Article Submitted to European Journal of Combinatorics

MacMahon's Partition Analysis: The Omega Package

GEORGE E. ANDREWS*[1], PETER PAULE[2] AND AXEL RIESE†[3]

[1] *Department of Mathematics*
The Pennsylvania State University
University Park, PA 16802, USA
E-mail: andrews@math.psu.edu

[2] *Research Institute for Symbolic Computation*
Johannes Kepler University Linz
A–4040 Linz, Austria
E-mail: Peter.Paule@risc.uni-linz.ac.at

[3] *Research Institute for Symbolic Computation*
Johannes Kepler University Linz
A–4040 Linz, Austria
E-mail: Axel.Riese@risc.uni-linz.ac.at

Abstract

In his famous book "Combinatory Analysis" MacMahon introduced Partition Analysis ("Omega Calculus") as a computational method for solving problems in connection with linear homogeneous diophantine inequalities and equations. The object of this paper is to show that partition analysis is ideally suited for being implemented in computer algebra. To this end we have developed the computer algebra package `Omega`. In addition to an introduction to basic facts of "Omega Calculus", we present a number of applications that illustrate the usage of the package.

1. Introduction

We begin with the beautiful refinement of Euler's classic result [1, p. 5] that was discovered by M. Bousquet-Mélou and K. Eriksson [5] only recently:

*Partially supported by the visiting researcher program of the J. Kepler University Linz and by the U.S. National Science Foundation.

†Supported by SFB-grant F1305 of the Austrian FWF.

Theorem 1.1 ("Lecture Hall Partition Theorem"): *The number of partitions of n of the form $n = b_j + b_{j-1} + \cdots + b_1$ wherein*

$$\frac{b_j}{j} \geq \frac{b_{j-1}}{j-1} \geq \cdots \geq b_1 \geq 0$$

and

$$b_j - b_{j-1} + \cdots + (-1)^{j-1} b_1 = m$$

equals the number of partitions of n into exactly m odd parts each of which is less than or equal to $2j-1$.

In [5] Bousquet-Mélou and Eriksson gave two different proofs of this theorem, one using Bott's formula for the affine Coxeter group $\tilde{C}_n$, and one of bijective-combinatorial nature. In [2] the first named author presented a proof following an entirely different approach, MacMahon's Partition Analysis [7, Vol. II, Sect. VIII, pp. 91–170]. In order to illustrate this point, we recall the definition of MacMahon's Omega operator $\Omega_{\geqq}$.

Definition 1.1: The operator $\Omega_{\geqq}$ is defined on functions with absolutely convergent multisum expansions

$$\sum_{s_1=-\infty}^{\infty} \cdots \sum_{s_r=-\infty}^{\infty} A_{s_1,\ldots,s_r} \lambda_1^{s_1} \cdots \lambda_r^{s_r}$$

in an open neighborhood of the complex circles $|\lambda_i| = 1$. The action of $\Omega_{\geqq}$ is given by

$$\underset{\geqq}{\Omega} \sum_{s_1=-\infty}^{\infty} \cdots \sum_{s_r=-\infty}^{\infty} A_{s_1,\ldots,s_r} \lambda_1^{s_1} \cdots \lambda_r^{s_r} := \sum_{s_1=0}^{\infty} \cdots \sum_{s_r=0}^{\infty} A_{s_1,\ldots,s_r}.$$

Note: In applications, the $A_{s_1,\ldots,s_r}$ themselves will be rational functions of several complex variables over $\mathbb{C}$. In each instance it is straightforward to specify the domain for these variables to guarantee the absolute convergence required in Definition 1.1.

While MacMahon did not carefully distinguish whether his Laurent series were analytic or merely formal, we emphasize that it is essential to treat everything analytically rather than formally because the method relies on unique Laurent series representations of rational functions. For instance, if we were to proceed formally, then

$$\underset{\geqq}{\Omega} \sum_{n=0}^{\infty} q^n \lambda^n = \sum_{n=0}^{\infty} q^n = \frac{1}{1-q}$$

while

$$-\underset{\geqq}{\Omega} \sum_{n=1}^{\infty} q^{-n} \lambda^{-n} = 0.$$

But if we allowed a purely formal application of the geometric series, then both initial expressions are

$$\frac{1}{1-\lambda q}.$$

To avoid confusion we will always have $\Omega_{\geqq}$ operate on variables denoted by letters in the middle of the Greek alphabet (e.g. λ, μ, ν). The parameters unaffected by $\Omega_{\geqq}$ will be denoted by letters from the Latin alphabet.

We can ensure that Definition 1.1 is mathematically well-posed provided we require that functions to which we apply $\Omega_{\geqq}$ have no singularities in the λ_i within a neighborhood of the circle $|\lambda_i| = 1$. While this suggests major problems in the abstract, it provides no difficulties whatsoever in practice because the only arguments of $\Omega_{\geqq}$ that ever arise are all of the form

$$\frac{P(x_1, \ldots, x_n; \lambda_1, \ldots, \lambda_r)}{\prod_{i=1}^{n}(1 - x_i\, \lambda_1^{v_1(i)} \cdots \lambda_r^{v_r(i)})} \tag{1}$$

where P is a Laurent polynomial in the $n+r$ variables and the $v_h(i)$ are integers not necessarily positive. As long as the x_i (which may be power products in other variables) are restricted to a small neighborhood of 0, we are guaranteed that we have avoided any singularity inside the annuli that provide the domain for the λ_i.

Let us now consider the instance $j = 3$ of Theorem 1.1. Obviously, the coefficient of $x^m q^n$ in

$$\frac{1}{(1-qx)(1-q^3x)(1-q^5x)} \tag{2}$$

equals the number of partitions of n into exactly m odd parts each of which is less than or equal to 5. On the other hand, the coefficient of $x^m q^n$ in

$$\underset{\geqq}{\Omega} \sum_{b_1,b_2,b_3 \geq 0} \lambda_1^{2b_3-3b_2} \lambda_2^{b_2-2b_1} x^{b_3-b_2+b_1} q^{b_1+b_2+b_3} \tag{3}$$

gives exactly the number of the desired lecture hall partitions for j being fixed to 3, because the Omega operator $\Omega_{\geqq}$ allows only those partitions $b_1 + b_2 + b_3 = n$ to be counted for which $2b_3 - 3b_2 \geq 0$ and $b_2 - 2b_1 \geq 0$. By geometric series expansion the three independent sums can be brought into product form, which means that expression (3) formulated as

$$\underset{\geqq}{\Omega} \sum_{b_1 \geq 0} \left(\frac{qx}{\lambda_2^2}\right)^{b_1} \sum_{b_2 \geq 0} \left(\frac{\lambda_2 q}{\lambda_1^3 x}\right)^{b_2} \sum_{b_3 \geq 0} (\lambda_1^2 qx)^{b_3}$$

can be rewritten as

$$\underset{\geqq}{\Omega} \frac{1}{\left(1-\frac{qx}{\lambda_2^2}\right)\left(1-\frac{\lambda_2 q}{\lambda_1^3 x}\right)(1-\lambda_1^2 qx)}. \tag{4}$$

Note that this is an instance of (1) with $n = 3$, $x_1 = x_3 = qx$, $x_2 = q/x$; so our convergence conditions require q to be in a small neighborhood of 0 while

x is in a neighborhood of 1. Therefore all that remains for proving the Lecture Hall Partition Theorem for $j = 3$ is to show equality of the generating function expressions (2) and (4). To do so we need the following lemma.

