Partition identities, functional equations and computer algebra

Jehanne Dousse

Université de Genève

Recent Trends in Computer Algebra
Computer Algebra for Functional Equations in Combinatorics and Physics
IHP, Paris, 8 December 2023

Outline

1. Introduction
   - Partitions and their generating functions
   - Partition identities
   - Motivation and goals
2. Preliminary step: obtaining a system of recurrences
   - Without colours
   - With colours (refinement)
3. First method: finding an iterable relation
4. Second method: the "back-and-forth" technique
   - From recurrences to q-difference equations
   - Back and forth between recurrences and q-difference equations

Outline

1 Introduction
- Partitions and their generating functions
- Partition identities
- Motivation and goals

2 Preliminary step: obtaining a system of recurrences
- Without colours
- With colours (refinement)

3 First method: finding an iterable relation

4 Second method: the "back-and-forth" technique
- From recurrences to q-difference equations
- Back and forth between recurrences and q-difference equations

Integer partitions

Definition

A *partition* λ of a positive integer n is a finite non-increasing sequence of positive integers $(\lambda_1, \dots, \lambda_m)$ such that $\lambda_1 + \cdots + \lambda_m = n$. The integers $\lambda_1, \dots, \lambda_m$ are called the *parts* of the partition λ.

Example

There are 5 partitions of 4: $(4), (3,1), (2,2), (2,1,1)$ and $(1,1,1,1)$.

Generating functions

Notation : $(a;q)_n = \prod_{k=0}^{n-1}(1-aq^k), n \in \mathbb{N} \cup \{\infty\}$.

Generating functions

Notation : $(a;q)_n = \prod_{k=0}^{n-1}(1-aq^k), n \in \mathbb{N} \cup \{\infty\}$.

Let $Q(n,k)$ be the number of partitions of n into k distinct parts. Then

$$\begin{aligned} 1 + \sum_{n\geq 1}\sum_{k\geq 1} Q(n,k) z^k q^n &= (1+zq)(1+zq^2)(1+zq^3)(1+zq^4)\cdots \\ &= (-zq;q)_\infty. \end{aligned}$$

Generating functions

Notation : $(a;q)_n = \prod_{k=0}^{n-1}(1-aq^k), n \in \mathbb{N} \cup \{\infty\}$.

Let $Q(n,k)$ be the number of partitions of n into k distinct parts. Then

$$\begin{aligned} 1 + \sum_{n\geq 1}\sum_{k\geq 1} Q(n,k) z^k q^n &= (1+zq)(1+zq^2)(1+zq^3)(1+zq^4)\cdots \\ &= (-zq;q)_\infty. \end{aligned}$$

Let $p(n,k)$ be the number of partitions of n into k parts. Then

$$\begin{aligned} 1 + \sum_{n\geq 1}\sum_{k\geq 1} p(n,k) z^k q^n &= \prod_{n\geq 1}\left(1+zq^n+z^2q^{2n}+\cdots\right) \\ &= \frac{1}{(zq;q)_\infty}. \end{aligned}$$

Generating functions

More generally:

- The generating function for partitions into distinct parts congruent to $k \mod N$ is
$$(-zq^k; q^N)_\infty.$$
- The generating function for partitions into parts congruent to k mod N is
$$\frac{1}{(zq^k; q^N)_\infty}.$$

Outline

1 Introduction
- Partitions and their generating functions
- Partition identities
- Motivation and goals

2 Preliminary step: obtaining a system of recurrences
- Without colours
- With colours (refinement)

3 First method: finding an iterable relation

4 Second method: the "back-and-forth" technique
- From recurrences to q-difference equations
- Back and forth between recurrences and q-difference equations

Partition identities

Partition identities

For all n, the number of partitions of n satisfying some conditions equals the number of partitions of n satisfying some other conditions.

Partition identities

Partition identities

For all n, the number of partitions of n satisfying some conditions equals the number of partitions of n satisfying some other conditions.

The simplest example of partition identity (Euler 1748)

For all n, the number of partitions of n into distinct parts equals the number of partitions of n into odd parts.

Example $n = 4$: (4), $(3, 1)$ into distinct parts,
$(3, 1)$, $(1, 1, 1, 1)$ into odd parts.

Partition identities

Partition identities

For all n, the number of partitions of n satisfying some conditions equals the number of partitions of n satisfying some other conditions.

The simplest example of partition identity (Euler 1748)

For all n, the number of partitions of n into distinct parts equals the number of partitions of n into odd parts.

Example $n = 4$: (4), $(3, 1)$ into distinct parts,
$(3, 1)$, $(1, 1, 1, 1)$ into odd parts.

Euler's theorem is easy to prove with generating functions (see next slide). But in general, partition identities are neither easy to guess nor easy to prove.

Euler's identity

Theorem (Euler 1748)

For all n, the number of partitions of n into distinct parts equals the number of partitions of n into odd parts.

Euler's identity

Theorem (Euler 1748)

For all n, the number of partitions of n into distinct parts equals the number of partitions of n into odd parts.

Proof.

$$\prod_{n\geq 1}(1+q^n) = \prod_{n\geq 1}\frac{(1+q^n)(1-q^n)}{1-q^n}$$
$$= \prod_{n\geq 1}\frac{1-q^{2n}}{1-q^n}$$
$$= \prod_{n\geq 1}\frac{1}{1-q^{2n-1}}.$$

□

Outline

1 Introduction
- Partitions and their generating functions
- Partition identities
- Motivation and goals

2 Preliminary step: obtaining a system of recurrences
- Without colours
- With colours (refinement)

3 First method: finding an iterable relation

4 Second method: the "back-and-forth" technique
- From recurrences to q-difference equations
- Back and forth between recurrences and q-difference equations

Where do partition identities come from?

- Combinatorics

Where do partition identities come from?

