examples rather than the theory.

Let's try to evaluate the sum $\sum_k \binom{n}{k}$ (again). So that, we can compare Sister Celine's Algorithm and Creative Telescoping algorithms. Also, in Example 13 we claim that Sister Celine's algorithm is one of the longest way to evaluate $\sum_k \binom{n}{k}$. However, we will see that using creative telescoping algorithm on this question gives us an even harder way!

Example 25. *Let's try to evaluate* $\sum_k \binom{n}{k}$.

Solution:

1. Let $F(n,k) = \binom{n}{k}$. Define

   $$t_k = a_0(n)F(n,k) + a_1(n)F(n+1,k). \tag{5.1}$$

   In other words, we assume that our sum satisfy a recurrence of order 1. Note that, in Sister Celine's algorithm we assume that $F(n,k)$ satisfies a recurrence of the the form $a(n)F(n,k)+b(n)F(n+1,k)+c(n)F(n,k+1)+d(n)F(n+1,k+1) = 0$. Thus, in Zeilberger's algorithm we only use shift in n in our recurrence.

2. Let's compute the term ratios

   $$\frac{F(n,k+1)}{F(n,k)} = \frac{n-k}{k+1} = \frac{r_1(n,k)}{r_2(n,k)},$$
   $$\frac{F(n,k)}{F(n-1,k)} = \frac{n}{n-k} = \frac{s_1(n,k)}{s_2(n,k)}.$$

   Thus, we have $r_1(n,k) = n-k$, $r_2(n,k) = k+1$, $s_1(n,k) = n$ and $s_2(n,k) = n-k$.

3. Let's compute $p_0(k)$, $r(k)$ and $s(k)$:

$$p_0(k) = a_0(n)s_2(n+1,k) + a_1(n)s_1(n+1,k) = a_0(n)(n+1-k) + a_1(n)(n+1),$$
$$r(k) = r_1(n,k)s_2(n+1,k) = (n-k)(n+1-k),$$
$$s(k) = r_2(n,k)s_2(n+1,k+1) = (k+1)(n-k).$$

4. Now, we will write $\frac{r(k)}{s(k)}$ in the canonical form.

$$\frac{r(k)}{s(k)} = \frac{p_1(k+1)}{p_1(k)}\frac{p_2(k)}{p_3(k)} = \frac{n+1-k}{k+1}.$$

   In other words, $p_1(k) = 1$, $p_2(k) = n+1-k$ and $p_3(k) = k+1$. Here, we are thinking in the following way: For each fix n we have a rational function $\frac{r(k)}{s(k)}$. Thus, from Gosper's algorithm we know that such a canonical form exists!

5. Let's compute $p(k) = p_0(k)p_1(k)$:

$$p(k) = p_0(k)p_1(k) = a_0(n)(n+1-k) + a_1(n)(n+1)$$

6. Now, we try to solve $p_2(k)b(k+1) - p_3(k-1)b(k) = p(k)$ as follows:

$$p_2(k)b(k+1) - p_3(k-1)b(k) = p(k) \rightarrow$$
$$(n+1-k)b(k+1) - kb(k) = a_0(n)(n+1-k) + a_1(n)(n+1)$$

7. Now, we will use Remark 24 to find $deg(b(k))$. We are in Case1. This means that $deg(b(k)) = 0$. Substituting $b(k) = c$ gives us

$$(n+1-k)(c) - k(c) = a_0(n)(n+1-k) + a_1(n)(n+1).$$

   Then we will match coefficients of each power of k in both sides. In other words,

$$k^1(-2c) = k^1(-a_0(n))$$
$$k^0(nc+c) = k^0(a_0(n)n + a_0(n) + a_1(n)n + a_1(n)$$

   This means that $a_0(n) = 2c$ and $a_1(n) = -c$.

8. Substituting $a_0(n) = 2c$ and $a_1(n) = -c$ gives us

$$2cF(n,k) - cF(n+1,k) = 0$$

Summing over k gives us,

$$2f(n) = f(n)$$

Combining $f(0) = 1$ gives us $f(n) = 2^n$.

As usual we will ask some questions about the application of the algorithm and state some possible problems:

1. We assume that our sum satisfies a recurrence of order 1. Suppose it is not true, then what should we do?

2. Which assumptions on the summand are needed so that the algorithm will work?

3. We use the canonical form of rational functions. Then, we get the same equation as in Gosper's algorithm. Is there a relationship between Gosper's algorithm and Zeilberger's algorithm?

4. It seems that Sister Celine's algorithm is much easier to use than Zeilberger's algorithm. Is this correct?

5. Suppose our algorithm inform us that a particular sum does not satisfy a recurrence of, say, order d. Can we conclude that we cannot find a recurrence of order d for our sum?

Before answering the questions, let's look at another example.

Example 26. *Let's try to evaluate* $\sum\limits_k \binom{n}{k}^3$

Solution:

1. First, we try to find a recurrence of order 1, and let $F(n,k) = \binom{n}{k}^3$. We need to find the term ratios:

$$\frac{r_1(n,k)}{r_2(n,k)} = \frac{F(n,k+1)}{F(n,k)} = \frac{\binom{n}{k+1}^3}{\binom{n}{k}^3} = \frac{(n-k)^3}{(k+1)^3},$$

$$\frac{s_1(n,k)}{s_2(n,k)} = \frac{F(n,k)}{F(n-1,k)} = \frac{\binom{n}{k}^3}{\binom{n-1}{k}^3} = \frac{n^3}{(n-k)^3}.$$

In other words, $r_1(n,k) = (n-k)^3$, $r_2(n,k) = (k+1)^3$, $s_1(n,k) = n^3$ and

$s_2(n,k) = (n-k)^3.$

2. Second, let's compute $p_0(k)$, $r(k)$ and $s(k)$:

$$\begin{aligned}
p_0(k) &= \sum_{j=0}^{J} a_j(n)(\prod_{i=0}^{j-1} s_1(n+j-i,k) \prod_{r=j+1}^{J} s_2(n+r,k)) \\
&= a_0(n)(s_2(n+1,k)) + a_1(n)(s_1(n+1,k)) \\
&= a_0(n)(n+1-k)^3 + a_1(n)(n+1)^3, \\
r(k) &= r_1(n,k) \prod_{r=1}^{J} s_2(n+r,k) \\
&= (n-k)^3 \cdot (n+1-k)^3, \\
s(k) &= r_2(n,k) \prod_{r=1}^{J} s_2(n+r,k+1) \\
&= (k+1)^3 \cdot (n-k)^3.
\end{aligned}$$

3. Third, we want to write $\frac{r(k)}{s(k)}$ in canonical form. In other words, we want to find $p_1(k)$, $p_2(k)$ and $p_3(k)$ such that $\frac{r(k)}{s(k)} = \frac{p_1(k+1)}{p_1(k)} \frac{p_2(k)}{p_3(k)}$. Obviously, we can choose $p_1(k) = 1$, $p_2(k) = (n+1-k)^3$ and $p_3(k) = (k+1)^3$.

4. Fourthly, let $p(k) = p_0(k)p_1(k)$. Then, $p(k) = a_0(n)(n+1-k)^3 + a_1(n)(n+1)^3$. We are ready to compute $p(k)$ as well.

$$\begin{aligned}
p_2(k)b(k+1) - p_3(k-1)b(k) = p(k) \rightarrow \\
(n+1-k)^3 b(k+1) - (k+1)^3 b(k) = a_0(n)(n+1-k)^3 + a_1(n)(n+1)^3.
\end{aligned}$$

5. Now, we need to find a degree bound for $b(k)$. Obviously, $deg(b(k)) = 0$, i.e $b(k)$ is a constant. Say, $b(k) = c$. Then, our equation becomes

$$(n+1-k)^3 c - (k+1)^3 c = a_0(n)(n+1-k)^3 + a_1(n)(n+1)^3.$$

6. Finally, we match coefficient of like powers of k in both sides.

$$\begin{aligned}
k^3(-2c) &= k^3(-a_0(n)) \\
k^2(3cn - 3cn^2) &= k^2(3a_0(n)n + 3a_0(n)) \\
k^1(-3cn^2 - 6cn - 6c) &= k^1(-3a_0(n)n^2 - 6a_0(n)n - 3a_0(n)) \\
k^0(cn^3 + 3cn^2 + 3cn) &= k^0(a_0(n)n^3 + 3a_0(n)n^2 + 3a_0(n)n + \\
& a_0(n) + a_1(n)n^3 + 3a_1(n)n^2 + 3a_1(n)n + a_1(n)).
\end{aligned}$$

Then, the only solution is $c=0$, $a_0(n)=0$ and $a_1(n)=0$. This means that no nontrivial solution exists for this linear system. In other words, Creative Telescoping algorithm cannot find a recurrence of order 1 satisfied by $f(n)=\sum\limits_k \binom{n}{k}^3$.

7. Now, we try to find a recurrence of order 2. To be more precise, we assume that

$$t_k = a_0(n)F(n,k)+a_1(n)F(n+1,k)+a_2F(n+2,k).$$

Since the computations would be extremely messy, let's use Maple to do the calculations for us. We will denote the order of the recurrence by J. Thus, let $J=1$ and run the creative telescoping algorithm: $ct(\binom{n}{k}^3,1,k,n,N)$ gives us 0. It means that, Zeilberger's algorithm cannot find a recurrence of order 1 as we found above. Let's try it with $J=2$. $ct(\binom{n}{k}^3,2,k,n,N)$ gives us

$$-8(n+1)^2+(-7n^2-21n-16)N+(n+2)^2N^2. \tag{5.2}$$

It means that (5.2) anihilates $f(n)=\sum_k \binom{n}{k}^3$ where N is the shift operator with respect to n. In other words, we have the following equation:

$$(n+2)^2f(n+2)-(7n^2-21n-16)f(n+1)-8(n+1)^2f(n)=0.$$

Since, $f(0)=1$ and $f(1)=2$, in principle we can solve this recurrence and find $f(n)$. Unfortunately, it is not that simple as Theorem 7 shows. We will talk about this issue in more detail in Chapter 6.

Now, we can answer the questions:

1. It is clear from the above example that, we should look for a recurrence of order 2. In general, if we cannot find a recurrence of order J for some J, then we should look for a recurrence of order $J+1$.

2. If $F(n,k)$ is a proper hypergeometric term, then Zeilberger's algorithm surely terminates. See Theorem 6 for more explanation.

3. See Remark 26.

4. It is a good time to talk about more general question: Suppose we can solve a particular problem using three different methods. Which one is better and why? Obviously, there is no good answer to this question, since its answer depends on the problem, methods, even one's preferences. However, in our case, we are comparing two algorithms we have an unbiased measure of goodness: efficiency of the algorithm. From computer science perspective, Zeilberger's

algorithm is (much) more efficient (faster) than Sister Celine's algorithm in terms of time complexity. We will not discuss the exact time complexities of these algorithms.

General Moral of the Story: Maybe an algorithm, say algorithm1, seems much harder to apply then another algorithm, say algorithm2, by hand; still it may be the case that algorithm1 would be preferable to algorithm2 from computer science perspective.

Specific Moral of the Story: If you have a computer, then use Zeilberger's algorithm instead of Sister Celine's algorithm!

5. Unfortunately, this is not true. See Example 27.

Remark 25. *As we will see in Theorem 6, if $F(n,k)$ is a proper hypergeometric term, then Zeilberger's algorithm finds a recurrence satisfied by $f(n) = \sum_k F(n,k)$. However, it is not a necessary condition, it is a sufficient condition. It would be better to have something of the form "Zeilberger's algorithm is applicable for $f(n) = \sum_k F(n,k)$ if and only if $F(n,k)$ satisfies something." Such a statement is given by Abrahamov at Abramov (2003). Moreover, there is a Maple function IsZApplicable which checks whether we can use Zeilberger's algorithm on $F(n,k)$ or not.*

Remark 26. *Let us try to motivate the steps we are following in Zeilberger's Algorithm. Interestingly enough, actually we are just applying Gosper's algorithm in a non-obvious way: Suppose we want to evaluate $\sum_k F(n,k)$. We always assume that our summand, $F(n,k)$, has a compact support. Then, we can see our sum as $\sum_{k=b}^{c} F(n,k)$ where b and c are natural bounds of our summand. This means that $\sum_{k=a}^{d} F(n,k) = \sum_{k=b}^{c} F(n,k)$ if $a < b$ and $d > c$. Also, $F(n,b) \neq 0$ and $F(n,c) \neq 0$. Now, we have a finite sum. Thus, maybe, we can apply Gosper's algorithm. Direct application of Gosper's algorithm does not give us anything since most of the functions we encounter are not Gosper summable. However, an undirect application gives us what we want: We will use $t_k = \sum_{j=0}^{J} a_j(n)F(n+j,k)$ as our summand and apply Gosper's algorithm!*

1. *Let $t_k = \sum_{j=0}^{J} a_j(n)F(n+j,k)$. Then, we have*

$$\frac{t_{k+1}}{t_k} = \frac{\sum_{j=0}^{J} a_j(n)F(n+j,k+1)}{\sum_{j=0}^{J} a_j(n)F(n+j,k)}. \tag{5.3}$$

We can write (5.3) *as*

$$\frac{t_{k+1}}{t_k} = \frac{F(n,k+1)}{F(n,k)} \frac{1+\sum_{j=1}^{J} a_j(n)\frac{F(n+j,k+1)}{F(n,k+1)}}{1+\sum_{j=1}^{J} a_j(n)\frac{F(n+j,k)}{F(n,k)}}.$$

Obviously, we have a rational function of k. *Thus, we can find the canonical form of the rational function* $\frac{t_{k+1}}{t_k}$ *as* $\frac{t_{k+1}}{t_k} = \frac{p_0(k+1)}{p_0(k)}\frac{r(k)}{s(k)}$. *Then, we find the canonical form of the rational function* $\frac{r(k)}{s(k)}$ *as* $\frac{r(k)}{s(k)} = \frac{p_1(k+1)}{p_1(k)}\frac{p_2(k)}{p_3(k)}$. *Combining these two canonical forms gives us* $\frac{t_{k+1}}{t_k} = \frac{p_{k+1}}{p_k}\frac{p_2(k)}{p_3(k)}$. *Now we are in the step3 of the Gosper's algorithm. Thus, we are looking for solutions of* $p_2(k)b(k+1) - p_3(k-1)b(k) = p(k)$. *Using step3-4-5 of Gosper's algorithm, we are done. This corresponds to steps 5-6-7 of Zeilberger's algorithm. See below for the details of steps of the algorithm.*

The following example shows that, Zeilberger's algorithm may not find the smallest order recurrence.

Example 27. *Evaluate* $f(n) = \sum_k (-1)^k \binom{n}{k}\binom{3k}{n}$.

Solution:

1. First, we try to find a recurrence of order 1, using the following Maple code: $ct((-1)^k \binom{n}{k}\binom{3k}{n}, 1, k, n, N)$ this gives us 0. It means that, the Zeilberger's algorithm cannot find an order 1 recurrence satisfied by $f(n)$.