LEMMA 1.1: *For any integer $s \geq 0$,*

$$\underset{\geqq}{\Omega} \frac{1}{(1-\lambda x)\left(1-\frac{y}{\lambda^s}\right)} = \frac{1}{(1-x)(1-x^s y)}.$$

Proof: By geometric series expansion and application of the $\Omega_{\geqq}$ operator the left hand side equals

$$\underset{\geqq}{\Omega} \sum_{i,j\geq 0} \lambda^{i-sj} x^i y^j = \sum_{j\geq 0, i\geq sj} x^i y^j = \sum_{j,k\geq 0} x^{sj+k} y^j.$$

Geometric series summation completes the proof. □

With this lemma in hand, the proof of "(2) = (4)" reduces to successive elimination of the $\Omega_{\geqq}$ parameters λ_1 and λ_2.

Proof of the Lecture Hall Partition Theorem for $j = 3$: Split (4) additively into two parts by applying partial fraction decomposition

$$\frac{1}{1-t^2} = \frac{1}{2(1-t)} + \frac{1}{2(1+t)}$$

to the term $1/(1-\lambda_1^2 qx)$. Then by using Lemma 1.1 eliminate from both summands the parameter λ_1. For the last step one observes that Lemma 1.1 can be applied again in order to eliminate λ_2; this way one arrives at (2). □

We mention that this elimination is carried out automatically by the `Omega` package in a slightly modified manner as shown in Section 3.2; see In[2] and In[3] there.

This example reveals that algebraic manipulation is a central element in MacMahon's method; consequently a computer algebra implementation should, indeed, allow many more applications than MacMahon could carry out by hand. In Section 2 we explain how such an implementation can be achieved in a fairly general setting based on the "fundamental recurrence". In Sections 3.1 and 3.2 a description of the corresponding `Omega` package is given. This package is a collection of procedures that implement the method in the computer algebra system Mathematica. In addition to the introductory examples of how to use the package, in the remaining sections the reader finds further applications that illustrate the powerful combination of MacMahon's classical Partition Analysis with modern computer algebra tools.

In Section 3.3 we apply a related operator, $\Omega_=$, to linear homogeneous diophantine equations. We have postponed the presentation of $\Omega_=$ to Section 3.3

in order to avoid overloading the Introduction with definitions. In Section 4 one finds further `Omega` applications of less elementary nature. Section 4.1 deals with MacMahon's problem of "solid partitions on a cube" which he was able to solve only after the introduction of 18 (!) case distinctions with hairy computations. With the `Omega` package the same problem now finds a straightforward *automatic* solution. Section 4.2 deals with a non-trivial problem originally raised by Hermite, whose solution is automatic with the `Omega` package. Based on heuristics extracted from `Omega` computations, we embed Hermite's problem in a general setting which then is treated in a purely combinatorial manner in Section 4.3. Finally, in Section 4.4 we introduce "k-gon partitions". In connection with various generating functions computed with the `Omega` package, we conclude with an open problem.

Note: The present paper is the third in a major project devoted to MacMahon's method. In all the forthcoming papers it is cited as number III in our series of "MacMahon's Partition Analysis" articles. For further information on the status of this project the interested reader is referred to the Web page mentioned in Section 3.1.

2. The Fundamental Recurrence

We examined MacMahon's use of Partition Analysis with the object of generalizing his method to a pure algorithm. To this end we begin by listing a condensed version of his catalog [7, Vol. II, pp. 102–103] of fundamental evaluations of the Omega operator. Note that the elimination rule described by Lemma 1.1 is the first entry in this list.

$$\underset{\geq}{\Omega}\frac{1}{(1-\lambda x)\left(1-\frac{y}{\lambda^s}\right)}=\frac{1}{(1-x)(1-x^s y)},$$

$$\underset{\geq}{\Omega}\frac{1}{(1-\lambda^s x)\left(1-\frac{y}{\lambda}\right)}=\frac{1+xy\,\frac{1-y^{s-1}}{1-y}}{(1-x)(1-xy^s)},$$

$$\underset{\geq}{\Omega}\frac{1}{(1-\lambda x)\left(1-\frac{y}{\lambda}\right)\left(1-\frac{z}{\lambda}\right)}=\frac{1}{(1-x)(1-xy)(1-xz)},$$

$$\underset{\geq}{\Omega}\frac{1}{(1-\lambda x)(1-\lambda y)\left(1-\frac{z}{\lambda}\right)}=\frac{1-xyz}{(1-x)(1-y)(1-xz)(1-yz)},$$

$$\underset{\geq}{\Omega}\frac{1}{(1-\lambda x)(1-\lambda y)\left(1-\frac{z}{\lambda^2}\right)}=\frac{1+xyz-x^2yz-xy^2z}{(1-x)(1-y)(1-x^2z)(1-y^2z)},$$

$$\underset{\geq}{\Omega}\frac{1}{(1-\lambda^2 x)\left(1-\frac{y}{\lambda}\right)\left(1-\frac{z}{\lambda}\right)}=\frac{1+xy+xz+xyz}{(1-x)(1-xy^2)(1-xz^2)},$$

$$\underset{\geq}{\Omega}\frac{1}{(1-\lambda^2 x)(1-\lambda y)\left(1-\frac{z}{\lambda}\right)}=\frac{1+xz-xyz-xyz^2}{(1-x)(1-y)(1-yz)(1-xz^2)},$$

$$\underset{\geq}{\Omega}\frac{1}{(1-\lambda x)(1-\lambda y)(1-\lambda z)\left(1-\frac{w}{\lambda}\right)}$$

$$= \frac{1 - xyw - xzw - yzw + xyzw + xyzw^2}{(1-x)(1-y)(1-z)(1-xw)(1-yw)(1-zw)},$$

$$\underset{\geqq}{\Omega} \frac{1}{(1-\lambda x)(1-\lambda y)\left(1-\frac{z}{\lambda}\right)\left(1-\frac{w}{\lambda}\right)}$$
$$= \frac{1 - xyz - xyw - xyzw + xy^2zw + x^2yzw}{(1-x)(1-y)(1-xz)(1-xw)(1-yz)(1-yw)}.$$

These nine formulas and the problems solved by MacMahon using these formulas make clear that we need a general algorithm for the evaluation of

$$\underset{\geqq}{\Omega} \frac{\lambda^a}{(1-x_1\lambda)(1-x_2\lambda)\cdots(1-x_n\lambda)\left(1-\frac{y_1}{\lambda}\right)\left(1-\frac{y_2}{\lambda}\right)\cdots\left(1-\frac{y_m}{\lambda}\right)} \tag{5}$$

(where m and n are nonnegative integers and a is any integer).

