Systematic counting of pattern-avoiding partitions and some new partition identities

Rutgers University, Experimental Math Seminar, November 2019

Matthew C. Russell, **Mingjia Yang***, Doron Zeilberger

Rutgers University

Quick introduction to integer partitions

* A partition of a positive integer n is a finite nonincreasing sequence of positive integers $\lambda_1, \lambda_2, ..., \lambda_k$ whose sum is equal to n. For example, $(4, 4, 2, 1)$ is a partition of 11.

Quick introduction to integer partitions

- A partition of a positive integer n is a finite nonincreasing sequence of positive integers $\lambda_1, \lambda_2, ..., \lambda_k$ whose sum is equal to n. For example, $(4, 4, 2, 1)$ is a partition of 11.

- The (ordinary) generating function of a sequence $\{a_n\}_{n \geq 0}$ is a power series whose coefficients are the terms in the sequence (lining up in order). That is, $\sum_{n=0}^{\infty} a_n q^n$ is the generating function for the sequence $\{a_n\}_{n \geq 0}$.

  For example, let a_n be the number of partitions of n into parts all equal to a positive integer i. Then the generating function of $\{a_n\}_{n \geq 0}$ is:

$$\frac{1}{1-q^i} = 1 + q^i + q^{2i} + ...$$

Some fascinating partition identities

q-Pochhammer Symbol: $(a;q)_n := \prod_{j=0}^{n-1}(1-aq^j)$

$$(a;q)_\infty := \prod_{j\geq 0}(1-aq^j)$$

Some fascinating partition identities

q-Pochhammer Symbol: $(a;q)_n := \prod_{j=0}^{n-1}(1-aq^j)$

$(a;q)_\infty := \prod_{j\geq 0}(1-aq^j)$

- Euler's Odd Distinct identity (1748):

$$\prod_{all\ i}(1+q^i) = \prod_{all\ i}\frac{1-q^{2i}}{1-q^i} = \prod_{i\ odd}\frac{1}{1-q^i} = \frac{1}{(q;q^2)_\infty}$$

distinct parts $\rightarrow$ odd parts

Some fascinating partition identities

q-Pochhammer Symbol: $(a;q)_n := \prod_{j=0}^{n-1}(1-aq^j)$

$$(a;q)_\infty := \prod_{j\geq 0}(1-aq^j)$$

* Euler's Odd Distinct identity (1748):

$$\prod_{all\ i}(1+q^i) = \prod_{all\ i}\frac{1-q^{2i}}{1-q^i} = \prod_{i\ odd}\frac{1}{1-q^i} = \frac{1}{(q;q^2)_\infty}$$

distinct parts $\rightarrow$ odd parts

* Rogers-Ramanujan identities (1894):

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

adjacent parts differ by at least 2 $\rightarrow$ parts 1 or 4 mod 5

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

adjacent parts differ by at least 2, smallest part at least 2 $\rightarrow$ parts 2 or 3 mod 5

Ramanujan's congruences (a digression)

$p(n)$: number of partitions of n.

Ramanujan's congruences (a digression)

$p(n)$: number of partitions of n.

$$\begin{aligned} p(5k+4) &\equiv 0 \pmod{5} \\ p(7k+5) &\equiv 0 \pmod{7} \\ p(11k+6) &\equiv 0 \pmod{11} \end{aligned}$$

Efficient algorithms for counting pattern-avoiding partitions

Motivation and background

$p(n)$: number of partitions of n.
$P(n, m)$: number of partitions of n with the largest part m.

$$p(n) = \sum_{m=1}^{n} P(n, m).$$

$$P(n, m) = \sum_{m'=1}^{m} P(n - m, m') \quad , \quad n \geq m \geq 1$$

$$\Rightarrow P(n, m) = P(n - 1, m - 1) + P(n - m, m).$$

Motivation and background

$p(n)$: number of partitions of n.
$P(n, m)$: number of partitions of n with the largest part m.

$$p(n) = \sum_{m=1}^{n} P(n, m).$$

$$P(n, m) = \sum_{m'=1}^{m} P(n-m, m') \quad , \quad n \geq m \geq 1$$

$$\Rightarrow P(n, m) = P(n-1, m-1) + P(n-m, m).$$

The last equation gives us an efficient way (quadratic in time and memory) to compute a table for $p(n)$.