2. Second, we look for a recurrence of order 2, using the following Maple code: $ct((-1)^k \binom{n}{k}\binom{3k}{n}, 2, k, n, N)$, this gives us $-9(n+2)(n+1)f(n) - 3(n+2)(5n+7)f(n+1) - 2(n+2)(2n+3)f(n+2) = 0$.

3. If we solve this recurrence using algorithm hyper, we get: $f(n) = (-3)^n$. Thus, $f(n)$ satisfies $f(n+1)+3f(n) = 0$. In other words, $f(n)$ satisfies an order 1 recurrence!

Moral of the Story: Zeilberger's algorithm may or may not give a recurrence of smallest order. It means that we cannot use it to prove that a particular sum $\sum_k F(n,k)$ does not satisfy a recurrence of order, say, 2.

Our next example is important to understand Wilf-Zeilberger phenomenon, see Wilf & Zeilberger (1992).

Example 28. *Let's try solve an AMM problem from 2008, see Beckwith (2008)*

Prove that $\sum_{k\geq 0}\binom{n}{k}\binom{2k}{k} = \sum_{k\geq 0}\binom{n}{2k}\binom{2k}{k}3^{n-2k}$.

Solution:

1. *First, let* $f(n) = \sum_k \binom{n}{k}\binom{2k}{k}$*, let* $g(n) = \sum_k \binom{n}{2k}\binom{2k}{k}3^{n-2k}$*. Now, let's give the left hand side to Zeilberger's algorithm to find a recurrence satisfied by it.*

2. *ct(*$\binom{n}{k}\binom{2k}{k}$*,*$1,k,n,N$*) gives us 0. It means that, Zeilberger's algorithm cannot find a recurrence of order* 1 *satisfied by* $f(n)$*. Let's try to find a recurrence of order* 2*. ct(*$\binom{n}{k}\binom{2k}{k}$*,*$2,k,n,N$*) gives* $5n+5+(-6n-9)N+(n+2)N^2$*. It means that* $(5n+5)f(n)+(-6n-9)f(n+1)+(n+2)f(n+2)=0$*.*

3. *ct(*$\binom{n}{2k}\binom{2k}{k}3^{n-2k}$*,*$1,k,n,N$*) gives us 0. It means that, Zeilberger's algorithm cannot find a recurrence of order* 1 *satisfied by* $g(n)$*. Let's try to find a recurrence of order* 2*. ct(*$\binom{n}{2k}\binom{2k}{k}3^{n-2k}$*,*$2,k,n,N$*) gives* $5n+5+(-6n-9)N+(n+2)N^2$*. Thus,* $(5n+5)g(n)+(-6n-9)g(n+1)+(n+2)g(n+2)=0$*. In other words, they both satisfies the same recurrence of order* 2*!*

4. $f(0)=1$ *,* $f(1)=3$ *and* $g(0)=1$ *,* $g(1)=3$*. Thus, they agree on two initial values, we are done!*

Remark 27. *Let's look at our question from broader perspective:*
Question: Prove that $\sum_k NICE_1(n,k) = \sum_k NICE_2(n,k)$*.*
Using ***Wilf-Zeilberger phenomenon*** *we can proceed as follows:*

1. *First, apply Zeilberger's algorithm to the left hand side to find a recurrence satisfied by it.*

2. *Second, apply Zeilberger's algorithm to the right hand side to find a recurrence satisfied by it. Since, the left hand side equals to right hand side, we hope to find the same recurrence.*

3. *Lastly, check that both sides agreed on enough initial values.*

Let's solve a famous example using the computer.

Example 29. *Show that* $A(n)$ *satisfies the following recurrence*
$(n+2)^3A_{n+2} - (2n+3)(17n^2+51n+39)A_{n+1} + (n+1)^3A_n = 0$ *where* $A_n = \sum_{k=0}^{n}\binom{n}{k}^2\binom{n+k}{k}^2$*. This played a crucial role in Apery's proof of irrationality of* $\zeta(3)$ *, see Apéry (1979).*

Solution:

1. *As usual let* $F(n,k) = \binom{n}{k}^2\binom{n+k}{k}^2$*. Assume that this sum satisfies a recurrence of order* 1*. We want to find this recurrence, i.e we want to determine coefficient* $a_0(n)$ *and* $a_1(n)$ *in equation* (5.1)*. Let left hand side of the equation* (5.1) *be*

t_k. *Thus, we have*

$$t_k = a_0(n)F(n,k) + a_1(n)F(n+1,k).$$

2. *Now we will find the term ratio,* $\frac{t_{k+1}}{t_k}$ *as*

$$\frac{t_{k+1}}{t_k} = \frac{a_0(n)F(n,k+1) + a_1(n)F(n+1,k+1)}{a_0(n)F(n,k) + a_1(n)F(n+1,k)}$$
$$= \frac{(k-n-1)^2((n+k+2)^2a_1(n) + a_0(n)(k-n)^2)(n+k+1)^2}{((n+k+1)^2a_1(n) + a_0(n)(k-n-1)^2)(k+1)^4}.$$

3. *Let's compute the term ratios* $\frac{r_1(n,k)}{r_2(n,k)} = \frac{F(n,k+1)}{F(n,k)}$ *and* $\frac{s_1(n,k)}{s_2(n,k)} = \frac{F(n,k)}{F(n-1,k)}$*:*

$$\frac{r_1(n,k}{r_2(n,k)} = \frac{(k-n)^2(n+k+1)^2}{(k+1)^4},$$
$$\frac{s_1(n,k)}{s_2(n,k)} = \frac{(n+k)^2}{(n-k)^2}.$$

4. *Let's compute* $p_0(k)$ *,* $r(k)$ *and* $s(k)$ *as follows:*

$$p_0(k) = a_0(n)s_2(n+1,k) = a_0(n)(n+1-k)^2,$$
$$r(k) = r_1(n,k)s_2(n+1,k) = (k-n)^2(n+k+1)^2(n+1-k)^2,$$
$$s(k) = r_2(n,k)s_2(n+1,k+1) = (k+1)^4(n-k)^2.$$

5. *Now, we will write* $\frac{r(k)}{s(k)}$ *in the canonical form:*

$$\frac{r(k)}{s(k)} = \frac{(n+k+1)^2(k-n-1)^2}{(k+1)^4}\frac{1}{1}.$$

6. *Try to find solution for* $p_2(k)b(k+1) - p_3(k-1)b(k) = p(k)$*. In other words,*

$$(n+k+1)^2(k-n-1)^2b(k+1) - k^4b(k) = 1$$

Then, we must have $deg(b(k)) = 0$ *if there is a polyonmial solution. However, writing* $b(k) = c$ *does not give a solution. Now, need to update our assumption as*

$$t_k = a_0(n)F(n,k) + a_1(n)F(n+1,k) + a_2(n)F(n+2,k).$$

7. *We start all over again... In other words, we need to look for a recurrence of order* 2. *Since the computations would be extremely messy, we will not do it by hand. Instead the following Maple code finishes the job :* $ct(binomial(n,k)^2 * binomial(n+k,k)^2, 1, k, n, N)$ *this gives an output 0 meaning that Zeilberger's algorithm cannot find a order* 1 *recurrence satisfied by* $f(n)$. $ct(binomial(n,k)^2 * binomial(n+k,k)^2, 2, k, n, N)$ *gives an output* $(n+1)^3 - (17n^2+51n+39)(2n+3)N + (n+2)^3N^2$. *Thus, we find a second order recurrence satisfied by* $f(n)$*!*

The following theorem guarantees the success of Zeilberger's algorithm, under the assumption that $F(n,k)$ is a proper hypergeometric function.

Theorem 6. *If* $F(n,k)$ *is a proper hypergeometric function, then there exists a nonnegative integer* d*, a rational function* $R(n,k)$ *and polynomials* $\{a_j(n)\}_{j=0}^{d}$*, such that* $F(n,k)$ *satisfies*

$$\sum_{j=0}^{d} a_j(n)F(n+j,k) = G(n,k+1) - G(n,k),$$

where $G(n,k) = R(n,k)F(n,k)$.

Proof. 1. We know that by Theorem 2 there exist I and J such that

$$\sum_{i=0}^{I}\sum_{j=0}^{J} a_{i,j}(n)F(n+i,k+j) = 0. \tag{5.4}$$

2. Obviously we can write (5.4) as

$$P(N,n,K)F(n,k) = 0,$$

where $P(N,n,K)$ is a linear recurrence operator of two-variables. Observe that, if we take any polynomial $S(u,v,w)$ and expand it as a power series in w about the point $w = 1$, we get

$$S(u,v,w) = S(u,v,1) + (1-w)Q(u,v,w). \tag{5.5}$$

where $Q(u,v,w)$ is a polynomial. Then, taking $P(N,n,K)$ insted of $S(u,v,w)$ in the (5.5), we get:

$$0 = P(N,n,K)F(n,k) = ((P(N,n,1) + (1-K)Q(N,n,K))F(n,k). \tag{5.6}$$

Hence, we have

$$P(N,n,1) = (K-1)Q(N,n,K)F(n,k). \tag{5.7}$$

If we let $G(n,k) := Q(N,n,K)F(n,k)$, then on the right hand side of (5.7) we simply have $G(n,k+1) - G(n,k)$. Also note that $G(n,k)$ is just a rational function multiple of $F(n,k)$ since $G(n,k)$ is obtained by applying a shift operator $Q(N,n,K)$ to $F(n,k)$ which is same as multiplying $F(n,k)$ by a rational function.

3. Now, we need to show that the recurrence is non-trivial. By Theorem 2 we know that there exists non-trivial operators $P(N,n,K)$ such that $P(N,n,K)$ annihilates $F(n,k)$, i.e $P(N,n,K)F(n,k) = 0$. Let P be the one that has the least degree in K. Then, we can divide P by $K-1$ to get

$$P(N,n,K) = P(N,n,1) - (K-1)Q(N,n,K) \tag{5.8}$$

Note that this equation completely specifies Q. Intuitively speaking, in equation (5.8), we decomposed $P(N,n,K)$ as an operator without K plus a operator with K.

4. Suppose $P(N,n,1) = 0$. In other words, all the terms of $P(N,n,K)$ contains K. Then, $(K-1)G(n,k) = 0$. This means that, $G(n,k)$ is independent of k. Thus, $G(n,k)$ is a hypergeometric term in the single variable. Recall that a function is hypergeometric if and only if there is a first-order operator $H(N,n)$ such that $H(N,n)G(n,k) = 0$. Thus, such a first-order $H(N,n)$ exists.

5. If $Q(N,n,K) = 0$, then $P(N,n,K) = P(N,n,1)$. In other words, P is a k-free operator.

6. If $Q(N,n,K) \neq 0$,then $H(N,n)Q(N,n,K)$ is a nonzero k-free operator annihilating $F(n,k)$.

7. In either case, we have found a nonzero k-free operator that annihilates $F(n,k)$ and whose degree in K is smaller than that of $P(N,n,K)$, which contradicts the fact that P is the one that has the least degree in K.

□

Steps of the Algorithm

1. Given a summand $F(n,k)$, let $J = 1$. Compute $t_k = \sum_{j=0}^{J} a_j(n)F(n+j,k)$.
2. Find term ratios $\frac{F(n,k+1)}{F(n,k)} = \frac{r_1(n,k)}{r_2(n,k)}$ and $\frac{F(n,k)}{F(n-1,k)} = \frac{s_1(n,k)}{s_2(n,k)}$.

3. Calculate $p_0(k)=\sum_{j=0}^{J}a_j(n)\{\prod_{i=0}^{j-1}s_1(n+j-i,k)\prod_{r=j+1}^{J}s_2(n+r,k)\}$, $r(k)=r_1(n,k)\prod_{r=1}^{J}s_2(n+r,k)$ and $s(k)=r_2(n,k)\prod_{r=1}^{J}s_2(n+r,k+1)$.

4. Find the canonical form of $\frac{r(k)}{s(k)}$, i.e find $p_1(k)$, $p_2(k)$ and $p_3(k)$ such that $\frac{r(k)}{s(k)}=\frac{p_1(k+1)}{p_1(k)}\frac{p_2(k)}{p_3(k)}$. Compute $p(k)=p_0(k)p_1(k)$.

5. Construct $p_2(k)b(k+1)-p_3(k-1)b(k)=p(k)$. Using Remark 24, determine the degree of $b(k)$, say $deg(b(k))=d$.

6. Substitute a generic polynomial of degree d in place of $b(k)$.

7. Try to match the coefficient of like powers of k in the both sides. If this is possible, then we find the recurrence. Otherwise, increase J by 1 and start from scratch.

Remark 28. *Now it is time to explain a big advantage of our methods: We do not need to believe what computer gives us, we can easily check the result as well. In other words, there is something called* ***proof certificate*** *which is an easy method to check whether our algorithm is true or not. Let's look at different examples:*

1. *Suppose we want to compute* $\sum\limits_{k=0}^{n-1}(k^4+3k^2+k+4)$. *Then, we can use Gosper's algorithm to evaluate it. Gosper's algorithm gives us* $\frac{n^5}{5}-\frac{n^4}{2}+\frac{4n^3}{3}-n^2+\frac{119n}{30}$. *How do we check this answer? Recall that, Gosper's algorithm finds a hypergeometric function* $d(n)$ *such that* $d(n+1)-d(n)=t_n$. *If we are given such a* $d(n)$, *then it is extremely easy to check the computation. In this case* $d(n)$ *is our proof certificate.*

2. *Similarly, suppose we want to find a recurrence satisfied by* $f(n)=\sum\limits_{k}\binom{n}{k}\binom{2k}{k}$. *Then, Zeilberger's algorithm gives us*

$$(5n+5)f(n)+(-6n-9)f(n+1)+(n+2)f(n+2)=0.$$

This time we can use

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

as our proof certificate. It shows that,

$$(5n+5)F(n,k)+(-6n-9)F(n+1,k)+(n+2)F(n+2,k)=$$
$$F(n,k+1)R(n,k+1)-F(n,k)R(n,k).$$

Again, it is an routinely verifiable identity. In both cases just one rational *function is enough to show the correctness of our computation!*

Remark 29. *It can be the case that, somehow we conjecture an identity of the form* $\sum_k F(n,k) = a(n)$. *We want to check whether our claim is correct or not. We can proceed as follows:*

1. *First we divide the both sides of our equation by* $a(n)$ *to get:*

$$\sum_k \frac{F(n,k)}{a(n)} = 1.$$

   Let $T(n,k) := \frac{F(n,k)}{a(n)}$. *Then, it is enough to prove that*

$$f(n) = \sum_k T(n,k) = 1.$$

2. *From a broader perspective we try to prove that* $f(n)$ *is a constant. Thus, it suffices to show* $f(n+1) - f(n) = 0$ *for all* n.

3. *Suppose there is a function* $G(n,k)$ *such that* $T(n+1,k) - T(n,k) = G(n,k+1) - G(n,k)$ *and* $\sum_k G(n,k+1) - G(n,k) = 0$. *Then, we are done.* $T(n,k)$ *and* $G(n,k)$ *is called a* ***WZ pair***.