- Combinatorics
- Representation theory of affine Lie algebras (often more complicated identities): Lepowsky–Wilson 1984, Capparelli 1992, Meurman–Primc 1999, Siladić 2002, Nandi 2014, D.–Konan 2020-2023+

Where do partition identities come from?

- Combinatorics
- Representation theory of affine Lie algebras (often more complicated identities): Lepowsky–Wilson 1984, Capparelli 1992, Meurman–Primc 1999, Siladić 2002, Nandi 2014, D.–Konan 2020-2023+
- A few examples from other fields:

Where do partition identities come from?

- Combinatorics
- Representation theory of affine Lie algebras (often more complicated identities): Lepowsky–Wilson 1984, Capparelli 1992, Meurman–Primc 1999, Siladić 2002, Nandi 2014, D.–Konan 2020-2023+
- A few examples from other fields:
    - commutative algebra and algebraic geometry: Bruschek–Mourtada–Schepers 2011, Afsharijoo 2020, El Manssour–Pogudin 2021

Where do partition identities come from?

- Combinatorics
- Representation theory of affine Lie algebras (often more complicated identities): Lepowsky–Wilson 1984, Capparelli 1992, Meurman–Primc 1999, Siladić 2002, Nandi 2014, D.–Konan 2020-2023+
- A few examples from other fields:
  - commutative algebra and algebraic geometry: Bruschek–Mourtada–Schepers 2011, Afsharijoo 2020, El Manssour–Pogudin 2021
  - mathematical physics: Andrews–Baxter 1981, 1984

Where do partition identities come from?

- Combinatorics
- Representation theory of affine Lie algebras (often more complicated identities): Lepowsky–Wilson 1984, Capparelli 1992, Meurman–Primc 1999, Siladić 2002, Nandi 2014, D.–Konan 2020-2023+
- A few examples from other fields:
  - commutative algebra and algebraic geometry: Bruschek–Mourtada–Schepers 2011, Afsharijoo 2020, El Manssour–Pogudin 2021
  - mathematical physics: Andrews–Baxter 1981, 1984

In representation theory, partition identities come from giving two different expressions for

$$\mathrm{ch}(V) = \sum_{\mu} \dim(V_\mu) e^{\mu},$$

the *character* of a Lie algebra representation $V = \bigoplus_\mu V_\mu$ (which is a vector space). Here the μ's are so-called *roots* and can be written as linear combinations of simple roots $\alpha_0, \ldots, \alpha_n$.

Where do partition identities come from?

- Combinatorics
- Representation theory of affine Lie algebras (often more complicated identities): Lepowsky–Wilson 1984, Capparelli 1992, Meurman–Primc 1999, Siladić 2002, Nandi 2014, D.–Konan 2020-2023+
- A few examples from other fields:
  - commutative algebra and algebraic geometry: Bruschek–Mourtada–Schepers 2011, Afsharijoo 2020, El Manssour–Pogudin 2021
  - mathematical physics: Andrews–Baxter 1981, 1984

In representation theory, partition identities come from giving two different expressions for

$$\mathrm{ch}(V) = \sum_{\mu} \dim(V_\mu) e^{\mu},$$

the *character* of a Lie algebra representation $V = \bigoplus_\mu V_\mu$ (which is a vector space). Here the μ's are so-called *roots* and can be written as linear combinations of simple roots $\alpha_0, \ldots, \alpha_n$.
The principal specialisation ($e^{-\alpha_i} \mapsto q$ for all i) gives an infinite product.

The Rogers–Ramanujan identities

Theorem (Rogers 1894, Rogers–Ramanujan 1919)

$$\sum_{n=0}^{\infty} \frac{q^{n^2}}{(q;q)_n} = \frac{1}{(q;q^5)_\infty (q^4;q^5)_\infty},$$

For every positive integer n, the number of partitions of n such that the difference between two consecutive parts is at least 2 is equal to the number of partitions of n into parts congruent to 1 or 4 modulo 5.

$$\sum_{n=0}^{\infty} \frac{q^{n^2+n}}{(q;q)_n} = \frac{1}{(q^2;q^5)_\infty (q^3;q^5)_\infty},$$

For every positive integer n, the number of partitions of n such that the difference between two consecutive parts is at least 2 and the smallest part is > 1 is equal to the number of partitions of n into parts congruent to 2 or 3 modulo 5.

Representation theoretic interpretation

Lepowsky and Wilson 1984: representation theoretic interpretation

$$(-q;q)_\infty \sum_{n=0}^{\infty} \frac{q^{n^2}}{(q;q)_n} = (-q;q)_\infty \frac{1}{(q;q^5)_\infty (q^4;q^5)_\infty}$$

Obtained by giving two different formulations for the principal specialisation of $e^{-(\Lambda_0+2\Lambda_1)}\mathrm{ch}L(\Lambda_0+2\Lambda_1)$, where $L(\Lambda_0+2\Lambda_1)$ is the irreducible highest weight $A_1^{(1)}$-module of level 3 with highest weight $\Lambda_0+2\Lambda_1$.

RHS: principal specialisation of the character formula

LHS: comes from the construction of a basis of $L(\Lambda_0+2\Lambda_1)$

Representation theoretic interpretation

Lepowsky and Wilson 1984: representation theoretic interpretation

$$(-q;q)_\infty \sum_{n=0}^{\infty} \frac{q^{n^2}}{(q;q)_n} = (-q;q)_\infty \frac{1}{(q;q^5)_\infty (q^4;q^5)_\infty}$$

LHS: comes from the construction of a basis of $L(\Lambda_0 + 2\Lambda_1)$.