Motivation and background

$p(n)$: number of partitions of n.
$P(n, m)$: number of partitions of n with the largest part m.

$$p(n) = \sum_{m=1}^{n} P(n, m).$$

$$P(n, m) = \sum_{m'=1}^{m} P(n-m, m') \quad , \quad n \geq m \geq 1$$

$$\Rightarrow P(n, m) = P(n-1, m-1) + P(n-m, m).$$

The last equation gives us an efficient way (quadratic in time and memory) to compute a table for $p(n)$. But what if we want to count not just any partition efficiently, but partitions with some restrictions? What if we only want to count partitions whose adjacent parts differ by at least 2 (Rogers-Ramanujan)?

Definition

- A **pattern** is a list $a = [a_1, a_2, \ldots, a_r]$ $(r \geq 1)$ of non-negative integers.

Definition

* A **pattern** is a list $a = [a_1, a_2, \ldots, a_r]$ $(r \geq 1)$ of non-negative integers.
* A partition $\lambda = (\lambda_1, \ldots, \lambda_k)$ **contains** the pattern $a = [a_1, a_2, \ldots, a_r]$ if there exists $1 \leq i \leq k - r$ such that: $\lambda_i - \lambda_{i+1} = a_1, \quad \lambda_{i+1} - \lambda_{i+2} = a_2, \ldots, \lambda_{i+r-1} - \lambda_{i+r} = a_r.$

Definition

- A **pattern** is a list $a = [a_1, a_2, \ldots, a_r]$ $(r \geq 1)$ of non-negative integers.
- A partition $\lambda = (\lambda_1, \ldots, \lambda_k)$ **contains** the pattern $a = [a_1, a_2, \ldots, a_r]$ if there exists $1 \leq i \leq k - r$ such that: $\lambda_i - \lambda_{i+1} = a_1, \quad \lambda_{i+1} - \lambda_{i+2} = a_2, \ldots, \lambda_{i+r-1} - \lambda_{i+r} = a_r.$
- For example, $(7, 6, 5, 4, 4)$ contains the pattern [1], the pattern [0], the pattern $[1, 1]$, the pattern $[1, 0]$, and so on.

Definition

* A **pattern** is a list $a = [a_1, a_2, \ldots, a_r]$ $(r \geq 1)$ of non-negative integers.
* A partition $\lambda = (\lambda_1, \ldots, \lambda_k)$ **contains** the pattern $a = [a_1, a_2, \ldots, a_r]$ if there exists $1 \leq i \leq k - r$ such that: $\lambda_i - \lambda_{i+1} = a_1, \quad \lambda_{i+1} - \lambda_{i+2} = a_2, \ldots, \lambda_{i+r-1} - \lambda_{i+r} = a_r.$
* For example, $(7, 6, 5, 4, 4)$ contains the pattern [1], the pattern [0], the pattern $[1, 1]$, the pattern $[1, 0]$, and so on.
* A partition λ **avoids** (globally) the pattern if it does **not** contain the pattern.

Definition

* A **pattern** is a list $a = [a_1, a_2, \ldots, a_r]$ $(r \geq 1)$ of non-negative integers.
* A partition $\lambda = (\lambda_1, \ldots, \lambda_k)$ **contains** the pattern $a = [a_1, a_2, \ldots, a_r]$ if there exists $1 \leq i \leq k - r$ such that: $\lambda_i - \lambda_{i+1} = a_1, \quad \lambda_{i+1} - \lambda_{i+2} = a_2, \ldots, \lambda_{i+r-1} - \lambda_{i+r} = a_r$.
* For example, $(7, 6, 5, 4, 4)$ contains the pattern [1], the pattern [0], the pattern $[1, 1]$, the pattern $[1, 0]$, and so on.
* A partition λ **avoids** (globally) the pattern if it does **not** contain the pattern.
* A partition λ **avoids** (globally) the set of patterns A, if it avoids *every* pattern in A. For example, partitions whose adjacent parts differ by at least 2 is equivalent to partitions that avoid $\{[0], [1]\}$.

Definition

- A **pattern** is a list $a = [a_1, a_2, \ldots, a_r]$ $(r \geq 1)$ of non-negative integers.
- A partition $\lambda = (\lambda_1, \ldots, \lambda_k)$ **contains** the pattern $a = [a_1, a_2, \ldots, a_r]$ if there exists $1 \leq i \leq k - r$ such that: $\lambda_i - \lambda_{i+1} = a_1, \quad \lambda_{i+1} - \lambda_{i+2} = a_2, \ldots, \lambda_{i+r-1} - \lambda_{i+r} = a_r.$
- For example, $(7, 6, 5, 4, 4)$ contains the pattern $[1]$, the pattern $[0]$, the pattern $[1, 1]$, the pattern $[1, 0]$, and so on.
- A partition λ **avoids** (globally) the pattern if it does **not** contain the pattern.
- A partition λ **avoids** (globally) the set of patterns A, if it avoids *every* pattern in A. For example, partitions whose adjacent parts differ by at least 2 is equivalent to partitions that avoid $\{[0], [1]\}$.
- A partition λ contains a pattern *at the beginning* if we begin from the largest part of the partition and the pattern immediately appears. For example, (7,6,5,4,4) contains the pattern $[1, 1]$ at the beginning.