At first glance, this appears to be inadequate to cover MacMahon's nine identities. Some of his identities have expressions of the form $(1-X\lambda^r)$ or $(1-Y\lambda^{-s})$ in the denominator. However these expressions are easily rendered as products that fit our example because

$$\begin{aligned} (1-X\lambda^r) &= \prod_{j=0}^{r-1}(1-\rho^j X^{1/r}\lambda), \\ (1-Y\lambda^{-s}) &= \prod_{j=0}^{s-1}(1-\sigma^j Y^{1/s}\lambda^{-1}), \end{aligned} \tag{6}$$

where $\rho = e^{2\pi i/r}$ and $\sigma = e^{2\pi i/s}$. The fundamental theorem of symmetric functions guarantees that the fractional roots of X and Y will disappear in the final answer.

Also one can envisage in the numerator of (5) a general Laurent polynomial in λ. However the linearity of $\Omega_{\geqq}$ shows that again our expression covers these cases as well.

Before we state our result, we must recall the homogeneous symmetric functions, denoted by $h_j(x_1, x_2, \ldots, x_n)$, which are given by

$$\sum_{j=0}^{\infty} h_j(x_1, x_2, \ldots, x_n)\, t^j = \frac{1}{(1-tx_1)(1-tx_2)\cdots(1-tx_n)}.$$

Finally we should dispose of the degenerate cases when either n or m is 0. In these cases, the effect of $\Omega_{\geqq}$ is immediate by inspection:

Lemma 2.1: *For any integer a,*

$$\underset{\geqq}{\Omega} \frac{\lambda^a}{(1-x_1\lambda)(1-x_2\lambda)\cdots(1-x_n\lambda)} = \underset{\geqq}{\Omega} \sum_{j=0}^{\infty} h_j(x_1,\ldots,x_n)\,\lambda^{a+j}$$
$$= \begin{cases} \dfrac{1}{(1-x_1)(1-x_2)\cdots(1-x_n)}, & \text{if } a \geq 0, \\ \dfrac{1}{(1-x_1)(1-x_2)\cdots(1-x_n)} - \displaystyle\sum_{j=0}^{-a-1} h_j(x_1,\ldots,x_n), & \text{if } a < 0. \end{cases}$$

Similarly, we have:

Lemma 2.2: *For any integer a,*

$$\underset{\geqq}{\Omega} \frac{\lambda^a}{\left(1-\frac{y_1}{\lambda}\right)\left(1-\frac{y_2}{\lambda}\right)\cdots\left(1-\frac{y_m}{\lambda}\right)} = \underset{\geqq}{\Omega} \sum_{j=0}^{\infty} h_j(y_1,\ldots,y_m)\,\lambda^{a-j}$$
$$= \begin{cases} 0, & \text{if } a < 0, \\ \displaystyle\sum_{j=0}^{a} h_j(y_1,\ldots,y_m), & \text{if } a \geq 0. \end{cases}$$

The main recurrence for the $\Omega_{\geqq}$ calculus reads as follows.

Theorem 2.1 ("Fundamental Recurrence"): *For n and m positive integers and a any integer,*

$$\underset{\geqq}{\Omega} \frac{\lambda^a}{(1-x_1\lambda)(1-x_2\lambda)\cdots(1-x_n\lambda)\left(1-\frac{y_1}{\lambda}\right)\left(1-\frac{y_2}{\lambda}\right)\cdots\left(1-\frac{y_m}{\lambda}\right)}$$
$$= \frac{P_{n,m,a}(x_1,\ldots,x_n;y_1,\ldots,y_m)}{\prod_{i=1}^{n}(1-x_i)\cdot\prod_{i=1}^{n}\prod_{j=1}^{m}(1-x_iy_j)}, \tag{7}$$

where for n > 1,

$$P_{n,m,a}(x_1,\ldots,x_n;y_1,\ldots,y_m) = \frac{1}{x_n - x_{n-1}}$$
$$\cdot\Big\{x_n(1-x_{n-1})\cdot\prod_{j=1}^{m}(1-x_{n-1}y_j)\cdot P_{n-1,m,a}(x_1,\ldots,x_{n-2},x_n;y_1,\ldots,y_m)$$
$$- x_{n-1}(1-x_n)\cdot\prod_{j=1}^{m}(1-x_ny_j)\cdot P_{n-1,m,a}(x_1,\ldots,x_{n-2},x_{n-1};y_1,\ldots,y_m)\Big\}$$

and for n = 1,

$$P_{1,m,a}(x_1;y_1,\ldots,y_m)$$
$$= \begin{cases} x_1^{-a}, & \text{if } a \leq 0, \\ x_1^{-a} + \displaystyle\prod_{j=1}^{m}(1-x_1y_j)\cdot\sum_{j=0}^{a} h_j(y_1,\ldots,y_m)(1-x_1^{j-a}), & \text{if } a > 0. \end{cases}$$

Remark: The form of the denominator in the evaluation guarantees that convergence conditions are maintained as we successively apply this theorem in the algorithm.

Proof: The proof of the main recurrence follows simply from the fact that

$$\frac{1}{(1-x_n\lambda)(1-x_{n-1}\lambda)} = \frac{1}{x_n - x_{n-1}}\Big(\frac{x_n}{1-x_n\lambda} - \frac{x_{n-1}}{1-x_{n-1}\lambda}\Big).$$

It is sufficient to carry out the proof under the assumption $x_i \neq x_j$ $(i \neq j)$, because the general case is an immediate consequence of the following elementary fact: if $T(x_1,\ldots,x_n;\lambda_1,\ldots,\lambda_r)$ is a term of the form (1) then

$$\underset{\geqq}{\Omega} \lim_{x_i\to x_j} T(x_1,\ldots,x_n;\lambda_1,\ldots,\lambda_r) = \lim_{x_i\to x_j} \underset{\geqq}{\Omega}\, T(x_1,\ldots,x_n;\lambda_1,\ldots,\lambda_r).$$

(The limit is understood to be taken within the corresponding domain of convergence.)

Hence for $n > 1$

$$\begin{aligned}
&\underset{\geqq}{\Omega} \frac{\lambda^a}{(1-x_1\lambda)(1-x_2\lambda)\cdots(1-x_n\lambda)\big(1-\frac{y_1}{\lambda}\big)\big(1-\frac{y_2}{\lambda}\big)\cdots\big(1-\frac{y_m}{\lambda}\big)}\\
&\quad = \frac{1}{x_n - x_{n-1}}\\
&\qquad \cdot \Bigg\{ \underset{\geqq}{\Omega} \frac{x_n\lambda^a}{(1-x_1\lambda)\cdots(1-x_{n-2}\lambda)(1-x_n\lambda)\big(1-\frac{y_1}{\lambda}\big)\cdots\big(1-\frac{y_m}{\lambda}\big)}\\
&\qquad - \underset{\geqq}{\Omega} \frac{x_{n-1}\lambda^a}{(1-x_1\lambda)\cdots(1-x_{n-2}\lambda)(1-x_{n-1}\lambda)\big(1-\frac{y_1}{\lambda}\big)\cdots\big(1-\frac{y_m}{\lambda}\big)} \Bigg\}.
\end{aligned}$$

This is exactly the main recurrence once the expressions involving $P_{n,m,a}$ have been substituted and the left denominator cleared.