Very rough idea:

- Start with a spanning set of $L(\Lambda_0 + 2\Lambda_1)$: here, monomials of the form $Z_1^{f_1} \dots Z_s^{f_s}$ for $s, f_1, \dots, f_s \in \mathbb{N}_{\geq 0}$.
- Using Lie theory, reduce this spanning set: here, one should remove all monomials containing Z_j^2 or $Z_j Z_{j+1}$ for any j.
- Show that the obtained set is a basis of the representation (difficult).

Goals

- Prove partition identities

Goals

- Prove partition identities
  - for the sake of combinatorics

Goals

- Prove partition identities
  - for the sake of combinatorics
  - to prove that conjectured bases of representations are indeed bases (using an approach similar to Lepowsky and Wilson's, other partition identities have been conjectured)

Goals

- Prove partition identities
  - for the sake of combinatorics
  - to prove that conjectured bases of representations are indeed bases (using an approach similar to Lepowsky and Wilson's, other partition identities have been conjectured)
  - to compute asymptotics

Goals

- Prove partition identities
  - for the sake of combinatorics
  - to prove that conjectured bases of representations are indeed bases (using an approach similar to Lepowsky and Wilson's, other partition identities have been conjectured)
  - to compute asymptotics
- Refine partition identities (i.e. add more variables to the generating functions, or colours on the parts of the partitions)

Goals

- Prove partition identities
  - for the sake of combinatorics
  - to prove that conjectured bases of representations are indeed bases (using an approach similar to Lepowsky and Wilson's, other partition identities have been conjectured)
  - to compute asymptotics
- Refine partition identities (i.e. add more variables to the generating functions, or colours on the parts of the partitions)
  - to have more precise combinatorial information (and be able to do some particular type of proofs, see later)

Goals

- Prove partition identities
  - for the sake of combinatorics
  - to prove that conjectured bases of representations are indeed bases (using an approach similar to Lepowsky and Wilson's, other partition identities have been conjectured)
  - to compute asymptotics
- Refine partition identities (i.e. add more variables to the generating functions, or colours on the parts of the partitions)
  - to have more precise combinatorial information (and be able to do some particular type of proofs, see later)
  - it can give clues about what a bijection could be

Goals

- Prove partition identities
  - for the sake of combinatorics
  - to prove that conjectured bases of representations are indeed bases (using an approach similar to Lepowsky and Wilson's, other partition identities have been conjectured)
  - to compute asymptotics
- Refine partition identities (i.e. add more variables to the generating functions, or colours on the parts of the partitions)
  - to have more precise combinatorial information (and be able to do some particular type of proofs, see later)
  - it can give clues about what a bijection could be
  - sometimes, the colours/new variables also have a representation theoretic meaning (D.–Konan 2020)

Goals

- Prove partition identities
  - for the sake of combinatorics
  - to prove that conjectured bases of representations are indeed bases (using an approach similar to Lepowsky and Wilson's, other partition identities have been conjectured)
  - to compute asymptotics
- Refine partition identities (i.e. add more variables to the generating functions, or colours on the parts of the partitions)
  - to have more precise combinatorial information (and be able to do some particular type of proofs, see later)
  - it can give clues about what a bijection could be
  - sometimes, the colours/new variables also have a representation theoretic meaning (D.–Konan 2020)

There are several ways to do this, but here we focus on those involving functional equations and computer algebra.

Outline

1. Introduction
   - Partitions and their generating functions
   - Partition identities
   - Motivation and goals
2. Preliminary step: obtaining a system of recurrences
   - Without colours
   - With colours (refinement)
3. First method: finding an iterable relation
4. Second method: the "back-and-forth" technique
   - From recurrences to q-difference equations
   - Back and forth between recurrences and q-difference equations

Outline

1 Introduction
- Partitions and their generating functions
- Partition identities
- Motivation and goals

2 Preliminary step: obtaining a system of recurrences
- Without colours
- With colours (refinement)

3 First method: finding an iterable relation

4 Second method: the "back-and-forth" technique
- From recurrences to q-difference equations
- Back and forth between recurrences and q-difference equations

Schur's theorem

Theorem (Schur 1926)

For any positive integer n, let $A(n)$ denote the number of partitions of n into distinct parts congruent to 1 or 2 modulo 3 and $B(n)$ denote the number of partitions $(\lambda_1, \dots, \lambda_m)$ of n such that

$$\lambda_i - \lambda_{i+1} \geq \begin{cases} 3 & \text{if } \lambda_{i+1} \equiv 1, 2 \mod 3, \\ 4 & \text{if } \lambda_{i+1} \equiv 0 \mod 3. \end{cases}$$

Then $A(n) = B(n)$.

Example

The partitions counted by $A(10)$ are (10), $(8, 2)$, $(7, 2, 1)$ and $(5, 4, 1)$.
The partitions counted by $B(10)$ are (10), $(9, 1)$, $(8, 2)$ and $(7, 3)$.
There are 4 partitions in both cases.

Andrews' idea (1968)

Let $B_k(n)$ be the number of partitions with difference conditions (difference at least 3 between consecutive parts, no consecutive multiples of 3) *such that the largest part is at most k*, and define

$$G_k(q) := \sum_{n \in \mathbb{N}} B_k(n) q^n.$$

By a combinatorial reasoning based on removing the first part of the partitions, one can prove the recurrences:

$$\begin{aligned} G_{3m+1}(q) &= G_{3m}(q) + q^{3m+1} G_{3m-2}(q), \\ G_{3m+2}(q) &= G_{3m+1}(q) + q^{3m+2} G_{3m-1}(q), \\ G_{3m+3}(q) &= G_{3m+2}(q) + q^{3m+3} G_{3m-1}(q). \end{aligned}$$

Andrews' idea (1968)

Let $B_k(n)$ be the number of partitions with difference conditions (difference at least 3 between consecutive parts, no consecutive multiples of 3) *such that the largest part is at most* k, and define

$$G_k(q) := \sum_{n \in \mathbb{N}} B_k(n) q^n.$$

By a combinatorial reasoning based on removing the first part of the partitions, one can prove the recurrences:

$$\begin{aligned} G_{3m+1}(q) &= G_{3m}(q) + q^{3m+1} G_{3m-2}(q), \\ G_{3m+2}(q) &= G_{3m+1}(q) + q^{3m+2} G_{3m-1}(q), \\ G_{3m+3}(q) &= G_{3m+2}(q) + q^{3m+3} G_{3m-1}(q). \end{aligned}$$

Goal: show that $\lim_{k\to\infty} G_k(q) = (-q;q^3)_\infty(-q^2;q^3)_\infty$.