Idea of our initial algorithm

* $P_{[1,1,1]}(n, m)$: number of partitions of n with largest part m, avoiding $[1, 1, 1]$.

Idea of our initial algorithm

- $P_{[1,1,1]}(n,m)$: number of partitions of n with largest part m, avoiding $[1,1,1]$.
- $P'_{[1,1,1]}(n,m)$: the number of partitions of n with largest part m, avoiding the pattern $[1,1,1]$, and in addition, avoiding the pattern $[1,1]$ at the very beginning.

Idea of our initial algorithm

- $P_{[1,1,1]}(n,m)$: number of partitions of n with largest part m, avoiding $[1,1,1]$.
- $P'_{[1,1,1]}(n,m)$: the number of partitions of n with largest part m, avoiding the pattern $[1,1,1]$, and in addition, avoiding the pattern $[1,1]$ at the very beginning.
- $P''_{[1,1,1]}(n,m)$: number of partitions of n with largest part m, avoiding $[1,1,1]$, and *in addition*, avoiding the pattern $[1]$ at the beginning.

Idea of our initial algorithm

- $P_{[1,1,1]}(n, m)$: number of partitions of n with largest part m, avoiding $[1, 1, 1]$.
- $P'_{[1,1,1]}(n, m)$: the number of partitions of n with largest part m, avoiding the pattern $[1, 1, 1]$, and in addition, avoiding the pattern $[1, 1]$ at the very beginning.
- $P''_{[1,1,1]}(n, m)$: number of partitions of n with largest part m, avoiding $[1, 1, 1]$, and *in addition*, avoiding the pattern $[1]$ at the beginning.

$$P_{[1,1,1]}(n, m) = \sum_{\substack{1 \le m' \le m \\ m' \ne m-1}} P_{[1,1,1]}(n - m, m') + P'_{[1,1,1]}(n - m, m - 1)$$

Idea of our initial algorithm

* $P_{[1,1,1]}(n,m)$: number of partitions of n with largest part m, avoiding $[1,1,1]$.
* $P'_{[1,1,1]}(n,m)$: the number of partitions of n with largest part m, avoiding the pattern $[1,1,1]$, and in addition, avoiding the pattern $[1,1]$ at the very beginning.
* $P''_{[1,1,1]}(n,m)$: number of partitions of n with largest part m, avoiding $[1,1,1]$, and *in addition*, avoiding the pattern $[1]$ at the beginning.

$$P_{[1,1,1]}(n,m) = \sum_{\substack{1 \le m' \le m \\ m' \neq m-1}} P_{[1,1,1]}(n-m,m') + P'_{[1,1,1]}(n-m,m-1)$$

$$P'_{[1,1,1]}(n,m) = \sum_{\substack{1 \le m' \le m \\ m' \neq m-1}} P_{[1,1,1]}(n-m,m') + P''_{[1,1,1]}(n-m,m-1)$$

Idea of our initial algorithm

* $P_{[1,1,1]}(n,m)$: number of partitions of n with largest part m, avoiding $[1,1,1]$.
* $P'_{[1,1,1]}(n,m)$: the number of partitions of n with largest part m, avoiding the pattern $[1,1,1]$, and in addition, avoiding the pattern $[1,1]$ at the very beginning.
* $P''_{[1,1,1]}(n,m)$: number of partitions of n with largest part m, avoiding $[1,1,1]$, and *in addition*, avoiding the pattern $[1]$ at the beginning.

$$P_{[1,1,1]}(n,m) = \sum_{\substack{1 \le m' \le m \\ m' \ne m-1}} P_{[1,1,1]}(n-m,m') + P'_{[1,1,1]}(n-m,m-1)$$

$$P'_{[1,1,1]}(n,m) = \sum_{\substack{1 \le m' \le m \\ m' \ne m-1}} P_{[1,1,1]}(n-m,m') + P''_{[1,1,1]}(n-m,m-1)$$

$$P''_{[1,1,1]}(n,m) = \sum_{\substack{1 \le m' \le m \\ m' \ne m-1}} P_{[1,1,1]}(n-m,m')$$