For the $n = 1$ case, we see that

$$\begin{aligned}
\frac{P_{1,m,a}(x_1;y_1,\ldots,y_m)}{(1-x_1)(1-x_1y_1)\cdots(1-x_1y_m)} &= \underset{\geqq}{\Omega} \frac{\lambda^a}{(1-x_1\lambda)\big(1-\frac{y_1}{\lambda}\big)\cdots\big(1-\frac{y_m}{\lambda}\big)}\\
&= \underset{\geqq}{\Omega} \sum_{n=0}^{\infty}\sum_{j=0}^{\infty} h_j(y_1,\ldots,y_m)\, x_1^n \lambda^{n+a-j}.
\end{aligned}$$

Now if $a \leq 0$, then this expression is

$$\begin{aligned}
\sum_{j=0}^{\infty}\sum_{n=j-a}^{\infty} h_j(y_1,\ldots,y_m)\, x_1^n &= \frac{\sum_{j=0}^{\infty} h_j(y_1,\ldots,y_m)\, x_1^{j-a}}{1-x_1}\\
&= \frac{x_1^{-a}}{(1-x_1)(1-x_1y_1)\cdots(1-x_1y_m)},
\end{aligned}$$

and so for $a \leq 0$

$$P_{1,m,a}(x_1; y_1, \ldots, y_m) = x_1^{-a}.$$

If $a > 0$, then we have

$$\begin{aligned}
&\underset{\geqq}{\Omega} \sum_{n=0}^{\infty} \sum_{j=0}^{\infty} h_j(y_1, \ldots, y_m)\, x_1^n \lambda^{n+a-j} \\
&= \underset{\geqq}{\Omega} \sum_{n=0}^{\infty} \Big(\sum_{j=0}^{a} + \sum_{j=a+1}^{\infty} \Big) h_j(y_1, \ldots, y_m)\, x_1^n \lambda^{n+a-j} \\
&= \frac{\sum_{j=0}^{a} h_j(y_1, \ldots, y_m)}{1 - x_1} + \sum_{j=a+1}^{\infty} \sum_{n=j-a}^{\infty} h_j(y_1, \ldots, y_m)\, x_1^n \\
&= \frac{\sum_{j=0}^{a} h_j(y_1, \ldots, y_m)}{1 - x_1} + \frac{\sum_{j=a+1}^{\infty} h_j(y_1, \ldots, y_m)\, x_1^{j-a}}{1 - x_1} \\
&= \frac{\sum_{j=0}^{a} h_j(y_1, \ldots, y_m)}{1 - x_1} + \frac{x_1^{-a}}{(1 - x_1)(1 - x_1 y_1) \cdots (1 - x_1 y_m)} \\
&\qquad - \frac{\sum_{j=0}^{a} h_j(y_1, \ldots, y_m)\, x_1^{j-a}}{1 - x_1} \\
&= \frac{1}{1 - x_1} \sum_{j=0}^{a} h_j(y_1, \ldots, y_m)(1 - x_1^{j-a}) + \frac{x_1^{-a}}{(1 - x_1)(1 - x_1 y_1) \cdots (1 - x_1 y_m)} \\
&= \frac{x_1^{-a} + \prod_{j=1}^{m}(1 - x_1 y_j) \cdot \sum_{j=0}^{a} h_j(y_1, \ldots, y_m)(1 - x_1^{j-a})}{(1 - x_1)(1 - x_1 y_1) \cdots (1 - x_1 y_m)},
\end{aligned}$$

which gives the desired formula for $P_{1,m,a}$. □

3. The Mathematica Implementation

The object of this section is to describe the usage of the `Omega` package which has been implemented by the third author. In order to illustrate how the package is used in practice, a few tutorial examples are given. In Section 4 the reader finds further applications of less elementary nature.

3.1. The `Omega` package

The package consists of the Mathematica file `Omega.m` and the small documentation file `Readme.txt`; both can be downloaded from the `Omega` homepage at *http://www.risc.uni-linz.ac.at/research/combinat/risc/software/Omega/*.

After loading the package with `<<Omega.m`, the functions `OR` (for `O`mega `R`ule) and `OEqR` (for `O`mega `Eq`ual `R`ule) are provided. According to the fundamental recurrence (with respect to n) given in Theorem 2.1, `OR` applies the operator $\Omega_{\geqq}$ with respect to a certain variable, say λ, to an expression. Analogously, `OEqR` applies the operator $\Omega_{=}$; see Section 3.3. The calling syntax is

$$\texttt{OR[}expr\texttt{, }\lambda\texttt{]} \quad \text{and} \quad \texttt{OEqR[}expr\texttt{, }\lambda\texttt{]}$$

where λ is a variable (i.e., a Mathematica symbol) and *expr* is a rational function of the form

$$\frac{L(\lambda)}{\big(1 \pm p_1(\lambda)\big) \cdots \big(1 \pm p_d(\lambda)\big)}$$

with

$L(\lambda)$ a Laurent polynomial in λ over $\mathbb{Q}(z_1, \dots, z_l)$, where the z_i are indeterminates different from λ,

$p_i(\lambda)$ power products (with integer exponents) in λ and $z_1, \dots, z_l$.

The output of `OR` and `OEqR` is a rational function free of λ. While the denominator of the result can be read off almost immediately from (7), for the numerator we have to compute and add all terms corresponding to the different powers of λ appearing in $L(\lambda)$. Moreover, if *expr* is of the form

$$\frac{L(\lambda_1, \dots, \lambda_r)}{\big(1 \pm p_1(\lambda_1, \dots, \lambda_r)\big) \cdots \big(1 \pm p_d(\lambda_1, \dots, \lambda_r)\big)},$$

then Theorem 2.1 together with the linearity of $\Omega_{\geqq}$ guarantees that all λ_i can be eliminated in turn from *expr*. In other words, each application of `OR` (or `OEqR`) produces valid input for the next elimination call.

For involved applications it turns out that the numerator of the result sometimes gets so complicated that it cannot be factored by Mathematica in reasonable time. In this case, calling

$$\texttt{OR[}expr\texttt{, }\lambda\texttt{, FactorProc->FactorSquareFree]}$$

performs only square free factorization. If this still does not solve the problem, factorization can be avoided completely by calling

$$\texttt{OR[}expr\texttt{, }\lambda\texttt{, FactorProc->None].}$$

However note that in both cases the numerator and denominator of the result might contain common factors. The option `FactorProc` is also accepted by `OEqR`.

From the programmer's point of view it is worth remarking that the main difficulties concerning the implementation were caused by decomposition (6), since Mathematica is not able to handle roots of unity efficiently. For instance, consider the factorization of $1 - x\lambda^5$ into $\prod_{i=0}^{4}(1 - x^{(i)}\lambda)$, where the $x^{(i)}$ denote the fifth roots of x. Then the only way to reconstruct the original term $1 - x\lambda^5$ from this decomposition is to apply the Mathematica functions `Expand` and `FullSimplify`, which results in an incredibly bad runtime behavior even for simple applications. We could finally overcome this problem heuristically by observing that after evaluating such polynomials numerically, the imaginary parts of the coefficients vanish immediately and the remaining real (integer!) parts can be reestablished easily. We want to emphasize that in the next release of the package (Version 2) we will utilize a generalized partial fraction decomposition to completely avoid these problems with roots of unity. The method will be described in a forthcoming paper of the authors.