Outline

1 Introduction
- Partitions and their generating functions
- Partition identities
- Motivation and goals

2 Preliminary step: obtaining a system of recurrences
- Without colours
- With colours (refinement)

3 First method: finding an iterable relation

4 Second method: the "back-and-forth" technique
- From recurrences to q-difference equations
- Back and forth between recurrences and q-difference equations

Schur's theorem (coloured version)

Consider integers in three colours $ab < a < b$, with

$$1_a < 1_b < 2_{ab} < 2_a < 2_b < 3_{ab} < 3_a < 3_b < \cdots .$$

Theorem (Schur 1926, Alladi–Gordon 1993)

For any positive integer n, let $A(n,i,j)$ denote the number of partitions of n into i distinct parts coloured a and j distinct parts coloured b.
Let $B(n,i,j)$ denote the number of partitions $\lambda_1 + \cdots + \lambda_m$ of n, with i parts coloured a or ab and j parts coloured b or ab such that

$$\lambda_i - \lambda_{i+1} \geq \begin{cases} 2 \text{ if } color(\lambda_i) = ab \text{ or } color(\lambda_i) < color(\lambda_{i+1}), \\ 1 \text{ otherwise,} \end{cases}$$

Then for all n, i, j, $A(n,i,j) = B(n,i,j)$.

$k_{ab} \mapsto 3k-3$, $k_a \mapsto 3k-2$, $k_b \mapsto 3k-1$ recovers Schur's original theorem.

Goal

Define

$$G_{k_c}(q, a, b) := \sum_{n,i,j \in \mathbb{N}} B_{k_c}(n, i, j) q^n a^i b^j,$$

where $B_{k_c}(n, i, j)$ is the number of partitions counted by $B(n, i, j)$ such that the largest part is at most k_c, $c \in \{a, b, ab\}$.
We know that

$$\sum_{n,i,j \in \mathbb{N}} A(n, i, j) q^n a^i b^j = (-aq; q)_\infty (-bq; q)_\infty.$$

Goal

Define

$$G_{k_c}(q, a, b) := \sum_{n,i,j \in \mathbb{N}} B_{k_c}(n, i, j) q^n a^i b^j,$$

where $B_{k_c}(n, i, j)$ is the number of partitions counted by $B(n, i, j)$ such that the largest part is at most k_c, $c \in \{a, b, ab\}$.
We know that

$$\sum_{n,i,j \in \mathbb{N}} A(n, i, j) q^n a^i b^j = (-aq; q)_\infty(-bq; q)_\infty.$$

Goal: show that for $c \in \{a, b, ab\}$,

$$\lim_{k \to \infty} G_{k_c}(q, a, b) = \sum_{n,i,j \in \mathbb{N}} B(n, i, j) q^n a^i b^j = (-aq; q)_\infty(-bq; q)_\infty.$$

Obtaining recurrence equations

Reminder:

$$\lambda_i - \lambda_{i+1} \geq \begin{cases} 2 \text{ if } color(\lambda_i) = ab \text{ or } color(\lambda_i) < color(\lambda_{i+1}), \\ 1 \text{ otherwise,} \end{cases}$$

$$1_a < 1_b < 2_{ab} < 2_a < 2_b < 3_{ab} < 3_a < 3_b < \cdots .$$

We obtain the recurrences:

$$\begin{aligned} G_{k_a}(q; a, b) &= G_{k_{ab}}(q; a, b) + aq^k G_{(k-1)_a}(q; a, b), \\ G_{k_b}(q; a, b) &= G_{k_a}(q; a, b) + bq^k G_{(k-1)_b}(q; a, b), \\ G_{k_{ab}}(q; a, b) &= G_{(k-1)_b}(q; a, b) + abq^k G_{(k-2)_b}(q; a, b), \end{aligned}$$

and we look at small cases by hand to find initial conditions for $k = 1, 2, 3$.

All of this is automatic!

D. 2017

When the difference conditions are between consecutive parts and depend either on congruence conditions on the parts or on the parts' colours, then the recurrences for the G_k's can be computed automatically.

All of this is automatic!

D. 2017

When the difference conditions are between consecutive parts and depend either on congruence conditions on the parts or on the parts' colours, then the recurrences for the G_k's can be computed automatically.

Idea: encoding the difference conditions in a matrix and reading the recurrences from it.

All of this is automatic!

D. 2017

When the difference conditions are between consecutive parts and depend either on congruence conditions on the parts or on the parts' colours, then the recurrences for the G_k's can be computed automatically.

Idea: encoding the difference conditions in a matrix and reading the recurrences from it.

Implemented in Mathematica by Jakob Ablinger and Ali Uncu in their package `qFunctions` (2021).