Idea of our initial algorithm

- $P_{[1,1,1]}(n, m)$: number of partitions of n with largest part m, avoiding $[1, 1, 1]$.
- $P'_{[1,1,1]}(n, m)$: the number of partitions of n with largest part m, avoiding the pattern $[1, 1, 1]$, and in addition, avoiding the pattern $[1, 1]$ at the very beginning.
- $P''_{[1,1,1]}(n, m)$: number of partitions of n with largest part m, avoiding $[1, 1, 1]$, and *in addition*, avoiding the pattern $[1]$ at the beginning.

$$P_{[1,1,1]}(n, m) = \sum_{\substack{1 \le m' \le m \\ m' \ne m-1}} P_{[1,1,1]}(n - m, m') + P'_{[1,1,1]}(n - m, m - 1)$$

$$P'_{[1,1,1]}(n, m) = \sum_{\substack{1 \le m' \le m \\ m' \ne m-1}} P_{[1,1,1]}(n - m, m') + P''_{[1,1,1]}(n - m, m - 1)$$

$$P''_{[1,1,1]}(n, m) = \sum_{\substack{1 \le m' \le m \\ m' \ne m-1}} P_{[1,1,1]}(n - m, m')$$

This can be made into a quadratic time and memory algorithm to compute a table for $p_A(n)$.

Going beyond...

What if we want to count partitions with more specific restrictions, for example, not just globally, but also based on congruence conditions?

Going beyond...

What if we want to count partitions with more specific restrictions, for example, not just globally, but also based on congruence conditions?

* One side of Schur's celebrated 1926 theorem: partitions of m into parts with minimal difference 3 and with no consecutive multiples of 3.

Going beyond...

What if we want to count partitions with more specific restrictions, for example, not just globally, but also based on congruence conditions?

* One side of Schur's celebrated 1926 theorem: partitions of m into parts with minimal difference 3 and with no consecutive multiples of 3.

* And how about the a more complicated Kanade-Russell conjecture:
  (1) No parts repeat.
  (2) Adjacent parts do not differ by 1 if the smaller part is even.
  (3) A sub-partition of type $(2j+4)+(2j+2)+2j$ is not allowed.
  (4) A sub-partition of type $(2j+4)+(2j+2)+(2j+1)$ is not allowed.
  (5) A sub-partition of type $(2j+4)+(2j+3)+(2j+1)$ is not allowed.
  (6) Smallest part is at least 3.

Definition refined

- m, n, A: same as before (A is the set of patterns to avoid globally.)

Definition refined

* m, n, A: same as before (A is the set of patterns to avoid globally.)

* *Mod*: (loosely speaking) the list of patterns to avoid according to mod conditions of the *largest part* of a sub-partition.

  Examples:

  1. $\mathit{Mod} = [\{[1,1]\}, \{[2]\}, \{\}]$

Definition refined

- m, n, A: same as before (A is the set of patterns to avoid globally.)
- *Mod*: (loosely speaking) the list of patterns to avoid according to mod conditions of the *largest part* of a sub-partition.

  Examples:
  1. $Mod = [\{[1,1]\}, \{[2]\}, \{\}] \rightarrow$ forbidding sub-partition of type $(3j+3)+(3j+2)+(3j+1)$ and type $(3j+4)+(3j+2)$.

Definition refined

* m, n, A: same as before (A is the set of patterns to avoid globally.)
* *Mod*: (loosely speaking) the list of patterns to avoid according to mod conditions of the *largest part* of a sub-partition.

  Examples:
  1. $Mod = [\{[1,1]\}, \{[2]\}, \{\}] \rightarrow$ forbidding sub-partition of type $(3j+3)+(3j+2)+(3j+1)$ and type $(3j+4)+(3j+2)$.

  2. $Mod = [\{[0]\}, \{[0,0]\}]$

Definition refined

- m, n, A: same as before (A is the set of patterns to avoid globally.)

- *Mod*: (loosely speaking) the list of patterns to avoid according to mod conditions of the *largest part* of a sub-partition.

  Examples:
  1. $Mod = [\{[1,1]\}, \{[2]\}, \{\}] \to$ forbidding sub-partition of type $(3j+3)+(3j+2)+(3j+1)$ and type $(3j+4)+(3j+2)$.

  2. $Mod = [\{[0]\}, \{[0,0]\}] \to$ even parts are not allowed to repeat, and odd parts can appear at most twice.

Definition refined

* m, n, A: same as before (A is the set of patterns to avoid globally.)