Example 30. *Prove that* $\sum_k k^2\binom{n}{k} = 2^{n-2}n(n+1)$.

Solution:

1. *Let* $F(n,k) = k^2\binom{n}{k}$ *and* $a(n) = 2^{n-2}n(n+1)$. *Then,* $T(n,k) = \frac{k^2\binom{n}{k}}{2^{n-2}n(n+1)}$. *Using Zeilberger's algorithm on* $T(n,k)$ *gives* $N-1$ *as an annihilating operator i.e* $f(n+1) - f(n) = 0$. *Thus, we proved the identity!*

2. *Moreover, Zeilberger's algorithm gives* $G(n,k) = \frac{(k-1)((n+2)k-n-1)}{2(k-n-1)(n+2)k}$ *which is our proof certificate!*

However, we still did not answer the question: Why this special case of Zeilberger's algorithm is important? The answer is two-fold:

1. *First, we can use it to give very simple proofs of identities. Since, once* $G(n,k)$ *is given, we can easily check the correctness of the identity.*

2. *Second, this gives us some bonus identities! See, Chapter 7 of Petkovšek et al. (1996).*

6. Algorithm Hyper

In Chapter 5 and Chapter 3 we have seen Sister Celine's algorithm and creative telescoping algorithm. The aim of both algorithms is to find a recurrence satisfied by our sum. To be more precise, they do the following job:

Given $f(n) = \sum_k F(n,k)$, where $F(n,k)$ is a proper hypergeometric term. These algorithms gives us a recurrence of the form

$$\sum_{i=0}^{m} a_i(n) f(n+i) = 0. \tag{6.1}$$

Then there are two options:

1. Equation (6.1) is an order 1 recurrence, then we are done. Since,

   $$\sum_{i=0}^{1} a_i(n) f(n+i) = 0 \to f(n) = f(0) \prod_{j=0}^{n-1} \frac{-a_0(n)}{a_1(n)}.$$

   Thus, we really find $f(n)$.

2. If its order is greater than equal to 2, then we still need to solve the recurrence to find $f(n)$.

The aim of algorithm hyper is to find hypergeometric solutions of this recurrence, given a linear recurrence of the form (6.1). Importantly, algorithm hyper can also shows (proves!) the non-existence of hypergeometric solutions.

Moral of the Story: Combining the creative telescoping algorithm with algorithm hyper, we can answer the following question completely: Given a sum of the form $f(n) = \sum_k F(n,k)$ where $F(n,k)$ is a proper hypergeometric function, check whether $f(n)$ is also hypergeometric or not. We can proceed as follows:

1. Use Zeilberger's algorithm to generate a recurrence that annihilates $f(n)$.

2. Use algorithm hyper to check whether this recurrence has a hypergeometric solution or not.

Before starting with examples as usual, we will divide our problem into different categories:

1. **Problem1**: Given a linear recurrence operator L, find all polynomials P such that $LP = 0$.
2. **Problem2**: Given a linear recurrence operator L, find all rational functions(sequences) R such that $LR = 0$.
3. **Problem3**: Given a linear recurrence operator L, find all hypergeometric functions(sequences) H such that $LH = 0$.

We will not answer Problem2. For details of these, see Abramov (1995).

Remark 30. *We can ask similar questions with different operators as well. For example, we can check whether a differential equation has a polynomial solution or not. We will look at this in a bit more detail in Chapter 7.*

Let us start with Problem1. We start with a relatively easy case: A homogeneous recurrence of order 2. We try to find all polynomial solutions of it:

Example 31. *Let's find polynomial solutions of*

$$(n+3)(n-5)y(n+2)+(n^3+5n-3)y(n+1)-(n+1)(n-7)y(n)=0.$$

Solution:

1. Let's start with fixing our constants:
   (a) First, the coefficients of $y(n+2)$, $y(n+1)$ and $y(n)$ are $p(n)=(n+3)(n-5)$, $q(n)=n^3+5n-3$ and $r(n)=-(n+1)(n-7)$ respectively. Thus, the the maximum degree of the coefficient, m, is 3.
   (b) Second, we will write the coefficients as follows:

$$\begin{aligned} p(n) &= u_0n^m+u_1n^{m-1}+u_2n^{m-2}+O(n^{m-3}) = 0\cdot n^3+1\cdot n^2-2n+O(1) \\ q(n) &= v_0n^m+v_1n^{m-1}+v_2n^{m-2}+O(n^{m-3}) = 1\cdot n^3+0\cdot n^2+5n+O(1) \\ r(n) &= w_0n^m+w_1n^{m-1}+w_2n^{m-2}+O(n^{m-3}) = 0\cdot n^3-n^2+6n+O(1) \end{aligned}$$

   Thus, we have $(u_0,u_1,u_2)=(0,1,-2)$, $(v_0,v_1,v_2)=(1,0,5)$ and $(w_0,w_1,w_2)=(0,-1,6)$.
2. Now observe that the leading coefficients of $p(n)$, $q(n)$ and $r(n)$ do not cancel each other, i.e $u_0+v_0+w_0\neq 0$, thus $D:=\emptyset$.

3. Lastly, since D is an empty set, we cannot find a polynomial solution to our recurrence.

Note that intuitively, our result makes sense. Since the right hand side of the recurrence is 0 which means that if at the left hand side the leading term does not vanish, i.e $u_0+v_0+w_0 \neq 0$, then we cannot find a polynomial solution! In other words, $u_0+v_0+w_0 = 0$ is certainly a necessary condition to hold. However, as we will see it is not sufficient.

Let's try another example:

Example 32. *Find polynomial solutions of*

$$n(n+1)y(n+2) - 2n(n+10)y(n+1) + (n+9)(n+10)y(n) = 0.$$

Solution:

1. First, we will fix our constants:

   (a) The polynomial coefficients of $y(n+2)$, $y(n+1)$ and $y(n)$ are $n(n+1)$, $-2n(n+10)$ and $(n+9)(n+10)$ respectively. Thus, $p(n) = n(n+1)$, $q(n) = -2n(n+10)$ and $r(n) = (n+9)(n+10)$. The maximum degree of the coefficients is 2, i.e $m = 2$.

   (b) Let's write the coefficients as in the above example:

   $$\begin{aligned} p(n) &= u_0 n^m + u_1 n^{m-1} + u_2 n^{m-2} + O(n^{m-3}) = 1 \cdot n^2 + n + 0 \\ q(n) &= v_0 n^m + v_1 n^{m-1} + v_2 n^{m-2} + O(n^{m-3}) = -2 \cdot n^2 - 20n \\ r(n) &= w_0 n^m + w_1 n^{m-1} + w_2 n^{m-2} + O(n^{m-3}) = 1 \cdot n^2 + 19n + 90. \end{aligned}$$

   Thus, we have $(u_0,u_1,u_2) = (1,1,0)$, $(v_0,v_1,v_2) = (-2,-20,0)$ and $(w_0,w_1,w_2) = (1,19,90)$.

2. Now, we need to check bunch of equalities holds for u, v and w:

   (a) Observe that this time the leading coefficients do cancel out, i.e $u_0+v_0+w_0 = 1-2+1 = 0$.

   (b) Next, we need to check whether u_0 is equal to w_0 or not. Since $u_0 = w_0 = 1$, we go to the next step.

   (c) This time, we compute $u_1+v_1+w_1 = 1-20+19 = 0$. Since, it is 0, we take $D := \{N \in \mathbb{N} : u_0 N^2 + (u_1 - u_0 - w1)N + u_2 + v_2 + w_2 = 0\} = \{$ integer roots of $N^2 - 19N + 90 = 0\} = \{9, 10\}$.

3. Let k be the maximum element of D, i.e $k = 10$. It means that, we are looking for polynomial solution of our equation whose degree is at most 10. Thus, we write $y(n) = \sum_{i=0}^{10} c_i n^i$ in place of $y(n)$ in our recurrence. Then, we get(via Maple of course!)

$$\begin{aligned} B := \{n^{10} - 750n^8 - 15120n^7 - 140847n^6 - 740880n^5 - 2304100n^4 - \\ 4142880n^3 - 3904704n^2 - 1451520n, n^9 + 36n^8 + \\ 546n^7 + 4536n^6 + 22449n^5 + \\ 67284n^4 + 118124n^3 + 109584n^2 + 40320n\} \end{aligned}$$

is a basis of solution set of our recurrence.

Now we can look at the algorithm of finding polynomial solutions of order 2 recurrences:

Algorithm 2: Algorithm poly for order 2 recurrences

Input: A recurrence of the form

$$p(n)y(n+2) + q(n)y(n+1) + r(n)y(n) = 0$$

where

$$\begin{aligned} p(n) &= u_0 n^m + u_1 n^{m-1} + u_2 n^{m-2} + O(n^{m-3}), \\ q(n) &= v_0 n^m + v_1 n^{m-1} + v_2 n^{m-2} + O(n^{m-3}), \\ r(n) &= w_0 n^m + w_1 n^{m-1} + w_2 n^{m-2} + O(n^{m-3}). \end{aligned}$$

m is the maximum degree of coefficient polynomials.

Output: A basis B for the space of solutions of our recurrence.

If $u_0 + v_0 + w_0 \neq 0$, then $D := \emptyset$.

else if $u_0 \neq w_0$, then $D := \{N \in \mathbb{N} : (u_0 - w_0)N + u_1 + v_1 + w_1 = 0\}$

else if $u_1 + v_1 + w_1 \neq 0$, then $D := \emptyset$

else $D := \{N \in \mathbb{N} : u_0 N^2 (u_1 - u_0 - w_1)N + u_2 + v_2 + w_2 = 0\}$.

If $D = \emptyset$, then $B := \emptyset$.

else $k := maxD$; find a basis B for the space of polynomials over $\mathbb{K}$ of degree at most k.

Return B

Note that our algorithm shows that our intuition is correct: If $u_0 + v_0 + w_0 \neq 0$, then there is no polynomial solution!

Before generalizing our algorithm to an arbitrary order recurrence, let's summarize the main point:

Moral of the Story: In general, if we have a problem whether a particular equation has a polynomial solution, say P, or not. We proceed as follows:

1. Find a degree bound for the polynomial, say d.
2. Substitute a generic degree d polynomial instead of P. Then, use method of undetermined coefficients to find the coefficients of our generic polynomial. If we succeed, we are done. Else, we showed(proved!) that this equation does not have a polynomial solution.

Remark 31. *After finding degree bound for the polynomial, there exists some other methods than undetermined coefficients to find the coefficients of our generic polynomial. In some cases, especially if we know that some properties of the polynomial, then we can use other (faster) methods. See Abramov, Bronstein & Petkovšek (1995).*

Before proceeding further, we will try to motivate our algorithm for $d = 2$:

1. Our intention is to find polynomial solutions of

$$p(n)y(n+2)+q(n)y(n+1)+r(n)y(n) = 0.$$

   where $p(n)$, $q(n)$ and $r(n)$ are given polynomials, $y(n)$ is our unknown. Our aim is to find a degree of $y(n)$. Suppose that $y_(n)$ is a polynomial solution to our equation and $deg(y(n)) = N$. We want to find N if possible.

2. Let m be the maximum degree of our polynomial coefficients, i.e $m = max\{degp(n), degq(n), degr(n)\}$. Then,

$$\begin{aligned} p(n) &= p_0n^m + p_1n^{m-1} + p_2n^{m-2} + O(n^{m-3}),\\ q(n) &= q_0n^m + q_1n^{m-1} + q_2n^{m-2} + O(n^{m-3}),\\ r(n) &= r_0n^m + r_1n^{m-1} + r_2n^{m-2} + O(n^{m-3)}. \end{aligned}$$

   where at least one of p_0, q_0 and r_0 is nonzero. With a similar spirit,

$$z(n) = z_Nn^N + z_{N-1}n^{N-1} + z_{N-2}n^{N-2} + O(n^{N-3}).$$

   where Z_N is nonzero.

3. Now, we can compute $z(n+1)$ and $z(n+2)$:

(a) We have

$$z(n+1) = z_N n^N + (Z_{N-1} + NZ_N)n^{N-1} + (Z_{N-2} + (N-1)Z_{N-1} + \binom{N}{2} Z_N)n^{N-2} + O(n^{N-3}),$$

(b) Similarly,

$$z(n+2) = z_N n^N + (Z_{N-1} + 2NZ_N)n^{N-1} + (Z_{N-2} + 2(N-1)Z_{N-1} + 4\binom{N}{2} Z_N)n^{N-2} + O(n^{N-3}).$$

4. Let's substitute $z(n)$, $z(n+1)$ and $z(n+2)$ to our recurrence, then we look at the coefficient of the following terms at the left hand side:

   (a) The coefficient of n^{N+m} is $p_0 + q_0 + r_0$. Thus, we must have $p_0 + q_0 + r_0 = 0$. If this is not satisfied, there is no polynomial solution to our recurrence.

   (b) The coefficient of n^{N+m-1} is $A = (2p_0 + q_0)N + p_1 + q_1 + r_1$, using $p_0 + q_0 + r_0 = 0$. Thus, we must have $A = 0$. If $2p_0 + q_0 \neq 0$. Then, we have only one choice of N, thus we are done. Note that since $p_0 + q_0 + r_0 = 0$, $2p_0 + q_0 \neq 0$ is equivalent to $p_0 \neq r_0$. If $p_0 = r_0$, then we must have $p_1 + q_1 + r_1 = 0$, otherwise we cannot have a polynomial solution to our recurrence.

   (c) Let's consider the coefficient of n^{m-2}. This time we have $B = p_0 N^2 + (2p_1 - p_0 + q_1)N + p_2 + q_2 + r_2$ as a coefficient. Thus, we must have $B = 0$. As a result, we have at most two possible values for N. Thus, we are done!

Let's generalize our algorithm for arbitrary order recurrences:

Algorithm 3: Algorithm Poly for order d recurrences

Input: A recurrence of the form

$$\sum_{k=0}^{d} p_k(n)y(n+k) = 0,$$

where

$$p_k(n) = \sum_{i=0}^{m} c_{k,i} n^{m-i},$$

and m is the maximum degree of coefficient polynomials.

Output: Find a basis B for the space of solutions of our recurrence.

initialize s=-1;

repeat

increment s by 1;

for j from 0 to s compute

$b_j^{(s)} = \sum_{i=0}^{d} i^j c_{i,s-j}$;

until there exists a j, $0 \leq j \leq s$ such that $b_j^{(s)} \neq 0$

Let S be the set of non-negative integer roots N of the polynomial

$$D(N) = \sum_{j=0}^{s} \binom{N}{j} b_j^{(s)};$$

If $S = \emptyset$, then $B = \emptyset$

else

k:=maxS;

find a basis B for the space of polynomial solutions of our recurrence of degree at most k, using the method of undetermined coefficients.

return B and stop.