Example (Primc's identity)

Partitions in four colours a, b, c, d, with the order

$$1_a < 1_b < 1_c < 1_d < 2_a < 2_b < 2_c < 2_d < \cdots ,$$

and difference conditions

$$P = \begin{array}{c c} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{pmatrix} 2 & 1 & 2 & 2 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 2 \end{pmatrix} \end{array}.$$

Let $G_k(q; a, b, c, d)$ (resp. $E_k(q; a, b, c, d)$) be the generating function for coloured partitions satisfying the difference conditions from matrix P, with largest part is at most (resp. equal to) k.
We have 4 equations of the following shape (one for each colour):

$$\begin{aligned} G^P_{k_d}(q; a, b, c, d) - G^P_{k_c}(q; a, b, c, d) &= E^P_{k_d}(q; a, b, c, d) \\ &= dq^k(E^P_{k_c}(q; a, b, c, d) + E^P_{k_a}(q; a, b, c, d) + G^P_{(k-1)_c}(q; a, b, c, d)). \end{aligned}$$

Outline

1. Introduction
   - Partitions and their generating functions
   - Partition identities
   - Motivation and goals
2. Preliminary step: obtaining a system of recurrences
   - Without colours
   - With colours (refinement)
3. First method: finding an iterable relation
4. Second method: the "back-and-forth" technique
   - From recurrences to q-difference equations
   - Back and forth between recurrences and q-difference equations

First method: finding an iterable relation

Start by experimenting with a computer (see Maple).

First method: finding an iterable relation

Start by experimenting with a computer (see Maple).

We want to show that for all k,

$$G_{(k+2)_{ab}}(q; a, b) = (1 + aq)(1 + bq)G_{k_b}(q; aq, bq).$$

First method: finding an iterable relation

Start by experimenting with a computer (see Maple).

We want to show that for all k,

$$G_{(k+2)_{ab}}(q; a, b) = (1 + aq)(1 + bq)G_{k_b}(q; aq, bq).$$

Then we can deduce that

$$\begin{aligned}
\sum_{n,i,j\in\mathbb{N}} B(n,i,j)q^n a^i b^j &= G_\infty(q; a, b) \\
&= (1 + aq)(1 + bq)G_\infty(q; aq, bq) \\
&= (1 + aq)(1 + bq)(1 + aq^2)(1 + bq^2)G_\infty(q; aq^2, bq^2) \\
&= \cdots \\
&= (-aq; q)_\infty(-bq; q)_\infty G_\infty(q; 0, 0) \\
&= (-aq; q)_\infty(-bq; q)_\infty \\
&= \sum_{n,i,j\in\mathbb{N}} A(n,i,j)q^n a^i b^j.
\end{aligned}$$

We uncouple our system of three recurrences (by hand or with a computer algebra system), and obtain:

$$\begin{aligned} G_{(k+2)_{ab}}(q;a,b) &= (1+aq^{k+1}+bq^{k+1})G_{(k+1)_{ab}}(q;a,b) \\ &\quad + abq^{k+2}(1-q^{k+1})G_{k_{ab}}(q;a,b), \\ G_{k_b}(q;aq,bq) &= (1+aq^{k+1}+bq^{k+1})G_{(k-1)_b}(q;aq,bq) \\ &\quad + abq^{k+2}(1-q^{k+1})G_{(k-2)_b}(q;aq,bq). \end{aligned}$$

We uncouple our system of three recurrences (by hand or with a computer algebra system), and obtain:

$$\begin{aligned} G_{(k+2)_{ab}}(q;a,b) &= (1+aq^{k+1}+bq^{k+1})G_{(k+1)_{ab}}(q;a,b) \\ &\quad + abq^{k+2}(1-q^{k+1})G_{k_{ab}}(q;a,b), \\ G_{k_b}(q;aq,bq) &= (1+aq^{k+1}+bq^{k+1})G_{(k-1)_b}(q;aq,bq) \\ &\quad + abq^{k+2}(1-q^{k+1})G_{(k-2)_b}(q;aq,bq). \end{aligned}$$

The recurrences are the same, so we only have to look at initial conditions:

$$\begin{aligned} &G_{2_{ab}}(q;a,b) = 1+aq+bq+abq^2 = (1+aq)(1+bq)G_{0_b}(q;aq,bq), \\ &G_{3_{ab}}(q;a,b) = (1+aq)(1+bq)G_{1_b}(q;aq,bq). \end{aligned}$$

We uncouple our system of three recurrences (by hand or with a computer algebra system), and obtain:

$$\begin{aligned} G_{(k+2)_{ab}}(q;a,b) &= (1+aq^{k+1}+bq^{k+1})G_{(k+1)_{ab}}(q;a,b) \\ &\quad + abq^{k+2}(1-q^{k+1})G_{k_{ab}}(q;a,b), \\ G_{k_b}(q;aq,bq) &= (1+aq^{k+1}+bq^{k+1})G_{(k-1)_b}(q;aq,bq) \\ &\quad + abq^{k+2}(1-q^{k+1})G_{(k-2)_b}(q;aq,bq). \end{aligned}$$

The recurrences are the same, so we only have to look at initial conditions:

$$\begin{aligned} G_{2_{ab}}(q;a,b) &= 1+aq+bq+abq^2 = (1+aq)(1+bq)G_{0_b}(q;aq,bq), \\ G_{3_{ab}}(q;a,b) &= (1+aq)(1+bq)G_{1_b}(q;aq,bq). \end{aligned}$$

Thus for all k,

$$G_{(k+2)_{ab}}(q;a,b) = (1+aq)(1+bq)G_{k_b}(q;aq,bq).$$

□

Other example: Siladić's identity (2002)

The generating function for partitions of n into coloured integers

$$1_{ab} < 1_a < 1_{b^2} < 1_b < 2_{ab} < 2_a < 3_{a^2} < 2_b < 3_{ab} < 3_a < 3_{b^2} < \cdots$$

with minimal differences between consecutive parts given by the matrix

$$A = \begin{array}{c} \\ a \\ b \\ ab \\ a^2 \\ b^2 \end{array} \begin{array}{c} \begin{array}{cccccccc} a_{odd} & b^2 & b_{odd} & ab_{even} & a_{even} & a^2 & b_{even} & ab_{odd} \end{array} \\ \begin{pmatrix} 2 & 2 & 2 & 1 & 2 & 2 & 2 & 2 \\ 1 & 2 & 2 & 1 & 1 & 1 & 2 & 1 \\ 2 & 3 & 3 & 2 & 2 & 2 & 2 & 2 \\ 4 & 4 & 4 & 3 & 3 & 4 & 3 & 4 \\ 2 & 4 & 4 & 3 & 3 & 2 & 3 & 2 \end{pmatrix} \end{array}$$

is $(-aq;q)_\infty(-bq;q)_\infty$.