* *Mod*: (loosely speaking) the list of patterns to avoid according to mod conditions of the *largest part* of a sub-partition.

  Examples:
  1. $Mod = [\{[1,1]\}, \{[2]\}, \{\}] \to$ forbidding sub-partition of type $(3j+3)+(3j+2)+(3j+1)$ and type $(3j+4)+(3j+2)$.

  2. $Mod = [\{[0]\}, \{[0,0]\}] \to$ even parts are not allowed to repeat, and odd parts can appear at most twice.

* B: the set of patterns to avoid at the beginning of the partition

* I: the set of sub-partitions to avoid (we call this "initial conditions")

Schur

(1) Parts with minimal difference 3. $\rightarrow A = \{[0],[1],[2]\}$

(2) No sub-partition of type $(3j+3)+3j$. $\rightarrow Mod = [\{[3]\},\{\},\{\}]$

Kanade-Russell

(1) No parts repeat. $\rightarrow A = \{[0]\}$

(2) A sub-partition of type $(2j+1)+2j$ is not allowed.
(3) A sub-partition of type $(2j+4)+(2j+2)+2j$ is not allowed.
(4) A sub-partition of type $(2j+4)+(2j+2)+(2j+1)$ is not allowed.
(5) A sub-partition of type $(2j+4)+(2j+3)+(2j+1)$ is not allowed.

(2),(3),(4),(5) $\rightarrow Mod = [\{[2,2],[2,1],[1,2]\},\{[1]\}]$

(6) Smallest part is at least 3. $\rightarrow I = \{[1],[2]\}$.

Generalized algorithm

Let $GP(m, n, A, Mod, B, I)$ be the number of partitions of n, with largest part m, and the restrictions A, Mod, B, I.

(1) If $m > n$, return 0. If $m = n$, return 1.

(2) Check if m is equal to the largest part of any of the forbidden sub-partitions in I: if so, and if the forbidden sub-partition is just [m] then return 0, otherwise we add the underlying partition pattern to B and we have a set of new beginning restrictions B'.

Generalized algorithm

(3) If $Mod = \{\}$, then by chopping off the largest part we get the recurrence:

$$GP(n, m, A, Mod, B, I) = \sum_{\substack{1 \le m' \le m \\ [m-m'] \notin A \cup B'}} GP(n-m, m', A, Mod, B'', I).$$

Note the "valid" m' will be those such that the singleton $[m - m']$ is not in the forbidden patterns (either globally or at the beginning). B'' is the set of new beginning restrictions, obtained from $A \cup B'$ by chopping off the difference $m - m'$ from the patterns in $A \cup B'$.

Generalized algorithm

(3) If $Mod = \{\}$, then by chopping off the largest part we get the recurrence:

$$GP(n, m, A, Mod, B, I) = \sum_{\substack{1 \le m' \le m \\ [m-m'] \notin A \cup B'}} GP(n-m, m', A, Mod, B'', I).$$

Note the "valid" m' will be those such that the singleton $[m - m']$ is not in the forbidden patterns (either globally or at the beginning). B'' is the set of new beginning restrictions, obtained from $A \cup B'$ by chopping off the difference $m - m'$ from the patterns in $A \cup B'$.

$\cancel{m}$ m' $[a, b] \in A \cup B'$

a b

Generalized algorithm

(3) If $Mod = \{\}$, then by chopping off the largest part we get the recurrence:

$$GP(n, m, A, Mod, B, I) = \sum_{\substack{1 \le m' \le m \\ [m-m'] \notin A \cup B'}} GP(n-m, m', A, Mod, B'', I).$$

Note the "valid" m' will be those such that the singleton $[m-m']$ is not in the forbidden patterns (either globally or at the beginning). B'' is the set of new beginning restrictions, obtained from $A \cup B'$ by chopping off the difference $m - m'$ from the patterns in $A \cup B'$.

$\cancel{m}$ $\quad m'$ $\qquad [a, b] \in A \cup B'$

$a \qquad b$

$\downarrow$

B''

Generalized algorithm

(4) If $Mod \neq \{\}$, let the length of Mod be k. If $m \equiv i \pmod{k}$ then we get the recurrence:

$$GP(n, m, A, Mod, B, I) = \sum_{\substack{1 \leq m' \leq m \\ [m-m'] \notin A \cup B' \cup Mod[i+1]}} GP(n-m, m', A, Mod, B'', I).$$

"Valid" m' will be those such that the singleton $[m - m']$ is not in the forbidden patterns (either globally or at the beginning or according to the Mod condition).

