The Mullineux map

Matt Fayers

Queen Mary
University of London

Representations of symmetric groups

Representations of symmetric groups

$\mathfrak{S}_n$ the symmetric group, p a prime, $\lambda = (\lambda_1, \lambda_2, \dots)$ a partition of n.

Representations of symmetric groups

$\mathfrak{S}_n$ the symmetric group, p a prime, $\lambda = (\lambda_1, \lambda_2, \dots)$ a partition of n.

The Specht module S^λ is an $\mathfrak{S}_n$-module defined over any field $\mathbb{F}$.

Representations of symmetric groups

$\mathfrak{S}_n$ the symmetric group, p a prime, $\lambda = (\lambda_1, \lambda_2, \dots)$ a partition of n.

The Specht module S^λ is an $\mathfrak{S}_n$-module defined over any field $\mathbb{F}$.

Say λ is p-regular if it does have not p equal positive parts.

Representations of symmetric groups

$\mathfrak{S}_n$ the symmetric group, p a prime, $\lambda = (\lambda_1, \lambda_2, \dots)$ a partition of n.

The Specht module S^λ is an $\mathfrak{S}_n$-module defined over any field $\mathbb{F}$.

Say λ is p-regular if it does have not p equal positive parts.

If $\mathrm{char}(\mathbb{F}) = p$ and λ is p-regular, then S^λ has a unique simple quotient D^λ (the James module). The set

$$\left\{ D^\lambda \;\middle|\; \lambda \text{ a } p\text{-regular partition of } n \right\}$$

is a complete set of simple modules for $\mathfrak{S}_n$ in characteristic p.

The Mullineux problem

The Mullineux problem

One of the simple modules is the one-dimensional sign module, on which each $\pi \in \mathfrak{S}_n$ acts as $\mathrm{sgn}(\pi)$.

The Mullineux problem

One of the simple modules is the one-dimensional sign module, on which each $\pi \in \mathfrak{S}_n$ acts as $\mathrm{sgn}(\pi)$.

If S is a simple module, then so is $S \otimes \mathrm{sgn}$. So there is a bijection

$$t_p : \{p\text{-regular partitions of } n\} \longrightarrow \{p\text{-regular partitions of } n\}$$

such that

$$\mathrm{D}^{\lambda} \otimes \mathrm{sgn} \cong \mathrm{D}^{t_p(\lambda)}.$$

The Mullineux problem

One of the simple modules is the one-dimensional sign module, on which each $\pi \in \mathfrak{S}_n$ acts as $\mathrm{sgn}(\pi)$.

If S is a simple module, then so is $S \otimes \mathrm{sgn}$. So there is a bijection

$$t_p : \{p\text{-regular partitions of } n\} \longrightarrow \{p\text{-regular partitions of } n\}$$

such that

$$\mathrm{D}^{\lambda} \otimes \mathrm{sgn} \cong \mathrm{D}^{t_p(\lambda)}.$$

Mullineux problem: Describe the map t_p combinatorially.

The Mullineux problem – examples

The Mullineux problem – examples

- Let λ' denote the conjugate (or transpose) partition.

The Mullineux problem – examples

- Let λ' denote the conjugate (or transpose) partition.

  If $p \gg 0$, then $\mathrm{D}^\lambda = \mathrm{S}^\lambda$, and

  $$\mathrm{S}^\lambda \otimes \mathrm{sgn} \cong \mathrm{S}^{\lambda'}.$$

  So $t_p : \lambda \mapsto \lambda'$ if p is large.

The Mullineux problem – examples

- Let λ' denote the conjugate (or transpose) partition.

  If $p \gg 0$, then $\mathrm{D}^\lambda = \mathrm{S}^\lambda$, and

  $$\mathrm{S}^\lambda \otimes \mathrm{sgn} \cong \mathrm{S}^{\lambda'}.$$

  So $t_p : \lambda \mapsto \lambda'$ if p is large.

- If $p = 2$, then $S \otimes \mathrm{sgn} = S$ for any S, so t_2 is the identity.

The Mullineux problem – examples

- Let λ' denote the conjugate (or transpose) partition.