Concerning the run-time, all examples shown in this paper only take a few seconds on an SGI Octane except the one presented in Section 4.1 which needs approximately 40 seconds. From Theorem 2.1 and the linearity of the Omega operator one sees that the complexity mainly depends on

- the number n, which in our setting just equals the sum of all positive exponents of λ in the power products p_i,
- the exponents of λ in $L(\lambda)$, which take influence on the initial values of the recurrences, and
- the number of monomials in $L(\lambda)$.

3.2. How to use the `Omega` package in practice

We run our Mathematica session in the same directory in which we have put the file `Omega.m` (together with the file `Readme.txt`). After invoking Mathematica we load the package:

```
In[1]:= <<Omega.m
        Axel Riese's Omega implementation version 1.4 loaded
```

Now the proof of Theorem 1.1 for $j = 3$ can simply be done as follows. First we input the expression the $\Omega_{\geq}$ operator acts on; see (4) and the preceding discussion:

```
In[2]:= f = 1 / ((1-q x/λ2^2)(1-λ2 q/(λ1^3 x))(1-λ1^2 q x))
```

Out[2]=

$$\frac{1}{\left(1-\frac{\lambda_2 q}{\lambda_1^3 x}\right)(1-\lambda_1^2 q x)\left(1-\frac{q x}{\lambda_2^2}\right)}$$

Then we call the procedure `OR` to eliminate the variables λ_1 and λ_2:

```
In[3]:= OR[f, λ1]
```

Out[3]=

$$\frac{1+\lambda_2 q^3 x}{(1-q x)\left(1-\frac{q x}{\lambda_2^2}\right)(1-\lambda_2^2 q^5 x)}$$

```
In[4]:= OR[%, λ2]
```

Out[4]=

$$\frac{1}{(1-q x)(1-q^3 x)(1-q^5 x)}$$

This proves the equality in question.

Alternatively, we can prove "(2) = (4)" by reversing the order of elimination:

```
In[5]:= OR[f, λ2]
```

Out[5]=

$$\frac{1}{\left(1-\frac{q}{\lambda_1^3\,x}\right)\left(1-\frac{q^3}{\lambda_1^6\,x}\right)\left(1-\lambda_1^2\,q\,x\right)}$$

```
In[6]:= OR[%, λ1]
```

Out[6]=

$$\frac{1}{(1-q\,x)\,(1-q^3\,x)\,(1-q^5\,x)}$$

This means, if we reverse the order in which the λ_i are eliminated, we obtain a different intermediate result. This fact is of particular importance in more involved situations. In other words, in applications where we need to eliminate several variables a certain order of elimination might turn out to be optimal with respect to running time. See, for instance, the generating function for solid partitions on a cube in Section 4.1.

3.3. Linear homogeneous diophantine equations

For one further introductory application we look at linear homogeneous diophantine *equations*. Also this problem area has been studied by MacMahon extensively. To this end MacMahon defined a different Omega operator.

Definition 3.1: The operator $\Omega_=$ is given by

$$\underset{=}{\Omega} \sum_{s_1=-\infty}^{\infty} \cdots \sum_{s_r=-\infty}^{\infty} A_{s_1,\ldots,s_r} \lambda_1^{s_1} \cdots \lambda_r^{s_r} := A_{0,\ldots,0}.$$

This means, all non-trivial power products in the λ's are killed by the $\Omega_=$ operator.

As already pointed out by MacMahon [7, Vol. II, Sect. VIII, p. 104], this operator is related to $\Omega_{\geqq}$, for instance, as follows:

$$\underset{=}{\Omega}\, F(\lambda) = \underset{\geqq}{\Omega}\, F(\lambda) + \underset{\geqq}{\Omega}\, F(1/\lambda) - F(1). \tag{8}$$

We use exactly this relation in order to find a parameterized representation of all tuples (a_1, a_2, a_3, a_4) of nonnegative integers satisfying $a_1 + a_2 - a_3 - a_4 = 0$. (See Stanley [9, Ch. 4, Example 4.6.15].) Equivalent to this is the computation of the corresponding generating function, i.e.,

$$\underset{=}{\Omega} \sum_{a_1,a_2,a_3,a_4 \geq 0} \lambda^{a_1+a_2-a_3-a_4} x_1^{a_1} x_2^{a_2} x_3^{a_3} x_4^{a_4} = \underset{=}{\Omega} \frac{1}{(1-x_1\lambda)(1-x_2\lambda)\left(1-\frac{x_3}{\lambda}\right)\left(1-\frac{x_4}{\lambda}\right)}.$$

Using (8) the elimination is done as follows:

```
In[2]:= f = 1 / ((1-x1 λ)(1-x2 λ)(1-x3/λ)(1-x4/λ));

In[3]:= Factor[OR[f, λ] + OR[f /. λ->1/λ, λ] - (f /. λ->1)]
```

Out[3]=

$$-\frac{-1+x_1\,x_2\,x_3\,x_4}{(-1+x_1\,x_3)\,(-1+x_2\,x_3)\,(-1+x_1\,x_4)\,(-1+x_2\,x_4)}$$

As another example from Stanley's book [9, Ch. 4, Prop. 4.6.21], let us count the number $S_3(r)$ of 3×3 *symmetric* matrices with nonnegative integer entries such that every row (and column) sum equals r. If (a_1, a_2, a_3) stands for the first, (a_2, a_4, a_5) for the second, and (a_3, a_5, a_6) for the third row, then the corresponding linear system of homogeneous equations is

$$a_1 + a_2 + a_3 - r = 0,\ a_2 + a_4 + a_5 - r = 0,\ \text{and } a_3 + a_5 + a_6 - r = 0.$$

The solution in generating function form can be written down immediately by means of the $\Omega_=$ operator,

$$\underset{=}{\Omega} \sum_{a_1,\dots,a_6,r\ge 0} \lambda_1^{a_1+a_2+a_3-r}\lambda_2^{a_2+a_4+a_5-r}\lambda_3^{a_3+a_5+a_6-r}x_1^{a_1}x_2^{a_2}x_3^{a_3}x_4^{a_4}x_5^{a_5}x_6^{a_6}y^r$$
$$= \underset{=}{\Omega}\ \frac{1}{(1-x_1\lambda_1)(1-x_2\lambda_1\lambda_2)(1-x_3\lambda_1\lambda_3)(1-x_4\lambda_2)(1-x_5\lambda_2\lambda_3)}\cdot \frac{1}{(1-x_6\lambda_3)\big(1-\frac{y}{\lambda_1\lambda_2\lambda_3}\big)}.$$

For the elimination we use the procedure "`OEqR`" that encodes relation (8):

```
In[4]:= SymMS = 1 / ((1-λ1 x1)(1-λ1 λ2 x2) (1-λ1 λ3 x3)(1-λ2 x4) *
                     (1-λ2 λ3 x5)(1-λ3 x6)(1-y/(λ1 λ2 λ3)));

In[5]:= OEqR[ OEqR[ OEqR[SymMS, λ1], λ2], λ3]
```

Out[5]=

$$\frac{1-x_1\,x_2\,x_3\,x_4\,x_5\,x_6\,y^3}{(1-x_3\,x_4\,y)\,(1-x_1\,x_5\,y)\,(1-x_2\,x_6\,y)\,(1-x_1\,x_4\,x_6\,y)\,(1-x_2\,x_3\,x_5\,y^2)}$$

```
In[6]:= % /. x_ -> 1
```

Out[6]=

$$\frac{1-y^3}{(1-y)^4\,(1-y^2)}$$

Obviously, $S_3(r)$ is the coefficient of y^r after setting all the x_i's to 1 in the computed λ-free generating function expression. This specialization, Out[6], can be found in [9, Ch. 4, after Prop. 4.6.21].