B'' is the set of new beginning restrictions, obtained from $A \cup B' \cup Mod[i+1]$ by chopping off the difference $m - m'$ from the patterns in $A \cup B' \cup Mod[i+1]$.

Generalized algorithm

(4) If $Mod \neq \{\}$, let the length of Mod be k. If $m \equiv i \pmod{k}$ then we get the recurrence:

$$GP(n, m, A, Mod, B, I) = \sum_{\substack{1 \leq m' \leq m \\ [m-m'] \notin A \cup B' \cup Mod[i+1]}} GP(n-m, m', A, Mod, B'', I).$$

"Valid" m' will be those such that the singleton $[m - m']$ is not in the forbidden patterns (either globally or at the beginning or according to the Mod condition).

B'' is the set of new beginning restrictions, obtained from $A \cup B' \cup Mod[i+1]$ by chopping off the difference $m - m'$ from the patterns in $A \cup B' \cup Mod[i+1]$.

Close analysis is still to be done but I think this algorithm can also be made quadratic in time and memory to compute a table for $p_{A,Mod,B,I}(n)$.

Searching for partition identities

A little (very incomplete) history of searching for identities

- 1894: Rogers-Ramanujan identities first published (MacMahon verified by hand, calculating 89 terms)

- 1952: "Slater list" (Lucy J. Slater worked out a list of 130 Rogers-Ramanujan type identities, deduced from Bailey pairs)

- 1970: Andrews computer search

- 1988: Capparelli identities (conjectured from VOA, proved by Andrews and many others later)

- 2009: Mc Laughlin, Sills and Zimmer computer search

- 2014+: Kanade-Russell computer search

- 2014: Nandi's conjectures (obtained from Lie algebra, 3 conjectures still open)

Preliminaries

"Sum side": a generating function that counts the pattern-avoiding partitions that we are currently interested in (according to A, Mod and I). May or may not have an analytic (multi-)sum.

"Product side": the side with infinite products. We use Frank Garvan's **qseries** Maple package to "factor" the generating function from the "sum side" into infinite products.

We use a "list notation" to denote a "product side", for example $[-2,-1,0,1,0]$ denotes $\frac{(q^4;q^5)_\infty}{(q;q^5)_\infty^2(q^2;q^5)_\infty}$. So if the list has only -1 and 0 in it, that means the "product side" satisfies certain congruence conditions. For example, $[-1,0,0,-1,0]$ denotes $\frac{1}{(q;q^5)_\infty(q^4;q^5)_\infty}$, that is, the parts are 1 or 4 modulo 5, which is a famous Rogers-Ramanujan "product side".

$Comp(m,k)$: the set of patterns of length at most m (i.e., at most m parts) and largest part at most k.

Our current search strategy

GP(n,m,A,Mod,B,I)

$\downarrow$

$\textbf{GxnSeq}(\mathbf{N}, \mathbf{A}, \mathbf{Mod}, \mathbf{B}, \mathbf{I})$

the first N terms of the sequence enumerating partitions obeying restrictions according to A, Mod, B and I

$\downarrow$

$\textbf{Search}(\mathbf{N}, \overline{\mathbf{A}}, \mathbf{Mod}, \mathbf{B}, \mathbf{I}, \mathbf{S})$

searches partition identities that have "product side" up to mod $\lfloor N/2 \rfloor$

$\overline{\mathbf{A}}$: a *set of sets* of forbidden patterns. Example: $\overline{\mathbf{A}} = \{\{[0]\}, \{[1], [2, 2]\}\}$

We will search through all sets in $\overline{\mathbf{A}}$, *"powersets"* of **Mod**, as well as powerset of **I**. S is the set of elements allowed to appear in the "list notation" of the "product side". We take $S = \{-2, -1, 0, 1\}$. And we ususally just assume B to be $\{\}$ since we are not interested in restrictions at the beginning at this moment.

Search examples

Let's look at some examples to see how $\mathbf{Search}(\mathbf{N}, \overline{\mathbf{A}}, \mathbf{Mod}, \mathbf{B}, \mathbf{I}, \mathbf{S})$ works.

* $\overline{A} = \{\{[0]\}, \{[1], [2]\}\}$, $Mod = [\ ]$, $I = \{\}$
  -This means we search for partitions that avoid either $A = \{[0]\}$ globally, or avoid $A = \{[1], [2]\}$ globally.