  If $p \gg 0$, then $D^\lambda = S^\lambda$, and

  $$S^\lambda \otimes \mathrm{sgn} \cong S^{\lambda'}.$$

  So $t_p : \lambda \mapsto \lambda'$ if p is large.

- If $p = 2$, then $S \otimes \mathrm{sgn} = S$ for any S, so t_2 is the identity.

- $p = 3$, $n = 3$

The Mullineux problem – examples

- Let λ' denote the conjugate (or transpose) partition.

  If $p \gg 0$, then $\mathrm{D}^\lambda = \mathrm{S}^\lambda$, and

  $$\mathrm{S}^\lambda \otimes \mathrm{sgn} \cong \mathrm{S}^{\lambda'}.$$

  So $t_p : \lambda \mapsto \lambda'$ if p is large.

- If $p = 2$, then $S \otimes \mathrm{sgn} = S$ for any S, so t_2 is the identity.

- $p = 3$, $n = 3$
  The 3-regular partitions of 3 are (3), $(2,1)$. So $\mathrm{D}^{(3)}$ and $\mathrm{D}^{(2,1)}$ are the trivial module and the sign module.

The Mullineux problem – examples

- Let λ' denote the conjugate (or transpose) partition.

  If $p \gg 0$, then $\mathrm{D}^\lambda = \mathrm{S}^\lambda$, and

  $$\mathrm{S}^\lambda \otimes \mathrm{sgn} \cong \mathrm{S}^{\lambda'}.$$

  So $t_p : \lambda \mapsto \lambda'$ if p is large.

- If $p = 2$, then $S \otimes \mathrm{sgn} = S$ for any S, so t_2 is the identity.

- $p = 3$, $n = 3$
  The 3-regular partitions of 3 are (3), $(2,1)$. So $\mathrm{D}^{(3)}$ and $\mathrm{D}^{(2,1)}$ are the trivial module and the sign module. So