Let's apply this algorithm for a recurrence of order 3:

Example 33. *Find polynomial solutions of*

$$(n+2)(n+5)y(n)+(n-2)(n+3)y(n+1)+n(n+1)y(n+2)+(-3n^2+5n+9)y(n+3) = 0.$$

Solution:

1. Firstly, let's fix our constants as usual:

(a) The coefficients of $y(n)$, $y(n+1)$, $y(n+2)$ and $y(n+3)$ are

$$p_0(n) = (n+2)(n+5),$$

$$p_1(n) = (n-2)(n+3),$$

$$p_2(n) = n(n+1)$$

and

$$p_3(n) = (-3n^2+5n+9)$$

respectively.

(b) Initialize s as $s=-1$ and $d=3$.

2. Increasing s by 1, s becomes 0. Let $j=0$. We need to compute $b_0^{(0)}$:

$$b_0^{(0)} = \sum_{i=0}^{3} c_{i,0} = 1+1+1-3 = 0.$$

3. Since, $b_0^{(0)} = 0$ we continue with increasing s by 1. s becomes 1. This time we will compute $b_0^{(1)}$:

$$b_0^{(1)} = \sum_{i=0}^{3} c_{i,1} = 7+1+1+5 = 14,$$
$$b_1^{(1)} = \sum_{i=0}^{3} c_{i,0} = 1+1+1-3 = 0.$$

Since $b_0^{(1)} \neq 0$, we stop there.

4. Now, we need to find the integer roots of the following polynomial:

$$D(N) = \sum_{j=0}^{1} \binom{N}{j} b_j^{(1)} = 14.$$

Since $D(N)$ has no integer roots, we cannot find a polynomial solution to our recurrence!

Let us try to solve Problem3. Now, we will try to find hypergeometric solutions. Like in the polynomial case, we start with a recurrence of order 2, and try to find its hypergeometric solutions.

Example 34. *Find a hypergeometric solution of*

$$(-n-4)h(n+2)+(n+3)h(n+1)+(2n+4)h(n) = 0.$$

Solution:

1. Firstly, we fix our constants as usual: The coefficients of $h(n+2)$, $h(n+1)$ and $h(n)$ are $p(n) = -(n+4)$, $q(n) = (n+3)$ and $r(n) = (2n+4)$.

2. Now, we need to find all monic factors of $r(n)$ and $p(n-1)$: Obviously, $r(n)$ and $p(n-1)$ have only one monic factor: $A(n) = n+2$ and $B(n) = (n+3)$ respectively. Let us define $P(n)$, $Q(n)$ and $R(n)$ as follows:

$$P(n) = \frac{p(n)}{B(n+1)}A(n+1) = \frac{-(n+4)}{(n+4)}(n+3) = -(n+3),$$
$$Q(n) = q(n) = (n+3),$$
$$R(n) = \frac{r(n)}{A(n)}B(n) = \frac{(2n+4)}{(n+2)}(n+3) = 2(n+3).$$

3. Let m be the maximum degree of $P(n)$, $Q(n)$ and $R(n)$, i.e $m = 1$. Let α, β and γ be the coefficients of n^m in $P(n)$, $Q(n)$ and $R(n)$ respectively. Thus, $(\alpha, \beta, \gamma) = (-1, 1, 2)$.

4. Construct $-Z^2 + Z + 2 = 0$. Then our roots are -1 and 2.

5. Now we need to construct two new recurrences:

   (a) When $Z = -1$ we have

$$Z^2 P(n)C(n+2) + ZQ(n)C(n+1) + R(n)C(n) = 0,$$
$$-(n+3)C(n+2) - (n+3)C(n+1) + 2(n+3)C(n) = 0.$$

   Now we need to find whether this recurrence has polynomial solution or not. Using Algorithm Poly we get any constant D is a solution to our recurrence. Then, we will compute $S(n)$ as follows:

$$S(n) = Z\frac{A(n)}{B(n)}\frac{C(n+1)}{C(n)} = \frac{-(n+2)}{(n+3)}.$$

   Lastly, we find a non-zero solution $a(n)$ of $a(n+1) = S(n)a(n)$. Then, $a(n) = \frac{(-1)^n}{n+2}$ is a solution. Actually, we can stop here since we find a solution. However, let's see what is going to happen when $Z = 2$ as well.

   (b) When $Z = 2$ we have

$$Z^2 P(n)C(n+2) + ZQ(n)C(n+1) + R(n)C(n) = 0,$$
$$-4(n+3)C(n+2) + 2(n+3)C(n+1) + 2(n+3)C(n) = 0.$$

Now we need to check whether this recurrence has a polynomial solution or not. Using Algorithm Poly we found that any constant D is a solution. Again, we will compute $S(n)$:

$$S(n) = Z\frac{A(n)}{B(n)}\frac{C(n+1)}{C(n)} = \frac{2(n+2)}{(n+3)}.$$

Again, we will search for a non-zero solution $a(n)$ of $a(n+1) = S(n)a(n)$. Then, $a(n) = \frac{2^n}{n+2}$ is also a solution. Thus, we are done!

Now, we try to find hypergeometric solutions of a recurrence of order 3:

Example 35. *Find a hypergeometric solution of*

$$(n^2+1)y(n)+(n^3-n+1)y(n+1)+(n^3-2n+1)y(n+2)+(n+8)y(n+3) = 0.$$

Solution:

1. As usual, we start by fixing our constants:
   (a) The coefficients of $y(n)$, $y(n+1)$, $y(n+2)$ and $y(n+3)$ are
   $$p_0(n) = n^2+1,$$
   $$p_1(n) = n^3-n+1,$$
   $$p_2(n) = n^3-2n+1$$
   and
   $$p_3(n) = n+8$$
   respectively.

   (b) The order of the recurrence is $d = 3$.

2. Firstly, we need to find all monic factors of $p_0(n)$ and $p_3(n-3+1)$:
   (a) It is obvious that $p_0(n) = n^2+1$ has only one monic factor over $\mathbb{R}$ which is $A(n) = n^2+1$.

   (b) Similarly, only monic factor of $p_3(n-2) = n+6$ is the $B(n) = n+6$.

3. Now, we need to compute $P_i(n) = p_i(n)\prod_{j=0}^{i-1} A(n+j)\prod_{j=i}^{d-1} B(n+j)$ for $i =$

0, 1, 2, 3:

$$
\begin{aligned}
P_0(n) &= p_0(n)B(n)B(n+1)B(n+2) = (n^2+1)(n+6)(n+7)(n+8),\\
P_1(n) &= p_1(n)A(n)B(n+1)B(n+2) = (n^3-2n+1)(n^2+1)(n+7)(n+8),\\
P_2(n) &= p_2(n)A(n)A(n+1)B(n+2) = (n^3-2n+1)(n^2+1)(n^2+2n+2)(n+8),\\
P_3(n) &= p_3(n)A(n)A(n+1)A(n+2) = (n+8)(n^2+1)(n^2+2n+2)(n^2+4n+5).
\end{aligned}
$$

Then, the maximum degree of $P_i(n)$'s is $m = 8$. Now, we will take the coefficient of n^8 from each $P_i(n)$ as follows $(\alpha_0, \alpha_1, \alpha_2, \alpha_3) = (0, 0, 1, 0)$.

4. Now, we will construct

$$\sum_{i=0}^{3} \alpha_i Z^i = 0 \rightarrow Z^2 = 0.$$

Thus, there is no non-zero solution. It means that there is no hypergeometric solution to our recurrence!

Let's look at the algorithm itself:

Algorithm 4: Algorithm Hyper for order d recurrences

Input: A recurrence of the form

$$\sum_{k=0}^{d} p_k(n)y(n+k) = 0$$

Output: A hypergeometric solution of the recurrence, if it exists.

For all monic factors $A(n)$ of $p_0(n)$ and $B(n)$ of $p_d(n-d+1)$ do:

$P_i(n) = p_i(n)\prod_{j=0}^{i-1} A(n+j)\prod_{j=i}^{d-1} B(n+j)$ for $i = 0,1,...,d$;

$m := \max_{0\le i\le d} degP_i(n)$;

Let α_i be the coefficient of n^m in $P_i(n)$, for $i = 0,1,...,d$;

for all non-zero Z such that

$$\sum_{i=0}^{d} \alpha_i Z^i = 0.$$

do:

If the recurrence

$$\sum_{i=0}^{d} Z^i P_i(n)C(n+i) = 0$$

has a non-zero polynomial solution $C(n)$ then,

$$S(n) = Z\frac{A(n)}{B(n)}\frac{C(n+1)}{C(n)};$$

return a non-zero solution $a(n)$ of $a(n+1) = S(n)a(n)$ and stop.

Return 0 and **stop**.

Remark 32. *In our description, algorithm hyper finds a hypergeometric solution to a recurrence if it exists, then stops. An obvious question is the following: "Can we find all hypergeometric solutions of the recurrence?" The answer is yes. In other words, we can extend Algorithm Hyper in such a way that, it will give us a basis of the solution set. For more information see Petkovšek (1992).*

In Chapter 5 Example 26, we try to evaluate $f(n) = \sum_k \binom{n}{k}^3$ using Zeilberger's algorithm. We find that $L = (n+2)^2N^2 - (7n^2+21n+16)N - 8(n+1)^2$ annihilates $f(n)$. In other words,

$$(n+2)^2 f(n+2) - (7n^2+21n+16)f(n+1) - 8(n+1)^2 f(n) = 0.$$

Let's check whether this recurrence has a hypergeometric solution or not.

Example 36. *Check whether*

$$REC := (n+2)^2 f(n+2) - (7n^2+21n+16) f(n+1) - 8(n+1)^2 f(n) = 0$$

has a hypergeometric solution or not.

Solution: Since solving this question without computer takes a lot of time, let us use Maple to solve it. The following code $hypergeomsols(REC, f(n),)$ gives 0 meaning that it has no hypergeometric solution!

Using algorithm hyper we can easily establish the following theorem:

Theorem 7. *None of the following sequences can be expressed in a hypergeometric closed form:*

1. *The sum of the cubes of the binomial coefficients of order n, i.e* $\sum_{k=0}^{n} \binom{n}{k}^3$.
2. *The number of* $3 \times n$ *Latin rectangles.*
3. *The number of involutions of* n *letters.*
4. *The derangement numbers.*
5. *The sum of first* n *of the binomial coefficients of order* pn.

Now,we present some computer generated examples to use both algorithm hyper and Zeilberger's algorithm to evaluate some sums.

Example 37. 1. *(Putnam 1990 A1,Kedlaya, Poonen & Vakil (2020)) Let* $T_0 = 2$, $T_1 = 3$ *and* $T_2 = 6$.

$$T_n = (n+4)T_{n-1} - 4nT_{n-2} + (4n-8)T_{n-3}.$$

Find a formula for $T_n = A_n + B_n$ *where* A_n *and* B_n *are well-known sequences.*
Solution:
Using Maple we write hypergeomsols(rec,T(n),{}) which gives $(2T(0) - T(1))\Gamma(n+1) + (-T(0)+T(1))2^n$. *Substituting* $T(0) = 2$, $T(1) = 3$ *and* $\Gamma(n+1) = n!$ *directly gives* $T_n = n! + 2^n$.

2. *(AMM Problem 11212,Beckwith, Kwong, Pratt & Singer (2008)) Prove that*

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

for all positive integer n.
Solution:

(a) *Using Zeilberger's algorithm with Maple gives us a recurrence of order* 1*:*

$$(n+1)(n-2)f(n)+(2n+4)f(n+1)=0.$$

(b) *Using algorithm hyper gives us a solution to our recurrence is* $f(0)$*. Checking* $f(0)=0$ *proves it.*

3. *(AMM 2008 Problem 11356, Poghosyan (2008)) Prove that for any positive integer* n*,*

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

Solution:

(a) *Zeilberger's algorithm gives us a recurrence of order* 1*:*

$$-4(n+1)^2f(n)+(2n+3)(2n+1)f(n+1)=0.$$

(b) *Using algorithm hyper, we have* $\frac{2^{4n}(n!)^4}{(2n)!(2n+1)!}$*. Hence, we are done!*

Remark 33. *All of these algorithms can be extended to non-homogeneous recurrences, as well. In other words, it is possible to answer the following questions:*

1. *Given a linear recurrence operator* L *and a polynomial* f*, find polynomial solutions,* y*, of* $Ly=f$*.*

2. *Given a linear recurrence operator* L *and a function* f*, find hypergeometric solutions,* y*, of* $Ly=f$*.*

See Petkovšek (1992).

Remark 34. *There is an application of algorithm hyper to factorize linear recurrence operators. To be more precise, we can use Hyper to solve the following problem: "Given a linear recurrence operator* L *of the form* (2.4) *find all linear operators* $L^{'}$ *such that* $L=QL^{'}$ *where* Q *is also a linear operator." See Petkovšek (1992).*

7. Inverse Zeilberger Problem

This chapter in nutshell:

1. Definition of the inverse Zeilberger problem.
2. Targeting a certain small subclass of this problem.
3. A good factorial basis and a good linear transformation.

In Chapter 5 we saw the creative telescoping algorithm which gives a recurrence satisfied by a sum. A natural question that comes to mind is: "Given a recurrence relation, can we find a sum that satisfies the recurrence?" In other words, it is the inverse problem Zeilberger's problem in some sense. To be more precise:

Zeilberger Problem : Given a sum $f(n) = \sum_k F(n,k)$, where $F(n,k)$ is a proper hypergeometric term, can we find a linear recurrence operator L of the form

$$\sum_{i=0}^{r} p_0(n) N^i \tag{7.1}$$

where $p_i(n)$'s are polynomials in n, N is the shift operator with respect to n, such that $Lf = 0$?

Inverse Zeilberger Problem: Given a linear recurrence operator L of the form (7.1) can we find a sum $f(n) = \sum_k F(n,k)$ such that $Lf = 0$?

As usual, we will resctrict our attention to a certain subclass of this problem:

Small Inverse Zeilberger Problem : Given a linear recurrence operator L of the form (7.1) and natural numbers m, a_1, a_2, ..., a_m and real numbers b_1, b_2, ..., b_m can we find a sum $f(n)$ of the form

$$f(n) = \sum_{k=0}^{\infty} H(n,k) t_k, \tag{7.2}$$

where $H(n,k)$ is given as

$$H(n,k) = \prod_{i=1}^{m} \binom{a_i n + b_i}{k},$$

such that L annihilates $f(n)$, i.e. $Lf = 0$? Note that due to the form of $H(n,k)$, we can extend the sum in (7.2) from $k = -\infty$ to ∞.

Before looking at examples we need to answer some questions:

1. Why are we considering linear operators of the form (7.1)? Is it possible to generalize it? If yes, how? If no, why?

2. What is so special about binomial coefficients? Can we look at the sums without binomial coefficients?

Answers:

1. In Zeilberger's algorithm we always use the shift operator. Therefore, we start with looking at linear operators in terms of shift operators since we are trying to answer inverse Zeilberger problem! However, obviously, we can use some other linear operators as well. For a concrete example, we can take differentiation operator and consider following three problems:

   (a) **Continuous Zeilberger Problem**: Given an indefinite integral $\int y(x)dx$ find a differential operator T with polynomial coefficient such that

   (7.3) $$T = \sum_{k=0}^{r} p_k(x) D^k(x),$$

   where $p_k(x)$ are polynomials, D is the differentation operator and $T \int y(x)dx = 0$.