Finally, we remark that there are more efficient ways than using (8) to implement the $\Omega_=$ operator. For more details, including an $\Omega_=$ analogue to the "Fundamental Recurrence" (Thm. 2.1), we refer the interested reader to the forthcoming article [4].

4. Further Applications

In this section we present further applications of `Omega` that are of less elementary nature.

4.1. Solid partitions on a cube

MacMahon devoted *Art. 98* of [6, Sect. 7] to the consideration of the simplest "lattice in solido"; namely, the lattice "in which the points are the summits of a cube and the branches the edges of the cube." In other words, following MacMahon let us put nonnegative integer weights a_i $(1 \leq i \leq 8)$ on the vertices of a cube as described by Figure 1 below.

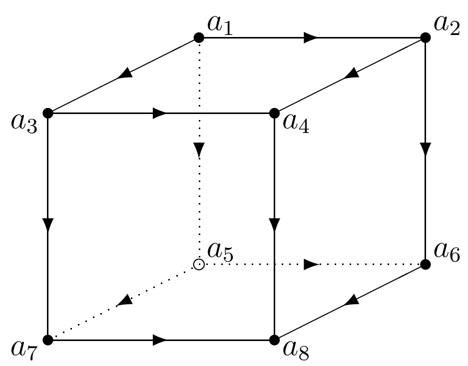

Figure 1: MacMahon's solid cube

The vertices of the cube in Figure 1 are connected by *directed* edges that are interpreted as inequalities. For instance, the directed edge from the vertex with weight a_1 to the vertex with weight a_2 corresponds to the inequality relation $a_1 \geq a_2$, and so on. This way we have introduced 12 diophantine inequalities to which we relate the generating function

$$\sum q^{a_1+\cdots+a_8},$$

where the sum runs over all nonnegative integer tuples $(a_1, \ldots, a_8)$ with entries satisfying the 12 inequalities induced by the cube.

It is easy to check that this generating function is nothing but the $\Omega_{\geqq}$ operator applied to the expression

```
In[2]:= f = 1 / ((1-q λ1 λ2 λ3)(1-q λ4 λ5/λ1)(1-q λ6 λ7/λ2) *
                 (1-q λ8/(λ4 λ6))(1-q λ9 λ10/λ3)(1-q λ11/(λ5 λ9)) *
                 (1-q λ12/(λ7 λ10))(1-q/(λ8 λ11 λ12)));
```

from which we first eliminate λ_{12}:

In[3]:= `OR[f, λ12]`

Out[3]=

$$1/\Big((1-\lambda_1\,\lambda_2\,\lambda_3\,q)\,(1-\frac{\lambda_4\,\lambda_5\,q}{\lambda_1})\,(1-\frac{q}{\lambda_{10}\,\lambda_7})\,(1-\frac{\lambda_6\,\lambda_7\,q}{\lambda_2})\,(1-\frac{\lambda_8\,q}{\lambda_4\,\lambda_6})$$
$$(1-\frac{\lambda_{11}\,q}{\lambda_5\,\lambda_9})\,(1-\frac{\lambda_{10}\,\lambda_9\,q}{\lambda_3})(1-\frac{q^2}{\lambda_{10}\,\lambda_{11}\,\lambda_7\,\lambda_8})\Big)$$

Now one proceeds by eliminating the remaining λ's in some convenient order, for instance, in the order (λ_{11}, λ_{10}, λ_1, λ_8, λ_7, λ_5, λ_9, λ_6, λ_3, λ_4).

This way one keeps the numerator equal to 1 until `OR[%, λ9]`. For readers who want to use the `Omega` package in practice we point to the important fact that other orders of elimination might result in (much) more involved numerator polynomials that might slow down the computation tremendously. A good heuristic principle for finding a "good" order is to apply `OR` with respect to a λ-variable that occurs less frequently than others.

After eliminating λ_4 the last step is:

In[14]:= `OR[%, λ2]`

Out[14]=

$$(1+2\,q^2+2\,q^3+3\,q^4+3\,q^5+5\,q^6+4\,q^7+8\,q^8+4\,q^9+5\,q^{10}+3\,q^{11}+$$
$$3\,q^{12}+2\,q^{13}+2\,q^{14}+q^{16})\;/$$
$$\big((1-q)(1-q^2)(1-q^3)(1-q^4)(1-q^5)(1-q^6)(1-q^7)(1-q^8)\big)$$

In this way we have computed the generating function $\sum q^{a_1+\cdots+a_8}$ in less than one minute of computation time.

At the time of MacMahon the situation was quite a different one. It is instructive to compare how he managed to solve the $\Omega_{\geqq}$ elimination problem. Namely, MacMahon divided the calculation into eighteen (!) parts (according to all possible inequality relations between weights corresponding to non-adjacent vertices [6, Art. 98]) whose sum then gives the desired generating function. We conclude this section with the footnote added by MacMahon after the description of his computation, "Mr. A.B. KEMPE, Treas. R.S., has verified this conclusion by a different and most ingenious method of summation, which also readily yields the result for any desired restriction on the part-magnitude."

4.2. A problem of Hermite

In their famous book [8, Ex. 31], Pólya and Szegő posed the following problem:

PROBLEM 4.1: *For an integer n greater than 2, let $h(n)$ be the number of positive integer triples (a,b,c) such that $a+b+c=n$ and $a\le b+c$, $b\le a+c$, and $c\le a+b$. Show that $h(n)=(n+8)(n-2)/8$, if n is even, and $h(n)=(n^2-1)/8$, if n is odd.*

Pólya and Szegő remark that the problem was originally posed by Ch. Hermite in 1868 and solved by V. Schlegel in 1869. (In [8, Solution to Ex. 31] the exact references can be found.)

For the first case, $n = 3$, it is obvious that $(1, 1, 1)$ is the only solution, thus $h(3) = 1$. If $n = 4$ we have three solutions, namely $(1, 1, 2)$, $(1, 2, 1)$, and $(2, 1, 1)$; this means, $h(4) = 3$. The general case can be settled *automatically* with the `Omega` package as follows.