The recurrences

For all $k \in \mathbb{N}^*$,

$$\begin{aligned}
G_{2k+1_{ab}}(a,b,q) &= G_{2k_b}(a,b,q) + abq^{2k+1}G_{2k-1_a}(a,b,q),\\
G_{2k+1_a}(a,b,q) &= G_{2k+1_{ab}}(a,b,q) + aq^{2k+1}G_{2k_{ab}}(a,b,q),\\
G_{2k+1_{b^2}}(a,b,q) &= G_{2k+1_a}(a,b,q) + b^2q^{2k+1}G_{2k-1_a}(a,b,q),\\
G_{2k+1_b}(a,b,q) &= G_{2k+1_{b^2}}(a,b,q) + bq^{2k+1}G_{2k_a}(a,b,q),\\
G_{2k+2_{ab}}(a,b,q) &= G_{2k+1_b} + abq^{2k+2}G_{2k_a} + ab^2q^{4k+2}G_{2k-1_a},\\
G_{2k+2_a}(a,b,q) &= G_{2k+2_{ab}} + aq^{2k+2}G_{2k_a} + abq^{4k+2}G_{2k-1_a},\\
G_{2k+3_{a^2}}(a,b,q) &= G_{2k+2_a} + a^2q^{2k+3}G_{2k_a} + a^2bq^{4k+3}G_{2k-1_a},\\
G_{2k+2_b}(a,b,q) &= G_{2k+3_{a^2}}(a,b,q) + bq^{2k+2}G_{2k+1_a}(a,b,q).
\end{aligned}$$

Proof (D. 2017)

Show that for all k,

$$G_{2k+1_{ab}}(q;a,b) = (1+aq)G_{2k_a}(q;b,aq),$$
$$G_{2k+1_{b^2}}(q;a,b) = (1+aq)G_{2k_b}(q;b,aq),$$
$$G_{2k+2_{ab}}(q;a,b) = (1+aq)G_{2k+1_a}(q;b,aq),$$
$$G_{2k+1_{a^2}}(q;a,b) = (1+aq)G_{2k-1_b}(q;b,aq).$$

Proof (D. 2017)

Show that for all k,

$$
\begin{aligned}
G_{2k+1_{ab}}(q;a,b) &= (1+aq)G_{2k_a}(q;b,aq),\\
G_{2k+1_{b^2}}(q;a,b) &= (1+aq)G_{2k_b}(q;b,aq),\\
G_{2k+2_{ab}}(q;a,b) &= (1+aq)G_{2k+1_a}(q;b,aq),\\
G_{2k+1_{a^2}}(q;a,b) &= (1+aq)G_{2k-1_b}(q;b,aq).
\end{aligned}
$$

Let k go to infinity and deduce

$$
\begin{aligned}
G_\infty(q;a,b) &= (1+aq)G_\infty(q;b,aq)\\
&= (1+aq)(1+bq)G_\infty(q;aq,bq)\\
&= (-aq;q)_\infty(-bq;q)_\infty.
\end{aligned}
$$

□

Summary of this method

Pros:

- The iterable relation can be guessed experimentally.
- The proof can be automated (via uncoupling of the system of recurrences).

Cons:

- Requires at least one colour variable (often more!).
- It is not always easy (or even possible?) to introduce the right colour variables.

Outline

1 Introduction
- Partitions and their generating functions
- Partition identities
- Motivation and goals

2 Preliminary step: obtaining a system of recurrences
- Without colours
- With colours (refinement)

3 First method: finding an iterable relation

4 Second method: the "back-and-forth" technique
- From recurrences to q-difference equations
- Back and forth between recurrences and q-difference equations

Outline

1. Introduction
   - Partitions and their generating functions
   - Partition identities
   - Motivation and goals
2. Preliminary step: obtaining a system of recurrences
   - Without colours
   - With colours (refinement)
3. First method: finding an iterable relation
4. Second method: the "back-and-forth" technique
   - From recurrences to q-difference equations
   - Back and forth between recurrences and q-difference equations

Back to Schur's theorem

Andrews had found the recurrences

$$
\begin{aligned}
G_{3m+1}(q) &= G_{3m}(q) + q^{3m+1} G_{3m-2}(q),\\
G_{3m+2}(q) &= G_{3m+1}(q) + q^{3m+2} G_{3m-1}(q),\\
G_{3m+3}(q) &= G_{3m+2}(q) + q^{3m+3} G_{3m-1}(q).
\end{aligned}
$$

Back to Schur's theorem

Andrews had found the recurrences

$$\begin{aligned} G_{3m+1}(q) &= G_{3m}(q) + q^{3m+1} G_{3m-2}(q), \\ G_{3m+2}(q) &= G_{3m+1}(q) + q^{3m+2} G_{3m-1}(q), \\ G_{3m+3}(q) &= G_{3m+2}(q) + q^{3m+3} G_{3m-1}(q). \end{aligned}$$

Uncouple them:

$$G_{3m+2}(q) = \left(1 + q^{3m+1} + q^{3m+2}\right) G_{3m-1}(q) + q^{3m}(1 - q^{3m}) G_{3m-4}(q).$$