* $\overline{A} = \{\}$, $Mod = [\{[0]\}, \{[0, 0]\}]$, $I = \{[1]\}$
  -This means we search for partitions that avoid nothing globally, but has either $Mod = [\{\}, \{[0, 0]\}]$, $Mod = [\{[0]\}, \{\}]$, $Mod = [\{[0]\}, \{[0, 0]\}]$ or $Mod = [\{\}, \{\}]$ (which is no restriction at all) for Mod restrictions. In addition, we are either avoiding 1 as a part, or nothing at all for "initial conditions".

Amarel cluster computing

Thanks to Amarel, we are able to split our job into smaller tasks and feed the tasks to 500 nodes and "theoretically" increase our speed by 500 times.

Search space

$\overline{\mathsf{A}}$	Mod	I
powerset of $Comp(2,2)$	$[Comp(1,3), Comp(1,3), Comp(1,3)]$	$\{[1],[2]\}$
($\leq$ 2 parts, parts $\leq$ 2)	($\leq$ 1 part, parts $\leq$ 3)	
$\{Comp(5,1), 5\}$	$[\ \ ]$	$\{[1]\}$
($\leq$ 5 parts, parts $\leq$ 1, forbidding at most 5 global patterns)		
$\{Comp(4,2), 5\}$	$[\ \ ]$	$\{[1]\}$
$\{Comp(4,1), 3\}$	$[Comp(1,3), Comp(1,3), Comp(1,3)]$	$\{[1]\}$
$\{\}$	$[\{Comp(2,3), 3\}, \{Comp(2,3), 3\}]$	$\{[1]\}$

Basically, we have to ensure that we are checking (approximately) at most 2^{27} "sum-sides" for Amarel to be able to handle in one day.
Disclaimer: even for the list above, we did not exhaust everything in them because a few of the tasks ran into problems on Amarel.

Discoveries

Needless to say, we discovered many old identities, like Gordon, Andrews-Bressoud, Capparelli, among many others. But many of them are new. I will present a very incomplete list here.

A, Mod, I $\rightarrow$ Product Side

(1) $\{\}$, $[\{[1], [2]\}, \{[0], [2], [3]\}, \{\}]$, $\{\}$ $\rightarrow$ 0, 1, 3, 6, 7, 8, 9, 11 *mod* 12

(2) $\{\}$, $[\{\}, \{[1], [2]\}, \{[0], [2], [3]\}]$, $\{[1], [2]\}$ $\rightarrow$ 0, 3, 4, 5, 6, 9, 11 *mod* 12

(3) $\{[1]\}$, $[\{[0], [3]\}, \{\}, \{\}]$, $\{\}$ $\rightarrow$ 1, 2, 4, 6, 8, 10, 11 *mod* 12

(4) $\{[1]\}$, $[\{\},\{[0], [3]\}, \{\}]$, $\{[1]\}$ $\rightarrow$ 0, 2, 3, 4, 6, 9, 10 *mod* 12

(5) $\{[1]\}$, $[\{\}, \{\},\{[0], [3]\}]$, $\{[1]\}$ $\rightarrow$ 0, 2, 3, 6, 8, 9, 10 *mod* 12

(6) $\{[1]\}$, $[\{\}, \{[0], [3]\},\{[0], [3]\}]$, $\{[1]\}$ $\rightarrow$ 0, 2, 3, 6, 9, 10 *mod* 12

A, Mod, I $\rightarrow$ Product Side

(7) $\{\}$, $[\{[0, 1]\}, \{[2], [1, 1]\}]$, $\{\} \rightarrow 0, 1, 2, 3, 6, 7, 8, 9, 10$ *mod* 12

(8) $\{\}$, $[\{[0, 1], [1, 2]\}, \{[0], [1, 1], [2, 2]\}]$, $\{\} \rightarrow 0, 1, 3, 4, 7, 8, 9, 10$ *mod* 12

(9) $\{[1, 0], [1, 1, 1]\}$, $[\ \ ]$, $\{[1]\} \rightarrow 0, 2, 3, 4, 5, 6, 8, 9, 11$ *mod* 12

(10) $\{[1, 1], [0, 0, 0], [1, 0, 1], [1, 0, 0, 1]\}$, $[\ \ ]$, $\{\} \rightarrow 1, 2, 3, 5, 7, 9, 10, 11$ *mod* 12

A, Mod, I $\to$ Product Side

(7) $\{\}$, $[\{[0, 1]\}, \{[2], [1, 1]\}]$, $\{\} \to 0, 1, 2, 3, 6, 7, 8, 9, 10$ *mod* 12

(8) $\{\}$, $[\{[0, 1], [1, 2]\}, \{[0], [1, 1], [2, 2]\}]$, $\{\} \to 0, 1, 3, 4, 7, 8, 9, 10$ *mod* 12