First we encode the situation in form of a generating function the $\Omega_{\geqq}$ operator acts on:

$$\sum_{n=3}^{\infty} h(n)\, q^n = \underset{\geqq}{\Omega} \sum_{a,b,c\geq 1} \lambda_1^{b+c-a}\lambda_2^{a+c-b}\lambda_3^{a+b-c}q^{a+b+c}$$
$$= \underset{\geqq}{\Omega} \frac{\lambda_1\lambda_2\lambda_3\, q^3}{\left(1-\frac{\lambda_2\lambda_3 q}{\lambda_1}\right)\left(1-\frac{\lambda_1\lambda_3 q}{\lambda_2}\right)\left(1-\frac{\lambda_1\lambda_2 q}{\lambda_3}\right)}.$$

```
In[2]:= hfu = λ1 λ2 λ3 q^3 /
             ((1-λ2 λ3 q/λ1)(1-λ1 λ3 q/λ2)(1-λ1 λ2 q/λ3));
```

Then we successively eliminate the λ variables:

```
In[3]:= OR[ OR[ OR[hfu, λ1], λ2], λ3]
```

Out[3]=

$$-\frac{q^3\,(-1-2\,q+2\,q^2)}{(1-q)\,(1-q^2)^2}$$

Finally, from the following partial fraction representation of Out[3] the explicit formulae for $h(n)$ can be read off almost directly,

$$\sum_{n\geq 3} h(n)\, q^n = \frac{q^3(1+2q-2q^2)}{(1-q)(1-q^2)^2} = \frac{q^3}{(1-q^2)^3} + \frac{q^4}{(1-q^2)^3} + \frac{2q^4}{(1-q^2)^2}.$$

4.3. A generalization of Hermite's problem

To illustrate the suggestive power of the `Omega` package, we consider a natural generalization of Hermite's problem. For instance, as the next case we consider positive integer tuples (a_1, a_2, a_3, a_4) such that $a_1 + a_2 + a_3 + a_4 = n$ where each a_i is less than or equal to the sum of the others; or equivalently, where we have $a_i \leq n - a_i$ for all $i \in \{1, 2, 3, 4\}$.

Definition 4.1: As the set of "compositions of n into k positive parts" we define

$$C_k(n) := \{(a_1, \ldots, a_k) \in \mathbb{Z}^k \mid a_i \geq 1 \text{ for all } i, \text{ and } a_1 + \cdots + a_k = n\}.$$

As the set of "k-gon compositions of n into positive parts" we define

$$H_k(n) := \{(a_1, \ldots, a_k) \in C_k(n) \mid a_i \leq n - a_i \text{ for all } i\}.$$

Note that $h(n) = |H_3(n)|$. With the `Omega` package it is as easy as in the case $k = 3$ to compute the generating functions of $|H_k(n)|$ for the next values. For instance, for $k = 4$ and $k = 5$ one obtains

$$\sum_{n\geq 4} |H_4(n)|\, q^n = \frac{q^4}{(1-q)^4} - 4\frac{q^7}{(1-q)^4(1+q)^3}$$

and

$$\sum_{n\geq 5} |H_5(n)|\, q^n = \frac{q^5}{(1-q)^5} - 5\frac{q^9}{(1-q)^5(1+q)^4}.$$

We leave the verification as an `Omega` exercise.

If one brings the generating function for $|H_3(n)|$ into the analogous form, i.e.,

$$\sum_{n\geq 3} |H_3(n)|\, q^n = \sum_{n\geq 3} h(n)\, q^n = \frac{q^3}{(1-q)^3} - 3\frac{q^5}{(1-q)^3(1+q)^2},$$

the underlying pattern already becomes obvious. We state the resulting conjecture as a theorem, because — once discovered — its proof causes no further difficulty.

Theorem 4.1: *Let k be an integer greater or equal to 3, then*

$$\sum_{n\geq k} |H_k(n)|\, q^n = \frac{q^k}{(1-q)^k} - k\frac{q^{2k-1}}{(1-q)^k(1+q)^{k-1}}.$$

While the `Omega` package cannot prove such a theorem in full generality, it reveals its power by leading us to the correct formulation. We prove the statement combinatorially; to this end we need a bit of preparatory work.

The first quotient is nothing but the generating function for compositions, i.e.,

$$\sum_{n\geq k} |C_k(n)|\, q^n = \left(\frac{q}{1-q}\right)^k. \tag{9}$$

For the combinatorial interpretation of the second quotient we introduce a suitable composition set.

Definition 4.2: For integers $n \geq 3$ and $k \geq 2$ the set of compositions of n into k positive parts where at most one part is odd can be described by the set

$$O_k(n) := \{(e_1, \ldots, e_{k-1}, e_k) \in \mathbb{Z}^k \mid \\ e_i \geq 1 \text{ for all } i,\ \text{and } 2e_1 + \cdots + 2e_{k-1} + e_k = n\}.$$

Its generating function is very close to the second quotient.

Lemma 4.1: *For integers $k \geq 2$ and $n \geq 3$,*

$$\sum_{n \geq 2k-1} |O_k(n)|\, q^n = \frac{q^{2k-1}}{(1-q)^k (1+q)^{k-1}}.$$

Proof: The proof is obvious from

$$\sum_{n \geq 2k-1} |O_k(n)|\, q^n = \left(\frac{q^2}{1-q^2}\right)^{k-1} \cdot \frac{q}{1-q}.$$

□

For the proof of Theorem 4.1 it is also convenient to introduce the following lemma.

Lemma 4.2: *Let $(a_1, \ldots, a_k) \in C_k(n)$. If $a_i \geq n - a_i$ for some $i \in \{1, \ldots, k\}$ then $a_j < n - a_j$ for all $j \in \{1, \ldots, k\} \setminus \{i\}$.*

Proof: It suffices to prove the lemma for the case $i = k$. Suppose $a_j \geq n/2$ for some $j \in \{1, \ldots, k-1\}$. Then, since all parts are positive, we have $n = (a_1 + \cdots + a_{k-1}) + a_k > n/2 + n/2 = n$, a contradiction. □

After this preparatory work we are ready for the desired proof.

Proof of Theorem 4.1: From Lemma 4.2 we have for each $(a_1, \ldots, a_k) \in C_k(n) \setminus H_k(n)$ that $a_i > n - a_i$ for some $i \in \{1, \ldots, k\}$ and $a_j < n - a_j$ for all other indices $j \neq i$. This induces the following partition of the set $C_k(n) \setminus H_k(n)$ into k disjoint subsets, namely, $C_k(n) \setminus H_k(n) = \bigcup_{i \in \{1, \ldots, k\}} CH_k^{(i)}(n)$, where $CH_k^{(i)}(n) := \{(a_1, \ldots, a_k) \in C_k(n) \setminus H_k(n) \mid a_i > n - a_i\}$.

All these sets are of the same cardinality, i.e., for $i, j \in \{1, \ldots, k\}$ we have $|CH_k^{(i)}(n)| = |CH_k^{(j)}(n)|$. This is immediate from the fact that for $i < j$ the map

$$(a_1, \ldots, a_i, \ldots, a_j, \ldots, a_k) \mapsto (a_1, \ldots, a_j, \ldots, a_i, \ldots, a_k)$$

is a bijection.

Hence, again by Lemma 4.2, we get that for all integers $k \geq 2$ and $n \geq 3$,

$$|C_k(n)| - |H_k(n)| = k\, |CH_k^{(k)}(n)|.$$

In view of (9) and of Lemma 4.1, the proof of Theorem 4.1 is completed once we are able to show that $|CH_k^{(k)}(n)| = |O_k(n)|$. To this end we define the map

$$\begin{aligned} \phi: \quad & CH_k^{(k)}(n) \to O_k(n), \\ & (a_1, \ldots, a_k) \mapsto (a_1, \ldots, a_{k-1}, 2a_k - n). \end{aligned}$$

The map ϕ is well-defined, since $2(a_1 + \cdots + a_{k-1}) + (2a_k - n) = n$ and $2a_k - n > 0$.