   (b) **Inverse Continuous Zeilberger Problem:** Given a differential operator T of the form (7.3) find $\int y(x)dx$ such that $T \int y(x)dx = 0$.

   (c) **Small Inverse Continuous Zeilberger Problem**: Given a differential operator T of the form (7.3) find $\int y(x)dx$ such that $y(x)$ is of the form $e^x G(x)$ and $T \int y(x)dx = 0$. In other words, we force the form of $y(x)$.

   Moral of the Story: We can choose different linear operators and consider different variations of (Inverse) Zeilberger's problem.
   In any case, we will focus on the shift operator.

2. First, we use binomial coefficients since binomial basis works fine with the

shift operator (explained later). Second, we guarantee finite support, i.e. we do not need to worry about convergence issues at all!

Remark 35. *Note that we will consider a special case of Inverse Zeilberger problem, called Small Inverse Zeilberger Problem. It is possible to consider another special case as well:*

Inverse Constant Coefficient Zeilberger Problem: *Given a linear recurrence operator* $L = \sum_{i=0}^{r} c_i N^i$ *where* c_i*'s are constants, find* $f(n) = \sum_k F(n,k)$ *such that* $Lf(n) = 0$.

Before discussing details of the questions and examples let's clarify the notation and give a definition:

Definition 14. *Let* $\mathbb{K}$ *be a field. Then,* $\mathbb{K}[X]$ *the* ***ring of the polynomials*** *whose coefficients are coming from* $\mathbb{K}$*. Similarly,* $\mathbb{K}[[X]]$ *is the* ***ring of formal power series****. In other words, elements of* $\mathbb{K}[X]$ *is of the form* $f(x) = \sum_{k=0}^{n} a_k x^k$ *where* a_k *is an element of* $\mathbb{K}$*. Elements of* $\mathbb{K}[[X]]$ *is of the form* $g(x) = \sum_{k=0}^{\infty} a_k x^k$ *where each* a_k *is an element of* $\mathbb{K}$*.*

Remark 36. *With the above definitions, clearly we can see* $\mathbb{K}[X]$ *as a subring of* $\mathbb{K}[[X]]$ *and* $\mathbb{K}$ *as a subring of* $\mathbb{K}[X]$*.*

Let $L = \sum_{i=0}^{r} a_i(n) N^i$ be a linear recurrence operator where $a_i(n)$'s are polynomials. Suppose that we somehow find $f(n) = \sum_{k=0}^{\infty} \prod_{i=1}^{m} \binom{a_i n + b_i}{k} h_k$ such that $Lf(n) = 0$. Thus,

$$Lf(n) = \sum_{k=0}^{\infty} h_k L \prod_{i=1}^{m} \binom{a_i n + b_i}{k} = 0.$$

If somehow, we understand the behaviour of L on $\prod_{i=1}^{m} \binom{a_i n + b_i}{k}$ and this behaviour is *nice*, we can understand its behaviour on $f(n)$. Note that we can see $\binom{a_i n + b_i}{k}$ as a polynomial in n of degree k for any fixed k. Hence, we try to understand effects of L on polynomials. We will make this idea more precise along the way.

Since we try to answer a question involving a linear operator, $L : \mathbb{K}[X] \to \mathbb{K}[X]$, to understand its behaviour on elements of $\mathbb{K}[X]$ it is enough to understand its effect on basis elements. However, we do not have a basis of $\mathbb{K}[X]$ yet. Thus, we need to find a basis of $\mathbb{K}[X]$. Obviously, we have a lot of different choices (even uncountably many choices!). So the obvious question is: Which basis to choose? What is our choosing criteria? We want a basis which works nicely with linear operators. However, finding a basis which works *nicely* with all linear operators is too much to ask. As noted above, our main linear operator is the shift operator. Thus, we want to find a basis which works *well* with the shift operator. We still need to answer the question: What does working nicely with an operator mean? We will answer this question later.

If we recall Gosper's algorithm and Zeilberger's algorithm, we use two main tools with polynomials: Degree of the polynomial and divisibility of polynomials, see Chapter5 and Chapter6. Thus, it is good to have a basis which gives us some control over these. It is even better to have a definition of a basis which satisfies these criterion!

Definition 15. *A sequence of polynomials $\{P_k(x)\}_{k=0}^{\infty}$ is called **factorial basis** of $\mathbb{K}[X]$ if the following conditions hold:*

1. $P_k | P_{k+1}$,
2. $deg P_k = k$.

Remark 37. *Note that the second condition in the definition guarantees that, $\{P_k(x)\}_{k=0}^{\infty}$ is indeed a basis of $\mathbb{K}[X]$!*

Example 38.
1. *Obviously, choosing $P_k(x) = x^k$ gives us a factorial basis.*
2. *Choosing $P_k(x) = \binom{x}{k}$ also gives a factorial basis.*
3. *Also, we can choose $P_k(x) = x^{\underline{k}} = x(x-1)(x-2)..(x-k+1)$.*
4. *Similarly, $P_k(x) = x^{\bar{k}} := x(x+1)..(x+k-1)$ is a factorial basis as well.*
5. *We can generalize the second basis as $P_k(x) = \binom{x-a}{k}$ for any a.*
6. *Lastly, $P_k(x) = \frac{(x-a)^k}{k!}$ is also a factorial basis for any a.*

Before trying to motivate the factorial basis definition let's look at expression of some polynomials over these bases:

1. Let $f(x) = x^3 + 4x^2 - 5x + 7$. Then,
   (a) If we choose our basis as $< P_k(x) = x^k >_{k=0}^{\infty}$, then, obviously, our coefficients are $(7, -5, 4, 1)$.
   (b) If we choose our basis as $< P_k(x) = \binom{x}{k} >_{k=0}^{\infty}$, then,

$$f(x) = x^3 + 4x^2 - 5x + 7 = 7\binom{x}{0} + 0\binom{x}{1} + 14\binom{x}{2} + 6\binom{x}{3}.$$

Thus, our coefficients are (7,0,14,6)

(c) If we choose our basis as $< P_k(x) = x^{\underline{k}} >_{k=0}^{\infty}$, then,

$$f(x) = x^3 + 4x^2 - 5x + 7 = 7x^{\underline{0}} + 0x^{\underline{1}} + 7x^{\underline{2}} + 1x^{\underline{3}}.$$

Thus, our coefficients are $(7,0,7,1)$.

(d) If we choose our basis as $< P_k(x) = x^{\bar{k}} >_{k=0}^{\infty}$, then,

$$f(x) = x^3 + 4x^2 - 5x + 7 = 7x^{\bar{0}} - 9x^{\bar{1}} + 1x^{\bar{2}} + 1x^{\bar{3}}.$$

(e) Choosing $< P_k(x) = \binom{x-a}{k} >_{k=0}^{\infty}$ gives us

$$f(x) = x^3 + 4x^2 - 5x + 7 = (a^3 + 4a^2 - 5a + 7)\binom{x-a}{0} + (3a^2 + 11a)\binom{x-a}{1} + (6a + 14)\binom{x-a}{2} + 6\binom{x-a}{3}.$$

Thus, our coefficients are $((a^3 + 4a^2 - 5a + 7), (3a^2 + 11a), (6a + 14), 6))$.

(f) Lastly, $\langle P_k(x) = \frac{(x-a)^k}{k!} \rangle_{k=0}^{\infty}$ gives us:

$$f(x) = x^3 + 4x^2 - 5x + 7 = (-2a^3 + 5a^2 - 5a + 7)\frac{(x-a)^0}{0!} + (-3a^2 + 10a - 5)\frac{(x-a)}{1!} + 10\frac{(x-a)^2}{2!} + 6\frac{(x-a)^3}{3!}.$$

Remark 38. *We note that the elements of* $\mathbb{K}[X]$ *is of the form* $\sum_{k=0}^{d} c_k x^k$. *However, we can pick a factorial basis,* $\beta =< P_k(x) >_{k=0}^{\infty}$, *of* $\mathbb{K}[X]$ *and consider* $\mathbb{K}[B]$ *whose elements are of the form* $\sum_{k=0}^{d} c_k P_k(x)$. *The same applies to* $\mathbb{K}[[X]]$. *In other words, we can talk about* $\mathbb{K}[[B]]$ *whose elements are of the form* $\sum_{k=0}^{\infty} c_k P_k(x)$.

Remark 39. *Note that the following famous formula can be seen as a expression over a particular factorial basis:* $p(x) = \sum_{k=0}^{degp} p^{(n)}(a)\frac{(x-a)^n}{n!}$ *is the Taylor expansion of* $p(x)$ *about* $x = a$. *It is the actually the expression of* $p(x)$ *with respect to the factorial basis* $P_k(x) = \frac{(x-a)^k}{k!}$.

Let's give some examples of famous linear operators on $\mathbb{K}[x]$:

1. $Dp(x) = p^{'}(x)$, in other words D is the differentiation operator.

2. $Np(x) = p(x+1)$, in other words, N is the shift operator.

3. $Qp(x) = p(qx)$, i.e Q is the q-shift operator.

4. $Xp(x) = xp(x)$, i.e. X is the multiplication by X operator.

5. $\Delta p(x) = p(x-1)$, in other words, Δ is the difference operator. Obviously, N and Δ are inverses of each other.

Example 39. *Let $p(x) = x^2 + 4x + 7$. Then,*

1. $Dp(x) = (x^2 + 4x + 7)' = 2x + 4$.

2. $Np(x) = p(x+1) = x^2 + 6x + 8$.

3. $Qp(x) = p(qx) = q^2x^2 + 4qx + 7$.

4. $Xp(x) = xp(x) = x^3 + 4x^2 + 7x$.

5. $\Delta p(x) = p(x-1) = x^2 + 2x + 4$

Now we have bunch of bases and operators. Next, we need to define precisely the meaning of a factorial basis β working nicely with an operator L. Moreover, using our definition, we will look at our bases and operators to decide which ones to choose.

Remark 40. *Lastly, we need to clarify one thing about the algebraic setting we are working in: Normally, we are working on a polynomial ring $\mathbb{K}[X]$ where K is a field. However, each polynomial can be seen as a sequence! Observe that the following map shows that we can embed our polynomial ring into $\mathbb{K}^{\mathbb{N}}$: $\pi : \mathbb{K}[X] \to \mathbb{K}^{\mathbb{N}}$ where $\pi(p(x)) = \{p(k)\}_{k=0}^{\infty}$. We will see that seeing polynomials as sequences has some advantages. See **Chapter6** for similar ideas as well.*

Definition 16. *A factorial basis β of $\mathbb{K}[x]$ is **compatible** or **((A,B)-compatible)** with an operator L if there are $A, B \in \mathbb{N}$, and $a_{k,i} \in \mathbb{K}$, for $k \in \mathbb{N}$, $-A \leq i \leq B$ such that*

$$LP_k = \sum_{i=-A}^{B} a_{k,i} P_{k+i}$$

for all $k \in \mathbb{N}$ with $P_j = 0$ when $j < 0$.

Intuitively speaking a factorial basis β is compatible with an operator L means that L is behaving nicely with our basis. More precisely, but still informally speaking, β is compatible with L means that we can write LP_k as a linear combination of a **fixed number** of basis elements. In other words, the number of basis elements we use in the expansion of LP_k is independent of k. Before giving examples about the compatibility of operators, first we will prove a proposition which is useful to check whether a particular operator is compatible with a particular factorial basis or not.

Theorem 8. *A factorial basis β of $\mathbb{K}[x]$ is compatible with an operator L if and only if there are natural numbers A and B such that*

(C1) $degLP_k \leq k+B$ for all $k \geq 0$.

(C2) $P_{k-A}|LP_k$ for all $k \geq A$.

Proof:

1. Let's start with the easy direction. Assume that β is compatible with L. We want to show that $degLP_k \leq k+B$ for all $k \geq 0$. Since, β is compatible with L, we have

$$LP_k = \sum_{i=-A}^{B} \alpha_{k,i} P_{k+i}. \tag{7.4}$$

The degree on the right hand side of (7.4) is at most $B+k$; thus, the degree of the left hand side is at most $B+k$, i.e. $degLP_k \leq B+k$. With a similar reasoning, the degree of left hand side is at least $k-A$, if $k \geq A$. Since $P_k|P_{k+1}$, obviously we have $P_{k-A}|LP_k$.

2. Let's look at the reverse direction. Assume that we find natural numbers A and B such that $degLP_k \leq k+B$ and $P_{k-a}|LP_k$. We need to show that β is compatible with L, i.e., LP_k can be written as a linear combination of a fixed number of terms. Since LP_k is a polynomial, we can write it as a linear combination of our basis elements. However, we should use only a fixed number of basis elements.

3. By our first condition (C1) we know that $degLP_k \leq k+B$, also $degP_{k+B} = K+B$. Thus, it is clear that we do not need to use P_j's to write LP_k as a linear combination of basis elements where $j > k+B$. Thus, there exist scalars $c_{k,i}$ such that

$$LP_k = \sum_{i=0}^{degLP_k} c_{k,i} P_i.$$

Since $degLP_k \leq k+B$, we can write the above equation as

$$LP_k = \sum_{i=0}^{k+B} c_{k,i} P_i \rightarrow LP_k - \sum_{i=k-A}^{k+B} c_{k,i} P_i = \sum_{i=0}^{k-A-1} c_{k,i} P_i. \tag{7.5}$$

Now, observe that P_{k-A} divides the left hand side of (7.5) by the second condition(C2). Thus, P_{k-A} divides the right hand side, as well. Also, the degree of the right hand side is less than $k-A$. Hence, both sides must be equal to 0.

4. As a result we have

$$LP_k = \sum_{i=k-A}^{k+B} c_{k,i} P_i = \sum_{i=-A}^{B} c_{k,k+i} P_{k+i} = \sum_{i=-A}^{B} \alpha_{k,i} P_{k+i}.$$

   As a result, β is compatible with L.

Example 40. *Let's look at some bases and behaviours with respect to some linear operators:*

1. *Let $\beta = \{n^k\}_{k=0}^{\infty}$ be the standart basis of $\mathbb{K}[X]$. Then,*

   (a) *Let's consider the differentiation operator.*

   $$DP_k(x) = Dx^k = kx^{k-1} = kP_{k-1}(x).$$

   In other words, we can choose $A = 1$, $B = 0$ and $\alpha_{k,-1} = k$, $\alpha_{k,0} = 0$. Thus, the standart basis is compatible with the differentiation operator.

   (b) *Let's consider the shift operator.*

   $$NP_k(x) = Nx^k = (x+1)^k = \sum_{i=0}^{k} \binom{k}{i} x^i.$$

   Thus, we cannot choose A independent of k, i.e. the standart basis is not compatible with the shift operator N. In other words, we cannot write each basis element as a linear combination of fixed number of basis elements.

   (c) *Now, let's check the q-shift operator.*

   $$QP_k(x) = Qx^k = (qx)^k = q^k x^k.$$

   Thus, the standart basis is compatible with the q-shift operator, since we can choose $A = 0$, $B = 0$ and $\alpha_{k,0} = q^k$.