  $$t_3 : (3) \longmapsto (2,1), \qquad (2,1) \longmapsto (3).$$

The Mullineux problem – history

The Mullineux problem – history

1979 Mullineux gave a combinatorial bijection m_p and conjectured that $t_p = m_p$.

The Mullineux problem – history

1979 Mullineux gave a combinatorial bijection m_p and conjectured that $t_p = m_p$.

1994 Kleshchev gave a different combinatorial bijection k_p and proved (using modular branching rules) that $t_p = k_p$.

The Mullineux problem – history

1979 Mullineux gave a combinatorial bijection m_p and conjectured that $t_p = m_p$.

1994 Kleshchev gave a different combinatorial bijection k_p and proved (using modular branching rules) that $t_p = k_p$.

1997 Ford and Kleshchev proved combinatorially that $k_p = m_p$, so proved the Mullineux conjecture.

The Mullineux problem – history

1979 Mullineux gave a combinatorial bijection m_p and conjectured that $t_p = m_p$.

1994 Kleshchev gave a different combinatorial bijection k_p and proved (using modular branching rules) that $t_p = k_p$.

1997 Ford and Kleshchev proved combinatorially that $k_p = m_p$, so proved the Mullineux conjecture.

1997 Xu gave another combinatorial bijection x_p, and showed that $x_p = m_p$.

The Mullineux problem – history

1979 Mullineux gave a combinatorial bijection m_p and conjectured that $t_p = m_p$.

1994 Kleshchev gave a different combinatorial bijection k_p and proved (using modular branching rules) that $t_p = k_p$.

1997 Ford and Kleshchev proved combinatorially that $k_p = m_p$, so proved the Mullineux conjecture.

1997 Xu gave another combinatorial bijection x_p, and showed that $x_p = m_p$.

1998 Bessenrodt and Olsson gave a shorter proof that $k_p = m_p$.

The Mullineux problem – history

1979 Mullineux gave a combinatorial bijection m_p and conjectured that $t_p = m_p$.

1994 Kleshchev gave a different combinatorial bijection k_p and proved (using modular branching rules) that $t_p = k_p$.

1997 Ford and Kleshchev proved combinatorially that $k_p = m_p$, so proved the Mullineux conjecture.

1997 Xu gave another combinatorial bijection x_p, and showed that $x_p = m_p$.

1998 Bessenrodt and Olsson gave a shorter proof that $k_p = m_p$.

1999 Wang and Xu gave a short proof that $x_p = k_p$.

The Mullineux problem – history

1979 Mullineux gave a combinatorial bijection m_p and conjectured that $t_p = m_p$.

1994 Kleshchev gave a different combinatorial bijection k_p and proved (using modular branching rules) that $t_p = k_p$.

1997 Ford and Kleshchev proved combinatorially that $k_p = m_p$, so proved the Mullineux conjecture.

1997 Xu gave another combinatorial bijection x_p, and showed that $x_p = m_p$.

1998 Bessenrodt and Olsson gave a shorter proof that $k_p = m_p$.

1999 Wang and Xu gave a short proof that $x_p = k_p$.

2003 Brundan and Kujawa gave another combinatorial map s_p, and proved (using Serganova's work on the supergroup $\mathrm{GL}(m|n)$) that $t_p = s_p = x_p$.

The Mullineux problem – history

1979 Mullineux gave a combinatorial bijection m_p and conjectured that $t_p = m_p$.

1994 Kleshchev gave a different combinatorial bijection k_p and proved (using modular branching rules) that $t_p = k_p$.

1997 Ford and Kleshchev proved combinatorially that $k_p = m_p$, so proved the Mullineux conjecture.

1997 Xu gave another combinatorial bijection x_p, and showed that $x_p = m_p$.

1998 Bessenrodt and Olsson gave a shorter proof that $k_p = m_p$.

1999 Wang and Xu gave a short proof that $x_p = k_p$.

2003 Brundan and Kujawa gave another combinatorial map s_p, and proved (using Serganova's work on the supergroup $\mathrm{GL}(m|n)$) that $t_p = s_p = x_p$.

So $m_p = k_p = x_p = s_p = t_p$.

The Mullineux problem – history

1979 Mullineux gave a combinatorial bijection m_p and conjectured that $t_p = m_p$.

1994 Kleshchev gave a different combinatorial bijection k_p and proved (using modular branching rules) that $t_p = k_p$.

1997 Ford and Kleshchev proved combinatorially that $k_p = m_p$, so proved the Mullineux conjecture.

1997 Xu gave another combinatorial bijection x_p, and showed that $x_p = m_p$.

1998 Bessenrodt and Olsson gave a shorter proof that $k_p = m_p$.

1999 Wang and Xu gave a short proof that $x_p = k_p$.

2003 Brundan and Kujawa gave another combinatorial map s_p, and proved (using Serganova's work on the supergroup $\mathrm{GL}(m|n)$) that $t_p = s_p = x_p$.

So $m_p = k_p = x_p = s_p = t_p$. Write these all as m_p.

Mullineux’s bijection

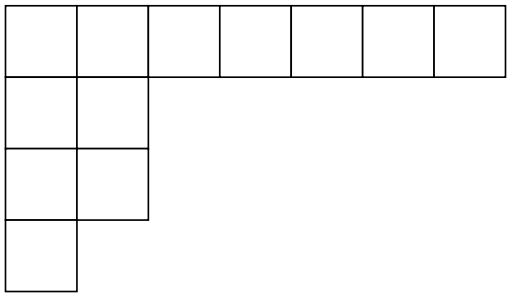
Mullineux's bijection

The rim of λ is the south-east edge of the Young diagram.

Mullineux's bijection

The rim of λ is the south-east edge of the Young diagram.

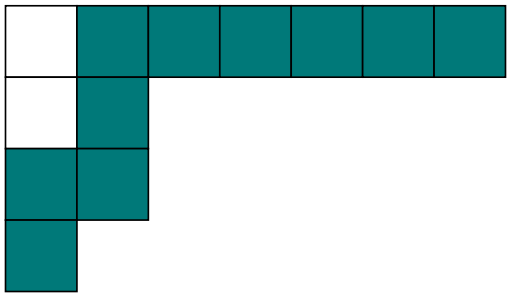
e.g. $\lambda = (7, 2^2, 1)$



Mullineux's bijection

The rim of λ is the south-east edge of the Young diagram.