Obviously, ϕ is an injective map. Finally we show that ϕ is also surjective, i.e., given any $(e_1, \ldots, e_{k-1}, e_k) \in O_k(n)$ one can find $(a_1, \ldots, a_k) \in CH_k^{(k)}(n)$ such that $\phi(a_1, \ldots, a_k) = (e_1, \ldots, e_{k-1}, e_k)$. We will verify that the choice

$$a_i := e_i \text{ for } i \in \{1, \ldots, k-1\} \text{ and } a_k := e_1 + \cdots + e_k$$

does the job. This can be seen as follows:

Obviously, $(a_1, \ldots, a_k)$ is a composition of n, since all the a_i's are positive and $a_1 + \cdots + a_k = 2(e_1 + \cdots + e_{k-1}) + e_k = n$. Also, $a_k > n - a_k$, since $2a_k = 2(e_1 + \cdots + e_k) = n + e_k > n$. Thus $(a_1, \ldots, a_k) \in CH_k^{(k)}(n)$. Finally, the fact that

$$\phi(a_1, \ldots, a_k) = (a_1, \ldots, a_{k-1}, 2a_k - n) = (e_1, \ldots, e_{k-1}, e_k),$$

completes the proof that ϕ is surjective. Therefore ϕ is a bijection, and Theorem 4.1 is proved. □

4.4. *k*-gon partitions

One can view the study of triangles with sides of integer size [3, Sect. 3] as a *partition* counterpart to Hermite's problem. More precisely, it gives rise to the following definition.

Definition 4.3: As the set of "non-degenerate k-gon partitions of n into positive parts" we define

$$T_k(n) := \{(a_1, \ldots, a_k) \in C_k(n) \mid a_1 \leq a_2 \leq \cdots \leq a_k \text{ and } a_k < n - a_k\}.$$

The term "non-degenerate" refers to the restriction to strict inequality, i.e., to $a_k < n - a_k$. (Note that $n - a_k = a_1 + \cdots + a_{k-1}$.) In [3, Sect. 3], Partition Analysis has been used to show that

$$\sum_{n \geq 3} |T_3(n)|\, q^n = \frac{q^3}{(1-q^2)(1-q^3)(1-q^4)}. \tag{10}$$

With the `Omega` package we are able to compute the next cases in purely mechanical way.

$$\sum_{n \geq 4} |T_4(n)|\, q^n = \frac{q^4(1+q+q^5)}{(1-q^2)(1-q^3)(1-q^4)(1-q^6)}, \tag{11}$$

$$\sum_{n \geq 5} |T_5(n)|\, q^n = \frac{q^5(1-q^{11})}{(1-q)(1-q^2)(1-q^4)(1-q^5)(1-q^6)(1-q^8)}, \tag{12}$$

and

$$\sum_{n\geq 6} |T_6(n)|\, q^n = \frac{q^6(1-q^4+q^5+q^7-q^8-q^{13})}{(1-q)(1-q^2)(1-q^3)(1-q^4)(1-q^6)(1-q^8)(1-q^{10})}. \quad (13)$$

From these results we can derive a number of consequences. For example,

$$|T_4(2n)| - |T_4(2n-1)| = g(n)$$

where $g(n)$ is the number of partitions of n into 2's and 3's with at least one 3. This implies that

$$g(n) = \begin{cases} \left\lceil \frac{n-1}{6} \right\rceil, & \text{if } n \equiv 3 \pmod 6, \\ \left\lceil \frac{n-1}{6} \right\rceil - 1, & \text{otherwise.} \end{cases}$$

Also we can show that

$$|T_5(2n)| - |T_5(2n-1)| = h(n)$$

where $h(n)$ is the number of partitions of n into 1's, 2's, 4's and 5's with at least one 5.

Each of these results is easily derived once we observe that if $a_{-1} := 0$ and

$$f(q) = \sum_{n=0}^{\infty} a_n q^n,$$

then

$$\sum_{n=0}^{\infty} (a_{2n} - a_{2n-1})\, q^{2n} = \frac{1}{2}(1-q)f(q) + \frac{1}{2}(1+q)f(-q).$$

It would be interesting to know if there are any simple combinatorial explanations of these observations.

The verification of the generating function representations (11), (12), and (13) is left to the reader as a routine `Omega` exercise. Moreover, it is also easily checked that

$$\sum_{n\geq k} |T_k(n)|\, q^n = \sum_{\substack{a_k\geq\cdots\geq a_1\geq 1\\ a_1+\cdots+a_{k-1}>a_k}} q^{a_1+\cdots+a_k}$$

$$= \underset{\geqq}{\Omega} \frac{q\lambda_1^{-1}}{(1-\lambda_{k-1}\, q/\lambda_k)\big(1-\lambda_{k-2}\lambda_k\, q/\lambda_{k-1}\big)\big(1-\lambda_{k-3}\lambda_k\, q/\lambda_{k-2}\big)\cdots(1-\lambda_k\, q/\lambda_1)}$$

is the corresponding $\Omega_{\geqq}$ representation for the generating function in full generality.

Despite the fact that the particular instances of $\sum_{n\geq k} |T_k(n)|\, q^n$ can be computed so easily, we were not able to find a common underlying pattern as in the case of k-gon compositions. This suggests to conclude by stating this question as an open problem:

PROBLEM 4.2: *In view of the generating function representations* (10)*,* (11)*,* (12)*, and* (13)*: Is it possible to find a common pattern for all possible choices of* k *as in the case of* k*-gon compositions (Theorem* 4.1*)?*

Acknowledgment. We wish to thank the anonymous referee and Burkhard Zimmermann for helpful remarks.

References

1. G.E. Andrews, *The Theory of Partitions*, Encyclopedia Math. Appl., Vol. 2, G.-C. Rota ed., Addison-Wesley, Reading, 1976. (Reissued: Cambridge University Press, Cambridge, 1985.)

2. G.E. Andrews, MacMahon's partition analysis I: The lecture hall partition theorem, in *Mathematical essays in honor of Gian-Carlo Rota*, Prog. Math., Vol. 161, pp. 1–22, Birkhäuser, Boston, 1998.

3. G.E. Andrews, MacMahon's partition analysis II: Fundamental theorems, *Ann. Comb.* **4** (2000), 327–338.

4. G.E. Andrews, P. Paule, A. Riese, and V. Strehl, MacMahon's partition analysis V: Bijections, recursions, and magic squares, in *Algebraic Combinatorics and Applications* (A. Betten et al., eds.), pp. 1–39, Springer, Berlin, 2001.

5. M. Bousquet-Mélou and K. Eriksson, Lecture hall partitions, *Ramanujan J.* **1** (1997), 101–111.

6. P.A. MacMahon, Memoir on the theory of the partition of numbers — Part VI, *Phil. Trans.* **211** (1912), 345–373.

7. P.A. MacMahon, *Combinatory Analysis*, 2 vols., Cambridge University Press, Cambridge, 1915–1916. (Reprinted: Chelsea, New York, 1960.)

8. G. Pólya and G. Szegő, *Aufgaben und Lehrsätze aus der Analysis I*, Springer, Berlin, 1970 (in German).

9. R.P. Stanley, *Enumerative Combinatorics — Vol. 1*, Wadsworth, Monterey, California, 1986.