   (d) *Lastly, let's look at the multiplication by x operator.*

   $$XP_k = Xx^k = x^{k+1}.$$

   Hence, the standart basis is compatible with X, since we can choose $A = 0$, $B = 1$ and $\alpha_{k,0} = 0$, $\alpha_{k,1} = 1$.

2. *Let's look at the binomial-coefficient basis of $\mathbb{K}[X]$, i.e $\beta = \{\binom{x}{k}\}_{k=0}^{\infty}$:*

   (a) *Let's start with the differentiation operator. Binomial-coefficient basis is* *not compatible with the differentiation operator.*

(b) *Second, we have the shift operator*

$$N\binom{x}{k} = \binom{x+1}{k} = \binom{x}{k} + \binom{x}{k-1}.$$

Hence, the binomial-coefficient basis is compatible the with shift operator since we can choose $A = 1$, $B = 0$ *and* $\alpha_{k,-1} = 1$, $\alpha_{k,0} = 1$.

(c) *It is obvious that the binomial-coefficient basis is not compatible with the q-shift operator.*

(d) *Also binomial-coefficient basis is compatible with multiplication by* X *operator.(A more general result is true, see the next example).*

3. *This time, we will not specify our basis. Because we will show that multiplication by* x *operator works nicely with any factorial basis! Let* β *be our factorial basis. Then,choosing* $A = 0$ *and* $B = 1$ *satisfies conditions of Theorem 8, i.e* β *is compatible with* X*. Note that*

   (a) $degLP_k = degXP_k = degP_k + 1 = k+1$

   (b) P_k *divides* $LP_k = XP_k$*. Thus we are done.*

Moral of the Story: Throughout our discussion of other algorithms, we have always considered the shift operator. Therefore, it makes sense to find a basis which is compatible with the shift operator. We did find it! Thus, we will continue with binomial-coefficient basis from now on.

Let's recall our aim, we want to find a sum of the form $f(n) = \sum\limits_{k=0}^{\infty}\prod\limits_{i=1}^{m}\binom{a_in+b_i}{k}h_k$ such that $Lf = 0$. Thus, it is a good idea to extend our operator L such that L is an operator on $\mathbb{K}[[X]]$. This can be done as follows:

$$L\sum_{k=0}^{\infty} c_kP_k := \sum_{k=0}^{\infty} c_kLP_k. \tag{7.6}$$

Now assume that our factorial basis β is compatible with the operator L. Then, equation (7.6) becomes:

$$\begin{aligned} L\sum_{k=0}^{\infty} c_kP_k = \sum_{k=0}^{\infty} c_kLP_k = \sum_{k=0}^{\infty} c_k \sum_{i=-A}^{B} \alpha_{k,i}P_{k+i} \\ = \sum_{k=0}^{\infty}(\sum_{i=-B}^{A} \alpha_{k+i,-i}c_{k+i})P_k \end{aligned}$$

Thus, clearly we have the following proposition:

Theorem 9. *A formal power series* $y = \sum_{k=0}^{\infty} c_k P_k$ *satisfies* $Ly = 0$ *if and only if its coefficient sequence satisfies the recurrence*

$$\sum_{i=-B}^{A} \alpha_{k+i,-i} c_{k+i} = 0.$$

Moral of the Story: For each operator L we have corresponding recurrence operator on coefficient sequence! Thus for each L defined on $\mathbb{K}[[X]]$, we have a *corresponding operator* $L^* := \sum_{i=-B}^{A} \alpha_{n+i,-i} E_n^i$ where E_n is the shift operator with respect to n.

We already defined some operators on $\mathbb{K}[[X]]$. Let's look at their corresponding operators:

Example 41. *Let* $y = \sum_{k=0}^{\infty} c_k P_k(n)$.

1. *Let* $P_k(n) = \binom{n}{k}$. *Let's look at the shift operator:*
   (a) *Suppose* $L = N$ *and* $Ly = 0$. *Then,*

$$Ly = 0 \to \{N\} \sum_{k=0}^{\infty} c_k \binom{n}{k} = 0 \to \sum_{k=0}^{\infty} c_k \{N\} \binom{n}{k} = 0.$$

We know that N *is* $(1,0)$*-compatible with the binomial-coefficient basis. Thus, the above equation becomes*

$$\sum_{k=0}^{\infty} c_k \Big(\sum_{i=-1}^{0} \alpha_{k,i} P_{k-i} \Big). \tag{7.7}$$

(7.7) *is equivalent to*

$$\sum_{k=0}^{\infty} \Big(\sum_{i=0}^{1} \alpha_{k+i,-i} c_{k+i} \Big) P_k.$$

Lastly, since both $\alpha_{k,-1}$ *and* $\alpha_{k,0}$ *are* 1 *we have:*

$$\sum_{k=0}^{\infty} (c_k + c_{k+1}) P_k = 0.$$

In other words, $(E_k + 1)c_k = 0$. *Hence, we found that the corresponding operator for* N *is* $(E_k + 1)$.

   (b) *With a similar spirit, let us find the corresponding operator for* n*, i.e* *multiplication by n, as well. In other words, $L = n$ and $Ly = 0$. This means that*

$$Ly = 0 \to \{n\}\sum_{k=0}^{\infty} c_k \binom{n}{k} = 0 \to \sum_{k=0}^{\infty} c_k \{n\}\binom{n}{k}. \tag{7.8}$$

We know that multiplication by n is $(1,0)$-compatible with each factorial basis. Thus, it is $(1,0)$-compatible with binomial-coefficient basis as well. This means that we have

$$\sum_{k=0}^{\infty} c_k (\sum_{i=0}^{1} \alpha_{k,i} P_{k+i}) = 0.$$

As a result, we get

$$\sum_{k=0}^{\infty} (\sum_{i=-1}^{0} \alpha_{k+i,-i} c_{k+i}) P_k = 0.$$

We know that $\alpha_{k,0} = k$ and $\alpha_{k,1} = k+1$. Hence, the final equation is

$$\sum_{k=0}^{\infty} (kc_{k-1} + kc_k) P_k = 0.$$

As a result, the corresponding linear operator of n is $k(E_k^{-1}+1)$.

Now, we can try to solve the problem for an even smaller class of **Small Inverse Zeilberger Problem**, i.e we are considering the following problem:

Even Smaller Inverse Zeilberger Problem: Given a linear recurrence operator L of the form (2.4) find h_k such that $Lf(n) = 0$ where $f(n) = \sum_k \binom{n}{k} h_k$. In other words, in **Small Inverse Zeilberger Problem**, choose $m = 1$, $a_1 = 1$ and $b_1 = 0$.

Example 42. *Let $L = (n+2)N^2 - (3n+4)N + (2n+2)$. We want to find a sum of the form $\sum_k \binom{n}{k} h_k$ such that L annihilates this sum.*

Solution:

1. Firstly, we replace each N with $E_k + 1$ and each n with $k(E_k^{-1}+1)$. Thus, we have

$$L^{'} = (k+2)E_k^2 - k$$

2. Finding h_k such that $L^{'} h_k = 0$ gives us $h_k = \frac{1}{k}$ or in general $h_k = \frac{1}{ck}$ for a constant k. As a result, $L\sum_k \frac{\binom{n}{k}}{k} = 0$.
3. As a quick check, giving this sum to the Zeilberger's algorithm gives L back!

Remark 41. *Note that if a linear recurrence operator L of the form (2.4) annihilates $f(n)$, i.e. $Lf(n)=0$, then $L(cf(n))=0$ for all constant c, as well. We usually take this constant $c=1$ as in the previous example.*

Let's do another example:

Example 43. *Let $L=(n+4)N^2+(-3n-8)N+(2n+4)$. We want to find a sum of the form $f(n)=\sum\limits_k \binom{n}{k}h_k$ such that $Lf(n)=0$.*
Solution:

1. *Again, we replace each N with E_k+1 and each n with $k(E_k^{-1}+1)$. Then, we have*

$$\begin{aligned}L^{'} &= (k(E_k^{-1}+1)+4)(E_k+1)^2\\ &+(-3(k(E_k^{-1}+1)-8)(E_k+1)+2(k(E_k^{-1}+1)+4\\ &=(kE_k^{-1}+4+k)(E_k^2+2E_k+1)-\\ &(3kE_k^{-1}+3k+8)(E_k+1)+2kE_k^{-1}+2k+4\\ &=(k+4)E_k^2+(3k+8)E_k+\\ &3k+4+kE_k^{-1}+(-3k-8)E_k-6k-8\\ &-3kE_k^{-1}+2kE_k^{-1}+2k+4\\ &=(k+4)E_k^2-k\end{aligned}$$

This means that,

$$(k+4)h_{k+2}-kh_k=0. \tag{7.9}$$

2. *Now, we can use algorithm hyper to find hypergeometric solutions of (7.9). Algorithm hyper gives us,*

$$h_k=\frac{1}{k(k+2)}.$$

Thus, $f(n)=\sum\limits_k \frac{\binom{n}{k}}{k^2+2k}$.

3. *Surely, Zeilberger's algorithm gives L back when we execute it with the summand $\frac{\binom{n}{k}}{k^2+2k}$!*

Example 44. *Let $L=N^3-(n+3)^3N^2+(2n+5)(n+3)(n+2)N-(n+1)(n+2)(n+3)$. Find a sum of the form $f(n)=\sum\limits_k \binom{n}{k}h_k$ such that $Lf(n)=0$.*

Solution:

1. Let's find $L^{'}$ using the replacements N with E_k+1 and n with $k(E_k^{-1}+1)$.

Thus, we find

$$\begin{aligned} L^{'} = E_k^3 + (-k^3 - 9k^2 - 27k - 24)E_k^2 + \\ (-3k^3 - 18k^2 - 36k - 21)E_k + \\ (-3k^3 - 9k^2 - 9k - 2) + (-k^3)E_k^{-1}. \end{aligned}$$

In other words, we have the following equation

$$\begin{aligned} h_{k+3} + (-k^3 - 9k^2 - 27k - 24)h_{k+2} + (-3k^3 - 18k^2 - 36k - 21)h_k + \\ (-3k^3 - 9k^2 - 9k - 2)h_k - k^3 h_{k-1} = 0. \end{aligned}$$

2. By algorithm hyper, $h_k = k!^3$. Moreover, Zeilberger algorithm returns L when we apply it on the summand $\binom{n}{k}k!^3$. Hence, we are done!

Note that, the last exercise is very hard to do by hand. Unfortunately, this would be true for almost all the examples in this section. Thus, we will use Maple in certain points of the calculations.

Example 45. *We evaluate $\sum\limits_k \binom{n}{k}$ using Sister Celine's algorithm and Zeilberger's algorithm. Now it is time to do the reverse! In other words, given $L = N - 2$ find h_k such that L annihilates $y = \sum\limits_{k=0}^{\infty} \binom{n}{k} h_k$.*

1. *Again, let's apply the transformation $N \to E_k + 1$. Thus, $L^{'} = E_k + 1 - 2 = E_k - 1$.*

2. *This means that $Ly = 0$ if and only if $L^{'}h_k = 0$. Equivalently, $\{E_k - 1\}h_k = 0$. As a result, $h_{k+1} - h_k = 0$, i.e h_k is a constant!*

3. *Thus, $y = \sum_{k=0}^{\infty} \binom{n}{k} c$ where c is any constant. As usual, we take $c = 1$ and get our beloved sum!*

From the examples, it is clear that we always use the transformation $N \to E_k + 1$ and $n \to k(E_k^{-1} + 1)$. However it is not clear why we are doing this. Let's clarify the transformation:

1. Firstly, we have a linear recurrence operator $L = \sum_{i=0}^{r} a_i(n)N^i$. Thus, L consists of n and N. To be more precise, L is an element of $\mathbb{K}[n, N]$. As a result, if we can transform these two, then we actually transformed L! Thus, it is clear that we just need to know the effect of transformation on N and n.

2. Secondly, from Example 41 we know that the corresponding operator for N is $E_k + 1$ and n is $k(E_k^{-1} + 1)$.

Thus, we kind of find a meta-algorithm to solve **Small Inverse Zeilberger Problem** which is the following:

1. Given a linear recurrence operator L, find the corresponding linear operator $L^{'}$.

2. Find h_k such that $L^{'}$ annihilates h_k. In other words, $L^{'}h_k = 0$.

By algorithm hyper we can solve the second step. However, for the first step it depends on the L and the factorial basis as well! Right now, we can solve the special case of the problem which is choosing binomial-coefficient basis as a factorial basis and $L = \sum_{i=0}^{r} a_i(n)N^i$, i.e choosing linear operator as the shift operator.

Remark 42. *It is a good opportunity to say something about to correctness of an algorithm. In some cases, it is not easy to understand whether our algorithm is correct or not. Especially by hand. Luckily for us, we can use Zeilberger's algorithm to check whether our answer is true or not! To be more precise, suppose we are given a linear operator of the form* $L = \sum_{i=0}^{r} p_i(n)N^i$*, then we find a sum* $f(n) = \sum_k F(n,k)$*, i.e we claim that* $Lf(n) = 0$*. We can easily check the correctness of the algorithm using Zeilberger's algorithm. In other words, if Zeilberger's algorithm gives L back when it takes* $F(n,k)$ *as an input, then our algorithm is certainly correct. However, the reverse implication is not true. In other words, maybe our algorithm takes linear opearator L and gives a sum* $\sum\limits_k F(n,k)$ *as its output. It may be the case that our algorithm is true and Zeilberger's algorithm does not give us the recurrence started with.*

Let's consider following three problems:

1. Given a linear operator L, can we find a sum of the form $f(n) = \sum_k \binom{n}{k} h_k$ such that $Lf = 0$?

2. Given a linear operator L, can we find a sum of the form $f(n) = \sum_k \binom{an+b}{k} h_k$ such that $Lf = 0$?

3. Given a linear operator L, can we find a sum of the form $f(n) = \sum_k \prod_{i=1}^{m} \binom{a_i n+b_i}{k} h_k$ such that $Lf = 0$?

Right now, we can solve the first problem. Now we try to solve the second problem. To solve it, we need to extend our definition of binomial-coefficient basis a bit:

Definition 17. *Let* a *be a natural number and* b *be a real number. Then,* $\{\binom{ax+b}{k}\}_{k=0}^{\infty}$ *is called a* ***generalized binomial-coefficient basis*** *of* $\mathbb{K}[X]$*. We denote* $\binom{ax+b}{k}$ *by* $P_k^{a,b}$*. Also, we denote this factorial basis by* $C_{a,b}$*. In other words,* $C_{a,b} = \langle P_k(x) = \binom{ax+b}{k} \rangle_{k=0}^{\infty}$.