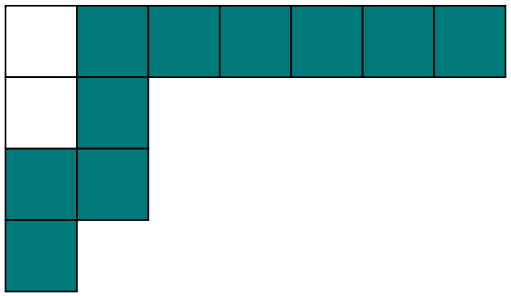
e.g. $\lambda = (7, 2^2, 1)$



Mullineux's bijection

The rim of λ is the south-east edge of the Young diagram.

e.g. $\lambda = (7, 2^2, 1)$

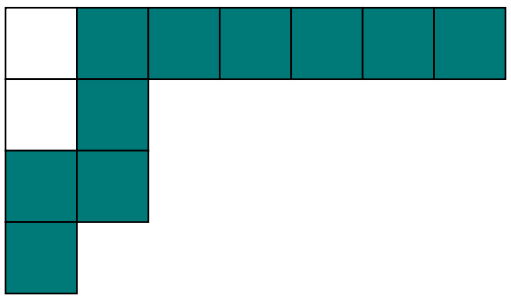


The p-rim of λ is obtained by working along the rim from north-east to south-west, marking nodes.

Mullineux's bijection

The rim of λ is the south-east edge of the Young diagram.

e.g. $\lambda = (7, 2^2, 1)$

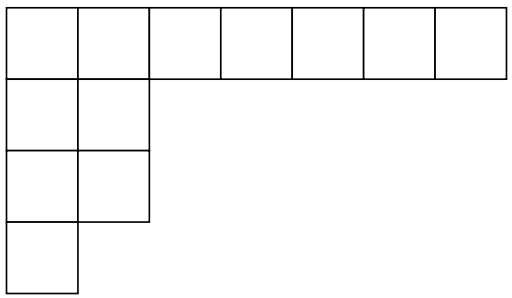


The p-rim of λ is obtained by working along the rim from north-east to south-west, marking nodes. Each time the number of marked nodes is divisible by p, move to the next row.

Mullineux's bijection

The rim of λ is the south-east edge of the Young diagram.

e.g. $\lambda = (7, 2^2, 1)$, $p = 3$

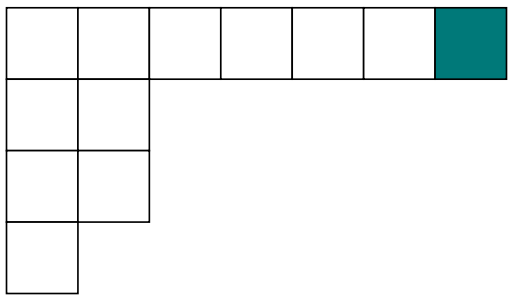


The p-rim of λ is obtained by working along the rim from north-east to south-west, marking nodes. Each time the number of marked nodes is divisible by p, move to the next row.

Mullineux's bijection

The rim of λ is the south-east edge of the Young diagram.

e.g. $\lambda = (7, 2^2, 1)$, $p = 3$

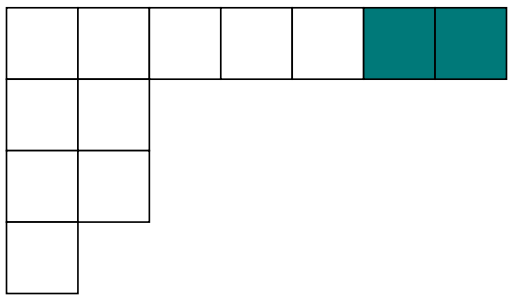


The p-rim of λ is obtained by working along the rim from north-east to south-west, marking nodes. Each time the number of marked nodes is divisible by p, move to the next row.

Mullineux's bijection

The rim of λ is the south-east edge of the Young diagram.

e.g. $\lambda = (7, 2^2, 1)$, $p = 3$

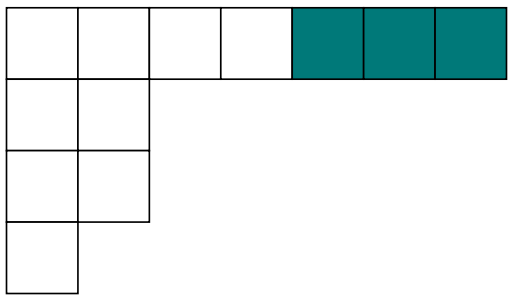


The p-rim of λ is obtained by working along the rim from north-east to south-west, marking nodes. Each time the number of marked nodes is divisible by p, move to the next row.