(9) $\{[1, 0], [1, 1, 1]\}$, $[\ \ ]$, $\{[1]\} \to 0, 2, 3, 4, 5, 6, 8, 9, 11$ *mod* 12

(10) $\{[1, 1], [0, 0, 0], [1, 0, 1], [1, 0, 0, 1]\}$, $[\ \ ]$, $\{\} \to 1, 2, 3, 5, 7, 9, 10, 11$ *mod* 12

From (9), we obtained its "companion identity" by hand:
(9) $\{[0, 1], [1, 1, 1]\}$, $[\ \ ]$, $\{[1,1], [3,2,1]\} \to 0, 1, 3, 4, 6, 7, 8, 9, 10$ *mod* 12

An infinite family

Revisit (10): $\{[1, 1], [0, 0, 0], [1, 0, 1], [1, 0, 0, 1]\}$, [ ], $\{\} \to$ 1, 2, 3, 5, 7, 9, 10, 11 *mod* 12

$*$ Observe that the "sum side" of (10) is equivalent to:
–At most 3 occurrences of every part
–For all i, not allowed to have i, $i+1$, $i+2$ in the partition

An infinite family

Revisit (10): $\{[1, 1], [0, 0, 0], [1, 0, 1], [1, 0, 0, 1]\}$, [ ], $\{\} \rightarrow$ 1, 2, 3, 5, 7, 9, 10, 11 *mod* 12

- Observe that the "sum side" of (10) is equivalent to:
  –At most 3 occurrences of every part
  –For all i, not allowed to have i, $i+1$, $i+2$ in the partition

- This seems to generalize to an infinite family:
  – At most k occurrences of any given part
  – For all i, not allowed to have i, $i+1, \ldots, i+k-1$ all as parts in the partition

Additional identities

The "product sides" of (1) – (10) all correspond to partitions whose parts satisfy certain congruence conditions, or equivalently, only 0 and -1 are present in the "list notation". Here are some identities we found that also allow 1 (again, a very incomplete list):

(11) $\{[0,1,0]\}$, $[\{[0]\}, \{\}, \{\}]$, $\{\} \rightarrow$ [-1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, 0] (mod 12)
(12) $\{\}$, $[\{[1], [2]\}, \{[2]\}, \{[0], [3]\}]$, $\{\} \rightarrow$ [-1, -1, -1, 1, 0, -1, -1, -1, -1, 0, 0, -1] (mod 12)
(13) $\{\}$, $[\{[1], [2]\}, \{[0], [2], [3]\}, \{[0], [3]\}]$, $\{\} \rightarrow$ [-1, 0, -1, 1, 0, -1, -1, -1, -1, 0, 0, -1] (mod 12)
(14) $\{[1, 2], [2, 1]\}$, $[\{\}, \{[0], [1], [2], [3]\}, \{[2]\}]$, $\{[1], [2]\} \rightarrow$ [-1, 0, -1, 1, 0, -1, -1, -1, -1, 0, 0, -1] (mod 12)
(15) $\{[0]\}$, $[\{[2], [1, 1]\}, \{[1, 2], [3, 2]\}]$, $\{\} \rightarrow$ [-1, 0, -1, 0, -1, 1, -1, -1, -1, 1, -1, -1, -1, 1, -1, 0, -1, 0, -1, 0] (mod 20)

Future work

1. Deal with the cases that ran into problems on Amarel. This will enable us to say something like: we have searched everything in this group, and we are sure no identities (less than certain modulo) can be found in this group.

2. Search for larger modulo identities by increasing the **N** in $\mathbf{Search}(\mathbf{N}, \overline{\mathbf{A}}, \mathbf{Mod}, \mathbf{B}, \mathbf{I}, \mathbf{S})$.

3. Put more variations on the initial conditions.

4. Currently our approach only deals with conditions on contiguous sub-partitions. It will be nice to develop a general frame work/an efficient way to search for identities that avoid sub-partitions that are not necessarily contiguous (like in the infinite family we presented).

Drew Sill and Ali Uncu's suggestions

1. Some identities have "wierd" "sum side", for example, the big Göllnitz companion identity "sum side" requires difference of at least 6 between parts EXCEPT that it is ok if the smallest two parts are 1 and 6. Maybe many such "wierd" partition identites are out there, we would like to search for them.

2. Incorporate Nandi's $*$ operator, which is the asterisk in the pattern $[3, 2*, 3, 0]$ into our program to search for more Nandi-type partition identities.

Thank you!