Example 46. *1. If we choose $a = 3$ and $b = 2$, then, $C_{3,2} =< P_k(x) = \binom{3x+2}{k} >_{k=0}^{\infty}$.*

2. Choosing $a = 6$ and $b = 5$, then, $C_{6,5} =< P_k(x) = \binom{6x+5}{k} >_{k=0}^{\infty}$.

One point needs an explanation, obviously: OK, we extended the definition of the binomial-coefficient basis. However, we need to make sure that the followings are true:

1. Generalized-binomial basis is really a factorial basis.

2. Generalized-binomial basis is working nicely with the shift operator N, i.e., it is compatible with N.

Fortunately, for us these are indeed true as the following theorem shows:

Theorem 10. *Any generalized binomial-coefficient basis $C_{a,b}$ is a factorial basis of $\mathbb{K}[X]$, which is compatible with the shift operator N.*

Proof:

1. First, we will show that generalized binomial-coefficient basis is really a factorial basis. Observe that:

   (a) $degP_k^{a,b} = deg\binom{ax+b}{k} = k$ is obvious.

   (b) Also $P_{k+1}^{a,b} = \frac{ax+b-k}{k+1} P_k^{a,b}$. In other words, P_k divides P_{k+1}. Thus, it is a factorial basis.

2. Second, we need to show that $C_{a,b}$ is compatible with the shift operator N. In other words, we need to find natural numbers A and B such that

$$P_k^{a,b} = \sum_{i=-B}^{A} \alpha_{k,i} P_{k+i}^{a,b}.$$

   The following chain of equations show that we are done:

$$NP_k^{a,b}(x) = P_k^{a,b}(x+1) = \binom{ax+a+b}{k} = \sum_{i=0}^{a} \binom{a}{i}\binom{ax+b}{k-i}$$
$$= \sum_{i=-a}^{0} \binom{a}{-i}\binom{ax+b}{k+i} = \sum_{i=-a}^{0} \binom{a}{-i} P_{k+i}^{a,b}(x).$$

   More precisely, choose $A = 0$, $B = a$ and $\alpha_{k,i} = \binom{a}{-i}$ for $i = -a, -a+1, ..0$ in the Definition 16.

Generally, it is good to have some methods to construct new objects using old objects of the same type. For example,

1. Given two groups G and H we can form a new group using direct product of G and H.

2. Given a vector space V and subspaces U and W, we can form another subspace using direct sum of U and W.

We want to do the same thing here, as well. In other words, we want to create a new factorial basis using an old factorial basis:

Definition 18. *Suppose we are given m factorial basis, say $\beta_i =< P_k^{(i)}(x) >_{k=0}^{\infty}$ be a basis of $\mathbb{K}[X]$ for $i = 1,2,3,...,m$. Let*

$$P_{mk+j}^{(\pi)}(x) := \prod_{i=1}^{j} P_{k+1}^{(i)}(x) \cdot \prod_{i=j+1}^{m} P_k^{(i)}(x).$$

Then the sequence $\prod_{i=1}^{m} \beta_i :=< P_n^{(\pi)}(x) >_{n=0}^{\infty}$ is the product of β_1, β_2,...,β_m.

Again, it is not clear whether our construction gives us a factorial basis or not. Also it is not clear whether this factorial basis will behave nicely with other linear operators. As the following theorem shows, everything works nicely:

Theorem 11. *Let β_1, β_2,...,β_m be factorial bases of $\mathbb{K}[X]$, and L be a linear operator of $\mathbb{K}[X]$. Then,*

1. *$\beta = \prod_{i=1}^{m} \beta_i$ is a factorial basis of $\mathbb{K}[X]$.*

2. *Suppose each β_i is (A_i, B_i)-compatible with L. Write $A = \max_{1 \le i \le m} A_i$ and $B = \min_{1 \le i \le m} B_i$. Then, $\prod_{i=1}^{m} \beta_i$ is (mA, B)-compatible with L.*

Proof:

1. First, we need to show that β is a factorial basis of $\mathbb{K}[X]$. In other words, we need to show that two conditions of Definition 15 are satisfied:

   (a) First condition is easy, since

   $$deg P_{mk+j}^{(\pi)} = j(k+1) + (m-j)k = mk + j.$$

   (b) For the second condition, let $n = mk + j$. We have two cases to consider:

      i. Suppose $j = m-1$. Then, $n = mk + (m-1)$, i.e., $n+1 = m(k+1) + 0$.

Thus,

$$\frac{P_{n+1}^{(\pi)}}{P_n^{(\pi)}} = \frac{1}{\prod_{i=1}^{m-1} P_{k+1}^{(i)}} \cdot \frac{\prod_{i=1}^{m} P_{k+1}^{(i)}}{P_k^{(m)}} = \frac{P_{k+1}^{(m)}}{P_k^{(m)}}.$$

ii. Suppose $0 \leq j \leq m-2$. Then, $n+1 = mk+(j+1)$. Thus,

$$\frac{P_{n+1}^{(\pi)}}{P_n^{(\pi)}} = \frac{\prod_{i=1}^{j+1} P_{k+1}^{(i)}}{\prod_{i=1}^{j} P_{k+1}^{(i)}} \cdot \frac{\prod_{i=j+2}^{m} P_k^{(i)}}{\prod_{i=j+1}^{m} P_k^{(i)}} = \frac{P_{k+1}^{(j+1)}}{P_k^{(j+1)}}.$$

Thus, in both cases we see that $P_n^{(\pi)}$ divides $P_{n+1}^{(\pi)}$. As a result, β is a factorial basis of $\mathbb{K}[X]$.

2. Now, we need to show that if L behaves nicely with all other factorial bases β_i, then L works nicely with β, as well. More precisely, we need to show that β is compatible with L. Using Theorem 8, it suffices to show that β satisfies $(C1)$ and $(C2)$. Let p be an arbitrary polynomial and $p = \sum_{k=0}^{degp} c_k^{(i)} P_k^{(i)}$ be the expansion of p with respect to the basis β_i for $i = 1,2,...,m$. Then, $Lp = \sum_{k=0}^{degp} c_k^{(i)} LP_k^{(i)}$.

   (a) Let's start with $C1$. We need to show that $deg(Lp) \leq deg(p)+B$. Fix i between 1 and m. Observe that

   $$deg(Lp) \leq \max_{0\leq k\leq m} deg(LP_k^{(i))} \leq \max_{0\leq k\leq degp} k+B_i = deg(p)+B_i. \tag{7.10}$$

   Thus, (7.10) holds for all i. This implies that $deg(Lp) \leq deg(p)+B$. If we take $p = P_k^{(\pi)}$, we have $deg(LP_k) \leq k+B$. Hence, $(C1)$ is satisfied.

   (b) Let's check $(C2)$. We need show that $P_{n-mA}^{(\pi)}$ divides $LP_n^{(\pi)}$ for all $n \geq mA$. Firstly, note that

   i. $P_{k+1-A}^{(i)}$ divides $LP_{k+1}^{(i)}$ for all $k \geq A$ since β_i is compatible with L.

   ii. Similarly. $P_{k-A}^{(i)}$ divides $LP_k^{(i)}$ for all $k \geq A$ since β_i is compatible with L.

   Then, observe that

   $$P_{m(k-A)+j}^{(\pi)} = \prod_{i=1}^{j} P_{k+1-A}^{(i)} \cdot \prod_{i=j+1}^{m} P_{k-A}^{(\pi)}. \tag{7.11}$$

   By our observations, the right hand side of (7.11) divides $\prod i = 1^j LP_{k+1}^{(i)} \cdot \prod_{i=j+1}^{m} LP_k^{(i)}$. Thus, the left hand side of (7.11) divides $\prod i = 1^j LP_{k+1}^{(i)} \cdot$

$\prod_{i=j+1}^{m} LP_k^{(i)}$ as well. Hence, (C2) satisfied. As a result we are done, i.e $\prod_{i=1}^{m} \beta_i$ is (mA,B)-compatible with L.

Intuitively, we showed that if you have a linear operator, say L, and some factorial bases which work nicely with L, i.e. they are compatible with L, then their product also works nicely with L. Now we can define the product of generalized binomial-coefficient basis.

Definition 19. *Let m be a positive natural number. Let $\vec{a} = (a_1, a_2, ..., a_m)$ and $\vec{b} = (b_1, b_2, ..., b_m)$ where a_i's are non-zero natural numbers and b_k's are real numbers. Then, we denote the product of generalized binomial coefficient bases by $C_{\vec{a},\vec{b}} = \prod_{i=1}^{m} C_{a_i,b_i}$ and call it **product binomial-coefficient basis** of $\mathbb{K}[X]$.*

Thus, at the end of the day we proceed as follows:

We know that the binomial-coefficient basis works nicely with the shift operator N. We use this fact to show that the generalized binomial-coefficient basis works nicely with the shift operator. Then, by Theorem10 we know that product binomial-coefficient basis is working nicely with N. Now, we want to use this new factorial basis to solve Small Inverse Zeilberger Problem.

Example 47. *Let's look at some examples of product factorial bases:*

1. *Let m=2, $\vec{a} = (2,3)$ and $\vec{b} = (-1,2)$. In other words, $\beta_1 =< \binom{2n-1}{k} >_{k=0}^{\infty}$ and $\beta_2 =< \binom{3n+2}{k} >_{k=0}^{\infty}$. Thus, $\beta =< P_r^{(\pi)} >_{r=0}^{\infty}$ where*

$$P_r^{(\pi)} = P_{2k+j} = \prod_{i=1}^{j} P_{k+1}^{(i)} \cdot \prod_{i=j+1}^{2} P_k^{(i)},$$

$$P_{2k} = \prod_{i=1}^{0} P_{k+1}^{(i)} \cdot \prod_{i=1}^{2} P_k^{(i)} = \binom{2n-1}{k}\binom{3n+2}{k},$$

$$P_{2k+1} = \prod_{i=1}^{1} P_{k+1}^{(i)} \cdot \prod_{i=2}^{2} P_k^{(i)} = \binom{2n-1}{k+1}\binom{3n+2}{k}.$$

2. *Let $\beta_1 =< x^k >_{k=0}^{\infty}$, $\beta_2 =< \binom{n}{k} >_{k=0}^{\infty}$. Then, $\beta =< P_r^{(\pi)} >_{r=0}^{\infty}$ where*

$$P_r^{(\pi)} = P_{2k+j} = \prod_{i=1}^{j} P_{k+1}^{(i)} \cdot \prod_{i=j+1}^{2} P_k^{(i)}.$$

$$P_{2k} = \prod_{i=1}^{0} P_{k+1}^{(i)} \cdot \prod_{i=1}^{2} P_k^{(i)} = x^k \binom{n}{k.}$$

$$P_{2k+1} = \prod_{i=1}^{1} P_{k+1}^{(i)} \cdot \prod_{i=2}^{2} P_k^{(i)} = x^{k+1} \binom{n}{k}.$$

The last example shows that we can pick different kinds (standart basis and generalized-binomial-coefficient basis) of factorial bases and take their product to generate a product factorial basis! However, there is a problem. If we want to find a linear operator which is working nicely with the product basis, then we need to find an operator which works nicely with both power basis and binomial-coefficient basis. In any case, from now on we will pick a generalized binomial-coefficient factorial basis and form a product factorial basis using them.

Example 48. *Certain examples of binomial-coefficient bases expansions:*

1. *Suppose we are given* $\beta = C_{(2,1),(3,2)}$. *We want to find the following extensions:*

   (a) *Write* $P_{2k}(x+1)$ *as a linear combination of basis elements.*

   (b) *Write* $P_{2k+1}(x+1)$ *as a linear combination of basis elements.*

   Let us fix our input in a clear manner. We have $m = 2$, $a_1 = 2$, $a_2 = 1$, $b_1 = 3$, $b_2 = 2$. *Thus,* $A = 2$ *and* $B = 2$. *Note that due to Theorem 11 we know that* β *is* $(4,2)$*-compatible with the shift operator* N. *This means that we can write* $P_{2k}(x+1)$ *as a linear combination of* $P_{2k+2}(x)$, $P_{2k+1}(x)$, $P_{2k}(x)$, $P_{2k-1}(x)$ *and* $P_{2k-2}(x)$. *Also,* $P_{2k+1}(x+1)$ *as a linear combination of* $P_{2k+3}(x)$, $P_{2k+2}(x)$, $P_{2k+1}(x)$, $P_{2k}(x)$ *and* $P_{2k-1}(x)$. *Using the method of undetermined coefficients we can find coefficients of basis elements in the expansion:*

$$P_{2k}(x+1) = P_{2k+2}(x) + 2P_{2k+1}(x) + \frac{3(k+1)}{2k}P_{2k}(x) + \frac{k+1}{4k}P_{2k-1}(x) + \frac{-(k^2-2k-3)}{4k(k-1)}P_{2k-2}(x).$$

   Similarly,

$$P_{2k+1}(x+1) = P_{2k+3}(x) + \frac{2(2k+1)}{k+1}P_{2k+2}(x) + \frac{7k-1}{2(k+1)}P_{2k+1}(x) + \frac{k+2}{2k}P_{2k}(x).$$

2. *Suppose we are given* $\beta = C_{(1,1),(0,1)}$. *We want to find the following extensions:*

   (a) *Write* $P_{2k}(x+1)$ *as a linear combination of basis elements.*

   (b) *Write* $P_{2k+1}(x+1)$ *as a linear combination of basis elements.*

   As we can see, we have $m = 2$, $a_1 = 1$, $a_2 = 1$, $b_1 = 0$ *and* $b_2 = 1$. *Thus,* $A = 1$ *and* $B = 0$. *In the same spirit, we can write* $P_{2k}(x+1)$ *as a linear combination of* $P_{2k}(x)$, $P_{2k-1}(x)$ *and* $P_{2k-2}(x)$. *Similarly,* $P_{2k+1}(x+1)$ *can be written as* *a linear combination of $P_{2k+1}(x)$, $P_{2k}(x)$ and $P_{2k-1}(x)$. Again, it is easy to find that*

$$P_{2k}(x+1) = P_{2k}(x) + 2P_{2k-1}(x) + \frac{k+1}{k}P_{2k-2}(x).$$

Also,

$$P_{2k+1}(x+1) = P_{2k+1}(x) + \frac{2k+1}{k+1}P_{2k}(x) + \frac{k-1}{k+1}P_{2k-1}(x).$$

We know that multiplication by x operator, i.e X, works nicely with all factorial bases. Thus, it must be the case that it works nicely with $C_{a,b}$ as well, as following theorem shows:

Theorem 12. *Every factorial basis $C_{a,b}$ is $(0,1)$-compatible with X.*
Proof:
Observe that

$$XP^{(\pi)}_{mk+j}(x) = \frac{k+1}{a_{j+1}}P^{(\pi)}_{mk+j+1}(x) + \frac{k-b_{j+1}}{a_{j+1}}P^{(\pi)}_{mk+j}.$$

Thus, we are done!

Now we can go back to our original problem and try to solve the problem.