Mullineux's bijection

The rim of λ is the south-east edge of the Young diagram.

e.g. $\lambda = (7, 2^2, 1)$, $p = 3$

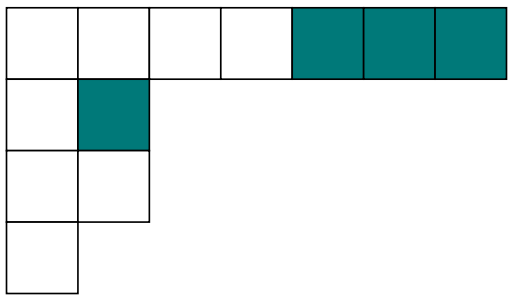


The p-rim of λ is obtained by working along the rim from north-east to south-west, marking nodes. Each time the number of marked nodes is divisible by p, move to the next row.

Mullineux's bijection

The rim of λ is the south-east edge of the Young diagram.

e.g. $\lambda = (7, 2^2, 1)$, $p = 3$

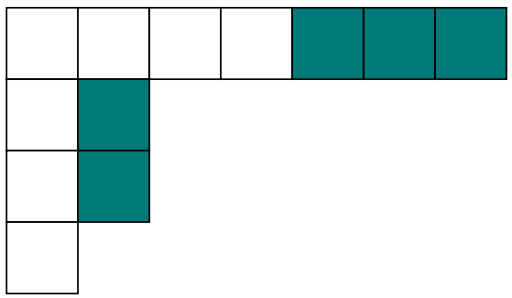


The p-rim of λ is obtained by working along the rim from north-east to south-west, marking nodes. Each time the number of marked nodes is divisible by p, move to the next row.

Mullineux's bijection

The rim of λ is the south-east edge of the Young diagram.

e.g. $\lambda = (7, 2^2, 1)$, $p = 3$

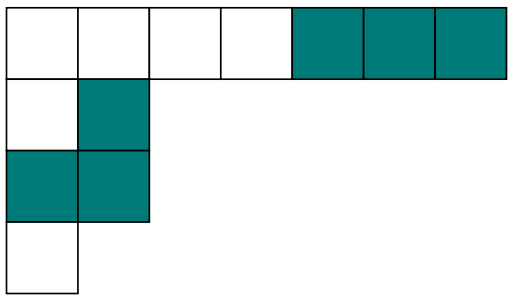


The p-rim of λ is obtained by working along the rim from north-east to south-west, marking nodes. Each time the number of marked nodes is divisible by p, move to the next row.

Mullineux's bijection

The rim of λ is the south-east edge of the Young diagram.

e.g. $\lambda = (7, 2^2, 1)$, $p = 3$

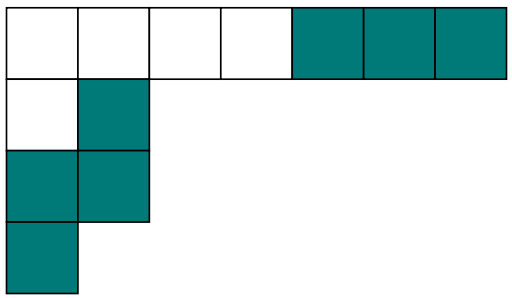


The p-rim of λ is obtained by working along the rim from north-east to south-west, marking nodes. Each time the number of marked nodes is divisible by p, move to the next row.

Mullineux's bijection

The rim of λ is the south-east edge of the Young diagram.

e.g. $\lambda = (7, 2^2, 1)$, $p = 3$



The p-rim of λ is obtained by working along the rim from north-east to south-west, marking nodes. Each time the number of marked nodes is divisible by p, move to the next row.

Mullineux's bijection

Mullineux's bijection

Mullineux's algorithm: Repeatedly remove the p-rim until you reach the empty partition.