Back to Schur's theorem

Andrews had found the recurrences

$$\begin{aligned} G_{3m+1}(q) &= G_{3m}(q) + q^{3m+1} G_{3m-2}(q), \\ G_{3m+2}(q) &= G_{3m+1}(q) + q^{3m+2} G_{3m-1}(q), \\ G_{3m+3}(q) &= G_{3m+2}(q) + q^{3m+3} G_{3m-1}(q). \end{aligned}$$

Uncouple them:

$$G_{3m+2}(q) = \left(1 + q^{3m+1} + q^{3m+2}\right) G_{3m-1}(q) + q^{3m}(1 - q^{3m}) G_{3m-4}(q).$$

Define

$$s_m(q) = \frac{1}{(q^3;q^3)_m} \times G_{3m-1}(q).$$

We have

$$(1 - q^{3m+3}) s_{m+1}(q) = \left(1 + q^{3m+1} + q^{3m+2}\right) s_m(q) + q^{3m} s_{m-1}(q).$$

Define

$$f(x,q) := \sum_{n\geq 0} s_n(q)x^n.$$

The equation

$$(1-q^{3m+3})s_{m+1}(q) = \left(1+q^{3m+1}+q^{3m+2}\right)s_m(q) + q^{3m}s_{m-1}(q)$$

becomes

$$(1-x)f(x;q) = (1+xq)(1+xq^2)f(xq^3;q).$$

Hence

$$f(x;q) = \frac{(-xq;q^3)_\infty(-xq^2;q^3)_\infty}{(x;q^3)_\infty}.$$

Lemma

$$\lim_{x\to 1^-}(1-x)\sum_{n\geq 0} u_n x^n = \lim_{n\to\infty} u_n.$$

$$\begin{aligned}
\lim_{m\to\infty} G_{3m+2}(q) &= (q^3;q^3)_\infty \lim_{m\to\infty} s_m(q) \\
&= (q^3;q^3)_\infty \lim_{x\to 1^-}(1-x)f(x;q) \\
&= (q^3;q^3)_\infty \lim_{x\to 1^-}(1-x)\frac{(-xq;q^3)_\infty(-xq^2;q^3)_\infty}{(x;q^3)_\infty} \\
&= (q^3;q^3)_\infty \frac{(-q;q^3)_\infty(-q^2;q^3)_\infty}{(q^3;q^3)_\infty} \\
&= (-q;q^3)_\infty(-q^2;q^3)_\infty
\end{aligned}$$

$\square$

Outline

1. Introduction
   - Partitions and their generating functions
   - Partition identities
   - Motivation and goals
2. Preliminary step: obtaining a system of recurrences
   - Without colours
   - With colours (refinement)
3. First method: finding an iterable relation
4. Second method: the "back-and-forth" technique
   - From recurrences to q-difference equations
   - Back and forth between recurrences and q-difference equations

Primc's identity (conjecture Primc 1999, D.–Lovejoy 2018, D. 2020)

Partitions in four colours a, b, c, d, with the order

$$1_a < 1_b < 1_c < 1_d < 2_a < 2_b < 2_c < 2_d < \cdots ,$$

and difference conditions

$$P = \begin{array}{c c} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{pmatrix} 2 & 1 & 2 & 2 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 2 \end{pmatrix} \end{array}.$$

Let $A(n; k, \ell, m)$ denote the number of partitions satisfying the difference conditions of matrix P, with k parts coloured a, ℓ parts coloured c and m parts coloured d. Then

$$\sum_{n,k,\ell,m \geq 0} A(n; k, \ell, m) q^n a^k c^\ell d^m = \frac{(-aq; q^2)_\infty (-dq; q^2)_\infty}{(q; q)_\infty (cq; q^2)_\infty}.$$

Recall that $G_k = G_k(q; a, c, d)$ denotes the g.f. for partitions satisfying the difference conditions and with largest part at most k.

Recall that $G_k = G_k(q; a, c, d)$ denotes the g.f. for partitions satisfying the difference conditions and with largest part at most k.
We uncouple the system of recurrences found before and obtain

$$\begin{aligned}(1 - cq^k)G_{k_d} &= \frac{1 - cq^{2k}}{1 - q^k} G_{(k-1)_d} \\ &+ \frac{aq^k + dq^k + adq^{2k}}{1 - q^{k-1}} G_{(k-2)_d} + \frac{adq^{2k-1}}{1 - q^{k-2}} G_{(k-3)_d}.\end{aligned}$$

Recall that $G_k = G_k(q; a, c, d)$ denotes the g.f. for partitions satisfying the difference conditions and with largest part at most k.
We uncouple the system of recurrences found before and obtain

$$\begin{aligned}(1 - cq^k)G_{k_d} &= \frac{1 - cq^{2k}}{1 - q^k} G_{(k-1)_d} \\ &+ \frac{aq^k + dq^k + adq^{2k}}{1 - q^{k-1}} G_{(k-2)_d} + \frac{adq^{2k-1}}{1 - q^{k-2}} G_{(k-3)_d}.\end{aligned}$$

Let

$$H_k = H_k(q; a, c, d) := \frac{G_{k_d}(q; a, c, d)}{1 - q^{k+1}}.$$

Then

$$\begin{aligned}(1 - cq^k - q^{k+1} + cq^{2k+1})H_k &= (1 - cq^{2k})H_{k-1} \\ &+ (aq^k + dq^k + adq^{2k})H_{k-2} + adq^{2k-1}H_{k-3}.\end{aligned} \tag{1}$$

Define

$$f(x) := \sum_{k\geq 0} H_{k-1}x^k,$$

and convert the recurrence (1) into a q-difference equation on f:

$$(1-x)f(x) = (1+\frac{c}{q}+ax^2q+dx^2q)f(xq)-(1+xq)(\frac{c}{q}-adx^2q^2)f(xq^2).$$