Example 49. *Suppose we are given $L = (2n+1)N - 8n - 12$, $m = 1$, $a_1 = 2$ and $b_1 = 1$. We want to find h_k such that L annihilates $\sum\limits_k \binom{2n+1}{k} h_k$.*

Solution:

1. First, let A be the maximum of $a_i's$. Since, we have only a_1, $A = a_1 = 2$.

2. Second, we need to compute

$$P_{mk+j}(n) = \prod_{i=1}^{j} \binom{a_i n + b_i}{k+1} \prod_{i=j+1}^{m} \binom{a_i n + b_i}{k}.$$

   Thus, we have

$$P_k(n) = \binom{2n+1}{k}.$$

3. Now we need to compute $\alpha_{k,j,i}$ such that

$$P_{mk+j}(n+1) = \sum_{i=-mA} \alpha_{k,j,i} P_{mk+j+i}(n).$$

In other words, we need to solve

$$\binom{2n+3}{k} = \sum_{i=-2}^{0} \alpha_{k,0,i} P_{k+i}(n).$$

This gives 1,2,1 as coefficients.

4. Let's compute $E_{r,j}$ and $X_{r,j}$ for all $0 \le r, j \le m-1$:

$$\begin{aligned} E_{0,0} := \sum_{\substack{-2\le i\le 0 \\ i+0=0(mod1)}} \alpha_{k-i,0,i} E_k^{-i} \\ = \alpha_{k-2,0,-2} E_k^2 + \alpha_{k-1,0,-1} E_k^1 + \alpha_{k,0,0} E_k^0 \\ = E_k^2 + 2E_k^1 + 1. \end{aligned}$$

Similarly,

$$X_{0,0} := \frac{k-1}{2} + \frac{k}{2} E_k^{-1}.$$

5. Now, let's apply the substitution $N \to E_{0,0} = E_k^2 + 2E_k + 1$ and $n \to X_{0,0} = \frac{k-1+kE_k^{-1}}{2}$:

$$\begin{aligned} (2n+1)N - 8n - 12 \to (kE_k^{-1} + k)(E_k^2 + 2E_k + 1) = \\ kE_k + 2k + kE_k^{-1} + kE_k^2 + 2kE_k + k - 4kE_k^{-1} - 4k - 8 = \\ kE_k^2 + 3kE_k - k - 8 - 3kE_k^{-1}. \end{aligned}$$

6. Then, we know that $kE_k^2 + 3kE_k - k - 8 - 3kE_k^{-1}$ annihilates h_k. In other words, $kh_{k+2} + 3kh_{k+1} - (k+8)h_k - 3kh_{k-1} = 0$. Then, we can use algorithm hyper to find h_k. Then, we find that $h_k = k$.

7. Surely enough, giving $\binom{2n+1}{k} \cdot k$ back to Zeilberger's algorithm gives us $(2n+1)N - 8n - 12$ back!

Let's look at the algorithm in general:

Algorithm 5: Algorithm Hyper for order d recurrences

Input:

1. A linear recurrence operator with polynomial coefficient, i.e $L = \sum_{i=0}^{d} a_i(n)N^i$.
2. A natural number m.
3. A vector $\vec{a} = (a_1, a_2, ..., a_m)$ where each a_i is a natural number.
4. A vector $\vec{b} = (b_1, b_2, ..., b_m)$ where each b_i is an integer.

Output: A sum of the form $f(n) = \sum_{k=0}^{\infty} H(n,k)t_k$ where $H(n,k) = \prod_{i=1}^{m} \binom{a_i n + b_i}{k}$ such that $Lf(n) = 0$, i.e L annihilates the sum $f(n)$.

Steps:

1. Let $A := max_{1 \le i \le m} a_i$.
2. **For** $j = 0, 1..., m-1$ let
$$P_{mk+j}(x) := \prod_{i=1}^{j} \binom{a_i x + b_i}{k+1} \cdot \prod_{i=j+1}^{m} \binom{a_i x + b_i}{k}.$$
3. **For** $j = 0, 1..., m-1$ compute $\alpha_{k,j,i} \in \mathbb{K}(k)$ such that
$$P_{mk+j}(x+1) = \sum_{i=-mA}^{0} \alpha_{k,j,i} P_{mk+j+i}(x).$$
4. **For** $r, j = 0, 1, ..., m-1$ let
$$E_{r,j} := \sum_{\substack{-mA \le i \le 0 \\ i+j = r (mod m)}} \alpha_{k+\frac{r-i-j}{m},j,i} E_k^{\frac{r-i-j}{m}},$$
$$X_{r,j} := [r = j]\frac{k - b_{j+1}}{a_{j+1}} + [r = 0 \wedge j = m-1]\frac{k}{a_{j+1}} E_k^{-1} + [r = j+1]\frac{k+1}{a_{j+1}}.$$
Note that we are using “[ ] “ as the **Iverson bracket**.
5. Let $[R_\beta E] = [E_{r,j}]_{r,j=0}^{m-1}$ and $[R_\beta X] = [X_{r,j}]_{r,j=0}^{m-1}$.
6. Let $[R_\beta L] = [L_{r,j}]_{r,j=0}^{m-1}$ to be the matrix of operators obtained by applying the following substitution to L:
   (a) $E \to [R_\beta E]$,
   (b) $x \to [R_\beta X]$,
   (c) $1 \to I_m$.
7. **Return** $L' := gcrd(L_{0,0}, L_{1,0}, ..., L_{m-1,0})$ and stop. We mean the greatest common right divisor by “ gcrd ”.

8. Discussion

Let's look at possible ways to expand the results:

1. Throughout the thesis, we always consider hypergeometric functions. We can consider general class of functions.(Say, P-recursive functions, see Remark 2.) Then, we can look at the question: Evaluate $\sum_{k=0}^{n} F(n,k)$ where $F(n,k)$ is a P-recursive function.

2. In the last chapter, we looked at the inverse Zeilberger problem, and take the generalized binomial coefficient as $\binom{ax+b}{k}$. We can take an even more generalized binomial coefficient as $\binom{ax+b}{ck+d}$. To be more precise, one can consider the following problem: Given a linear recurrence operator L, a natural number m, integers $a_1,a_2,...,a_m$, $b_1,b_2,...,b_m$, $c_1,c_2,...,c_m$ and $d_1,d_2,...,d_m$, find h_k such that L is the annihilating operator for $f(n) := \sum_{k=0}^{\infty} \prod_{i=1}^{m} \binom{a_ix+b_i}{c_ik+d_i} h_k$, i.e. $Lf(n) = 0$.

3. We can take different linear operators in the last chapter as well. We considered the shift operator N. We can also take multiplication by x operator or the differentiation operator etc.

4. Algorithm Hyper checks whether a linear recurrence operator has hypergeometric solutions or not. Maybe it is possible to extend it to the non-linear recurrence operators.

We can use the following as a check list for our algorithms:

(a) Do we know the necessary and sufficient conditions for an algorithm to work? For example, we know that if $F(n,k)$ is a proper hypergeometric term then Zeilberger's algorithm will find a recurrence satisfied by $f(n) = \sum_k F(n,k)$. In other words, it is a sufficient condition for Zeilberger's algorithm to work. Is it of practical use? In other words, maybe we find a condition for our algorithm to work, however, it is not possible to check this condition!

(b) Is it of practical use, i.e. is it fast? In other words, what is the running time of our algorithm with respect to the size of our input?

(c) Is it complete? That means, whether the algorithm gives a definite answer for all instances of the problem. Let's give an easy example: Suppose we have an algorithm which checks whether a given integer is prime or not. Also suppose it only checks whether the given integer is divisible by first fifty primes or not. If it is divisible by any of those primes, obviously the given integer is not a prime and our algorithm detects it. However, if it is not divisible by any of those primes, then the algorithm is inconclusive. So, it is not complete.

(d) Can we check the correctness of the algorithm easily? See, the *proof certificate*.

(e) Can we extend the algorithm? In other words, suppose we have an algorithm which works for all polynomial inputs. Can we extend our algorithm such that it works for rational functions as well?

(f) Is it possible to find all solutions? In other words, suppose a problem has three different solutions, can our algorithm find all of them or just find one?

(g) Can we use some instances to create new instances of the algorithm easily? For example, suppose we have a factorization algorithm for polynomials and we find a factorization of f and g. Then, actually we find the factorization of fg just using our previous results!

(h) Can we guess some properties of the result beforehand? Suppose we have an algorithm which gives a polynomial as a output. Can we guess (or find!) the degree of it without actually finding the solution?

(i) Is there a continuous (discrete) analog of the algorithm? See our discussion about variations of Zeilberger's algorithm.

Now, we will look at whether the algorithms we discussed have the properties or not. Also, we will discuss some other important concepts as well.

1. Let's check whether Gosper's algorithm has these properties or not:

   (a) We know that we must start with a sum $f(n) = \sum_{k=0}^{n} t_k$ where t_k is a hypergeometric term. Thus, we know the necessary condition for Gosper's algorithm to work. However, in general, without applying Gosper's algorithm we cannot decide whether the sum $f(n)$ is Gosper summable or not.

   (b) Gosper's algorithm is fast. We will not discuss the details here. In other words, we will not give the time complexity of the algorithm. Since in this section our aim is not to give complete details of the algorithm.

(c) Gosper's algorithm is complete. In other words, suppose Gosper's algorithm concludes that a particular summand is not Gosper summable. Then, the summand is really not Gosper summable and vice versa!

(d) We can check the correctness of the Gosper's algorithm easily using the proof certificate, which is the hypergeometric function $d(n)$ in this case, see Definition 9.

(e) We extend Gosper's algorithm to answer the following question: Given a linear combination c_n of hypergeometric terms, how can we decide if the sum $s_n = \sum_{k=0}^{n} c_k$ is expressible as a linear combination of hypergeometric terms? (See Petkovšek et al. (1996) for more details)

(f) This question does not make sense for Gosper's algorithm. Since there is no concept of all solutions in Gosper's algorithm.

(g) We can sum similar Gosper summable summands to get another Gosper summable summand.

(h) This question does not make sense for Gosper's algorithm.

(i) There is a continious analog of our algorithm as well.

2. Consider Zeilberger's algorithm now:

(a) We know the sufficient condition for Zeilberger's algorithm to work, see Theorem 6. Also, Abramov found a necessary condition for Zeilberger's algorithm to work as well, see Abramov (2003).

(b) It is a pretty fast algorithm. Again we will not give the details. However, it is faster than Sister Celine's algorithm since in Sister Celine's algorithm we try to solve a much bigger linear system than Zeilberger's algorithm. For details, see Koepf (1998).

(c) Unfortunately, this is not correct. In other words, suppose Zeilberger's algorithm cannot find a recurrence of order 1 for a summand $F(n,k)$, we cannot conclude that $f(n) = \sum_k F(n,k)$ does not satisfy a recurrence of order 1. See Example 27.

(d) It is easy to check the correctness of the algorithm due to $G(n,k)$, see **Chapter5** for details.

(e) Chyzak extended Zeilberger's algorithm in a way that it works for certain classes of non-hypergeometric functions as well. See Chyzak (2000).

(f) This question does not make sense in this context since there is no concept of all solutions in Zeilberger's algorithm.

(g) Zeilberger's algorithm is closed under addition, obviously!

(h) Abrahom shows that we can find a non-trivial bound for the degree of recurrence constructed by Zeilberger's algorithm. See Abramov & Le (2005).

(i) There exist kind of a continuous analog of this algorithm. For details, see Koepf (1998).

3. Consider the Hyper's algorithm now:

(a) We need to have a linear recurrence operator.

(b) It is not fast. For details, see Petkovšek (1992).

(c) It is definite. In other words, if algorithm hyper concludes that there is no hypergeometric solution to a particular recurrence, there is no hypergeometric solution!

(d) We can easily check the correctness of the algorithm just by pluging the answer to the recurrence back.

(e) Abramov extend the algorithm to find D'Alembertian solutions to linear recurrences. See Abramov & Petkovšek (1994).

(f) It is possible to find all solutions. In other words, we can modify algorithm Hyper in a way that it gives us a basis for the solution space of the recurrence. For details, see Petkovšek et al. (1996).

(g) To the knowledge of the author it is not known.

(h) It does not make sense in our context.

(i) To the knowledge of the author it is not known.

BIBLIOGRAPHY

Abramov, S. A. (1995). Rational solutions of linear difference and q-difference equations with polynomial coefficients. In *Proceedings of the 1995 international symposium on Symbolic and algebraic computation*, (pp. 285–289).

Abramov, S. A. (2003). When does zeilberger's algorithm succeed? *Advances in Applied Mathematics*, *30*(3), 424–441.

Abramov, S. A., Bronstein, M., & Petkovšek, M. (1995). On polynomial solutions of linear operator equations. In *Proceedings of the 1995 international symposium on Symbolic and algebraic computation*, (pp. 290–296).

Abramov, S. A. & Le, H. Q. (2005). On the order of the recurrence produced by the method of creative telescoping. *Discrete mathematics*, *298*(1-3), 2–17.

Abramov, S. A., Le, H. Q., & Petkovšek, M. (2003). Rational canonical forms and efficient representations of hypergeometric terms. In *Proceedings of the 2003 international symposium on Symbolic and algebraic computation*, (pp. 7–14).

Abramov, S. A. & Petkovšek, M. (1994). D'alembertian solutions of linear differential and difference equations. In *Proceedings of the international symposium on Symbolic and algebraic computation*, (pp. 169–174).

Apéry, R. (1979). Irrationalité de ζ (2) et ζ (3). *Astérisque*, *61*(11-13), 1.

Beckwith, D. (2008). 11343. *The American Mathematical Monthly*, *115*(2), 166–166.

Beckwith, D., Kwong, H., Pratt, R., & Singer, N. (2008). A vanishing alternating sum: 11212/11220. *The American Mathematical Monthly*, *115*(4), 366–366.

Chyzak, F. (2000). An extension of zeilberger's fast algorithm to general holonomic functions. *Discrete Mathematics*, *217*(1-3), 115–134.

Fasenmyer, S. M. C. (1949). A note on pure recurrence relations. *The American Mathematical Monthly*, *56*(1P1), 14–17.

Kedlaya, K. S., Poonen, B., & Vakil, R. (2020). *The William Lowell Putnam Mathematical Competition 1985–2000: Problems, Solutions, and Commentary*, volume 33. American Mathematical Soc.

Koepf, W. (1998). Hypergeometric summation. *Vieweg, Braunschweig/Wiesbaden*, *5*, 6.

Malm, D. & Subramaniam, T. (1995). The summation of rational functions by an extended gosper algorithm. *Journal of symbolic computation*, *19*(4), 293–304.

Matiyasevich, Y. V. (1993). Hilbert's tenth problem. foundations of computing series.

Petkovšek, M. (1992). Hypergeometric solutions of linear recurrences with polynomial coefficients. *Journal of symbolic computation*, *14*(2-3), 243–264.

Petkovšek, M., Wilf, H. S., & Zeilberger, D. (1996). A= b, ak peters ltd. *Wellesley, MA*, *30*.

Poghosyan, M. (2008). 11356. *The American Mathematical Monthly*, *115*(4), 365–365.

Richardson, D. (1966). Some unsolvable problems involving functions of a real variable. *Notices, Amer. Math. Soc*, *13*, 135.

Wilf, H. S. & Zeilberger, D. (1992). An algorithmic proof theory for hypergeometric (ordinary and "q") multisum/integral identities. *Inventiones mathematicae*, *108*(1), 575–633.