Mullineux's bijection

Mullineux's algorithm: Repeatedly remove the p-rim until you reach the empty partition. At the ith stage, record

$$n_i = \text{size of the } p\text{-rim}$$
$$r_i = \text{number of rows of the current partition.}$$

Mullineux's bijection

Mullineux's algorithm: Repeatedly remove the p-rim until you reach the empty partition. At the ith stage, record

$$n_i = \text{size of the } p\text{-rim}$$
$$r_i = \text{number of rows of the current partition.}$$

Mullineux symbol of λ: $\begin{pmatrix} n_1 & n_2 & \dots \\ r_1 & r_2 & \dots \end{pmatrix}$.

Mullineux's bijection

Mullineux's algorithm: Repeatedly remove the p-rim until you reach the empty partition. At the ith stage, record

$$n_i = \text{size of the } p\text{-rim}$$
$$r_i = \text{number of rows of the current partition.}$$

Mullineux symbol of λ: $\begin{pmatrix} n_1 & n_2 & \dots \\ r_1 & r_2 & \dots \end{pmatrix}$.

Now define

$$s_i = \begin{cases} n_i - r_i & \text{if } p \mid n_i \\ n_i - r_i + 1 & \text{if } p \nmid n_i. \end{cases}$$

Mullineux's bijection

Mullineux's algorithm: Repeatedly remove the p-rim until you reach the empty partition. At the ith stage, record

$$n_i = \text{size of the } p\text{-rim}$$
$$r_i = \text{number of rows of the current partition.}$$

Mullineux symbol of λ: $\begin{pmatrix} n_1 & n_2 & \dots \\ r_1 & r_2 & \dots \end{pmatrix}$.

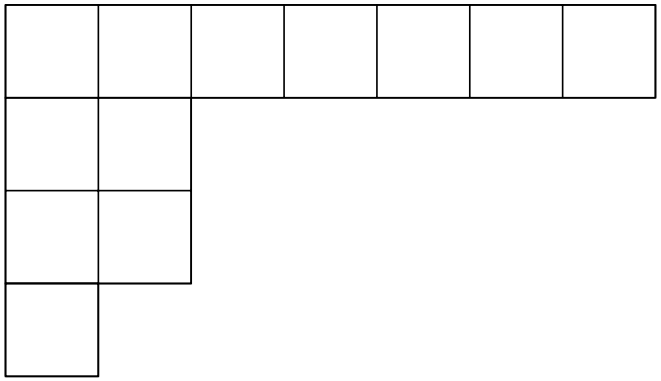
Now define

$$s_i = \begin{cases} n_i - r_i & \text{if } p \mid n_i \\ n_i - r_i + 1 & \text{if } p \nmid n_i. \end{cases}$$

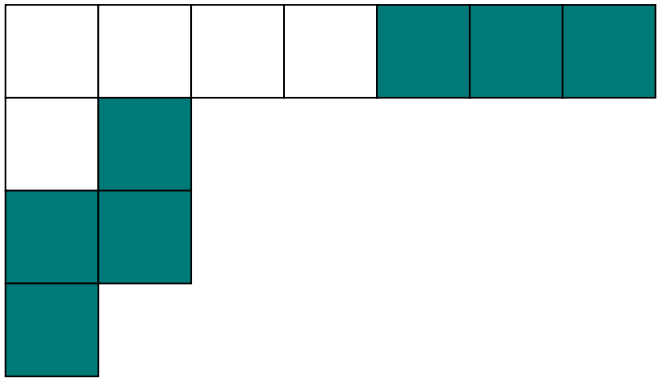
Mullineux bijection: $m_p(\lambda)$ is the p-regular partition whose Mullineux symbol is