Define

$$f(x) := \sum_{k \geq 0} H_{k-1} x^k,$$

and convert the recurrence (1) into a q-difference equation on f:

$$(1-x)f(x) = (1 + \frac{c}{q} + ax^2q + dx^2q)f(xq) - (1+xq)(\frac{c}{q} - adx^2q^2)f(xq^2).$$

Define

$$g(x) := \frac{f(x)}{\prod_{k \geq 0}(1+xq^k)}.$$

We obtain:

$$(1-x^2)g(x) = (1 + \frac{c}{q} + ax^2q + dx^2q)g(xq) - (\frac{c}{q} - adx^2q^2)g(xq^2).$$

Go back to recurrence equations again: define $(u_n)_{n\in\mathbb{N}}$ as

$$\sum_{n\geq 0} u_n x^n := g(x).$$

Then (u_n) satisfies

$$u_n = \frac{\left(1+aq^{n-1}\right)\left(1+dq^{n-1}\right)}{\left(1-q^n\right)\left(1-cq^{n-1}\right)} u_{n-2},$$

and the initial conditions

$$u_0 = 1, u_1 = 0.$$

Thus for all $n \geq 0$, we have

$$u_{2n} = \frac{(-aq;q^2)_n(-dq;q^2)_n}{(q^2;q^2)_n(cq;q^2)_n},$$

$$u_{2n+1} = 0.$$

Reminder

$$\lim_{x\to 1^-}(1-x)\sum_{n\geq 0} u_n x^n = \lim_{n\to\infty} u_n.$$

$$\begin{aligned}
\lim_{k\to\infty} G_k(q;a,c,d) &= \lim_{x\to 1^-}(1-x)f(x)\\
&= \lim_{x\to 1^-} g(x)\prod_{k\geq 0}(1+xq^k)\\
&= (-q;q)_\infty \lim_{x\to 1^-}(1-x^2)\sum_{n\geq 0} u_{2n}x^{2n}\\
&= (-q;q)_\infty \lim_{n\to\infty} u_{2n}\\
&= \frac{(-q;q)_\infty(-aq;q^2)_\infty(-dq;q^2)_\infty}{(q^2;q^2)_\infty(cq;q^2)_\infty}\\
&= \frac{(-aq;q^2)_\infty(-dq;q^2)_\infty}{(q;q)_\infty(cq;q^2)_\infty}.
\end{aligned}$$

Exact expression for $G_k(q; a, b, c, d)$

It is also possible to track back all the changes of unknown functions to find exact expressions, and keep track of the colour b.

Theorem (Finite version of Primc's identity (D. 2020))

We have, for every positive integer k,

$$G_k(q; a, b, c, d) = \left(1 - bq^{k+1}\right) \sum_{j=0}^{k+1} \frac{v_j(a, b, c, d) q^{\binom{k+1-j}{2}}}{(q; q)_{k+1-j}},$$

where for all $n \geq 0$,

$$v_{2n}(a, b, c, d) = (1 - b) \sum_{\ell=0}^{n} \frac{(-aq^{2\ell+1}; q^2)_{n-\ell}(-dq^{2\ell+1}; q^2)_{n-\ell}}{(bq^{2\ell}; q^2)_{n-\ell+1}(cq^{2\ell+1}; q^2)_{n-\ell}} \frac{q^{2\ell}}{(q; q)_{2\ell}},$$

$$v_{2n+1}(a, b, c, d) = (b-1) \sum_{\ell=0}^{n} \frac{(-aq^{2\ell+2}; q^2)_{n-\ell}(-dq^{2\ell+2}; q^2)_{n-\ell}}{(bq^{2\ell+1}; q^2)_{n-\ell+1}(cq^{2\ell+2}; q^2)_{n-\ell}} \frac{q^{2\ell+1}}{(q; q)_{2\ell+1}}.$$

Applications of the "back-and-forth" method

- Several generalisations of Schur's theorem (D. 2014-2018) : n steps of "back-and-forth" starting from the equation

$$\prod_{j=0}^{r-1}\left(1-dxq^{2^j}\right) f_1^r(x) = f_1^r(xq^N)$$
$$+\sum_{j=1}^{r}\left(\sum_{m=0}^{r-j} d^m \sum_{\substack{\alpha<2^r\\ w(\alpha)=j+m}} xq^{\alpha}\left((-x)^{m-1}\begin{bmatrix} j+m-1\\ m-1\end{bmatrix}_{q^N}\right.\right.$$
$$\left.\left.+(-x)^m\begin{bmatrix} j+m\\ m\end{bmatrix}_{q^N}\right)\right)\prod_{h=1}^{j-1}\left(1-xq^{hN}\right) f_1^r\left(xq^{jN}\right).$$

Applications of the "back-and-forth" method

- Bringmann–Jennnings-Shaffer–Mahlburg 2019: proof of the mod 12 Kanade–Russell conjectures (partition identities conjectured via computer search)
- Takigiku–Tsuchioka 2019: proof of the Nandi conjecture (partition identity coming from Lie algebras)
- D. 2020: proof of Capparelli's and Primc's identities (both coming from Lie algebras), and bijection between them (thanks to the colours and the similarity in the "back-and-forth" proofs)
- Tsuchioka 2022: proof of a Fibonacci variant of the Rogers–Ramanujan identities (coming from perfect crystals)

Questions

- Can we characterise the set of equations on which each of the two methods is applicable? (neither method works on all identities, for example the Kanade–Russell mod 9 identities)
- Can we automate and implement the "back-and-forth" method?

Questions

- Can we characterise the set of equations on which each of the two methods is applicable? (neither method works on all identities, for example the Kanade–Russell mod 9 identities)
- Can we automate and implement the "back-and-forth" method?

Thank you very much!