$$\begin{pmatrix} n_1 & n_2 & \dots \\ s_1 & s_2 & \dots \end{pmatrix}.$$

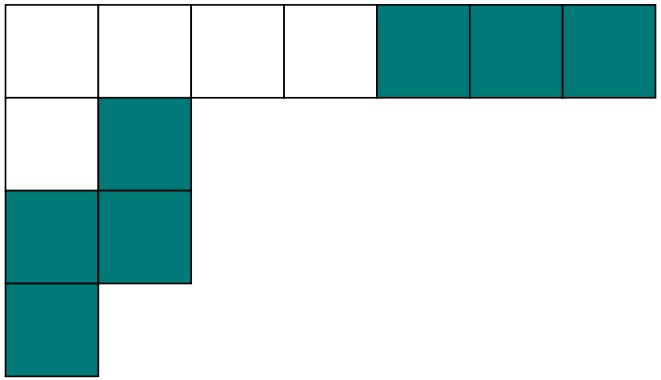
Mullineux's bijection – example



Mullineux's bijection – example



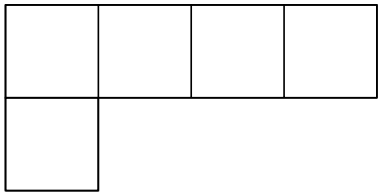
Mullineux's bijection – example



n_i 7

r_i 4

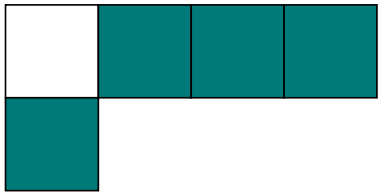
Mullineux's bijection – example



n_i 7

r_i 4

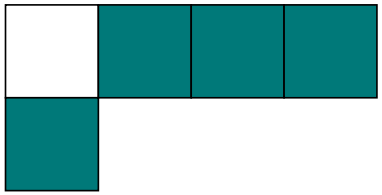
Mullineux's bijection – example



n_i 7

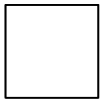
r_i 4

Mullineux's bijection – example



n_i	7	4
r_i	4	2

Mullineux's bijection – example



n_i	7	4
r_i	4	2

Mullineux's bijection – example

n_i	7	4
r_i	4	2

Mullineux's bijection – example

n_i	7	4	1
r_i	4	2	1

Mullineux's bijection – example

n_i	7	4	1
r_i	4	2	1

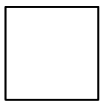
Mullineux's bijection – example

n_i	7	4	1
r_i	4	2	1
s_i	4	3	1

Mullineux's bijection – example

n_i	7	4	1
r_i	4	2	1
s_i	4	3	1

Mullineux's bijection – example



n_i	7	4	1
r_i	4	2	1
s_i	4	3	1

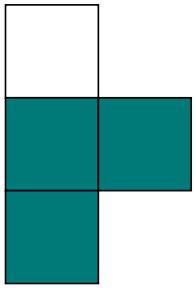
Mullineux's bijection – example

n_i	7	4	1
r_i	4	2	1
s_i	4	3	1

Mullineux's bijection – example

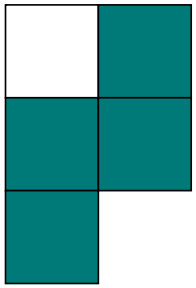
n_i	7	4	1
r_i	4	2	1
s_i	4	3	1

Mullineux's bijection – example



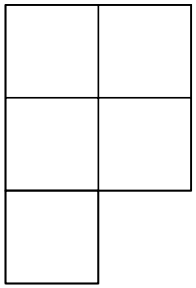
n_i	7	4	1
r_i	4	2	1
s_i	4	3	1

Mullineux's bijection – example



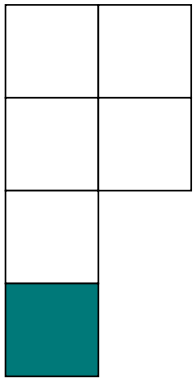
n_i	7	4	1
r_i	4	2	1
s_i	4	3	1

Mullineux's bijection – example



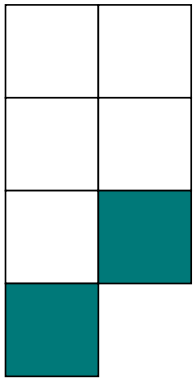
n_i	7	4	1
r_i	4	2	1
s_i	4	3	1

Mullineux's bijection – example



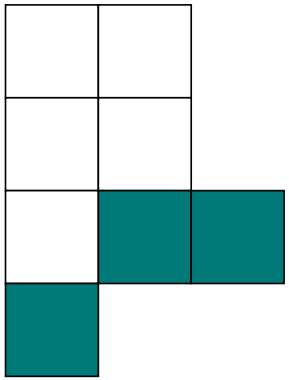
n_i	7	4	1
r_i	4	2	1
s_i	4	3	1

Mullineux's bijection – example



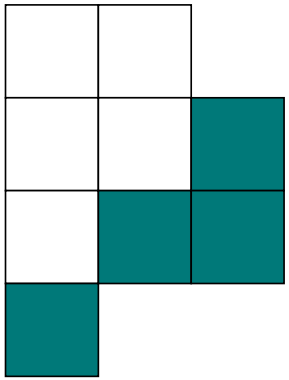
n_i	7	4	1
r_i	4	2	1
s_i	4	3	1

Mullineux's bijection – example



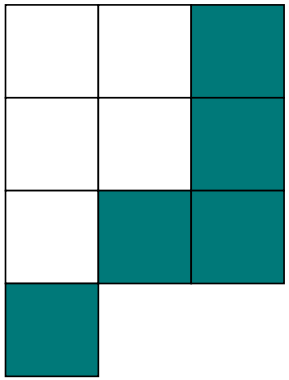
n_i	7	4	1
r_i	4	2	1
s_i	4	3	1

Mullineux's bijection – example



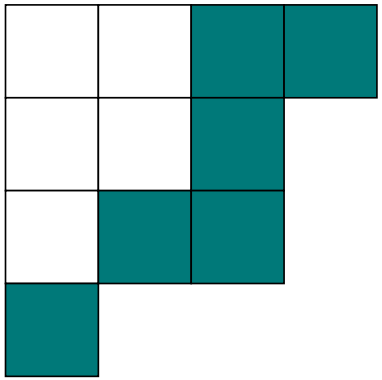
n_i	7	4	1
r_i	4	2	1
s_i	4	3	1

Mullineux's bijection – example



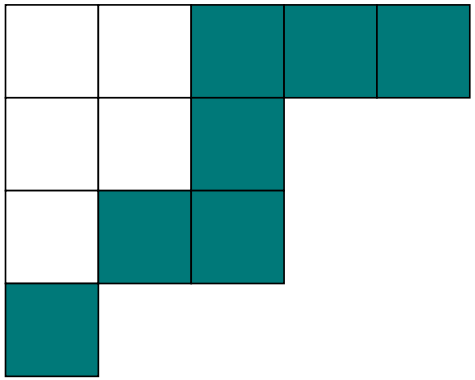
n_i	7	4	1
r_i	4	2	1
s_i	4	3	1

Mullineux's bijection – example



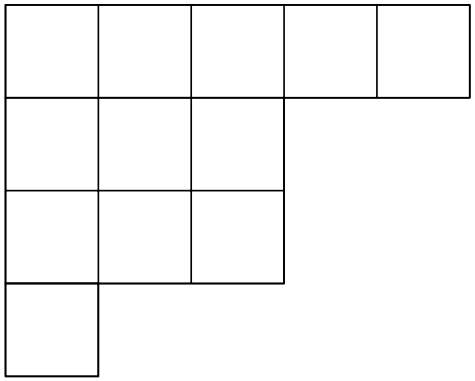
n_i	7	4	1
r_i	4	2	1
s_i	4	3	1

Mullineux's bijection – example



n_i	7	4	1
r_i	4	2	1
s_i	4	3	1

Mullineux's bijection – example



n_i	7	4	1
r_i	4	2	1
s_i	4	3	1

$$m_3(7, 2^2, 1) = (5, 3^2, 1).$$

Xu’s bijection

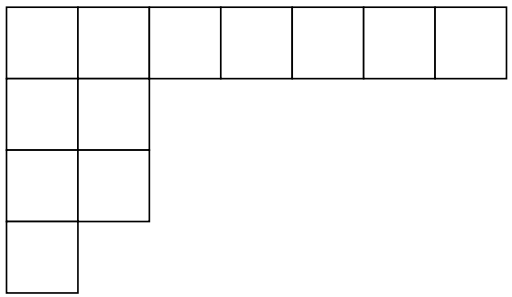
Xu's bijection

Truncated p-rim of λ: construct the p-rim, then remove the first node in each row (except the last row, if the size of the p-rim is not divisible by p).

Xu's bijection

Truncated p-rim of λ: construct the p-rim, then remove the first node in each row (except the last row, if the size of the p-rim is not divisible by p).

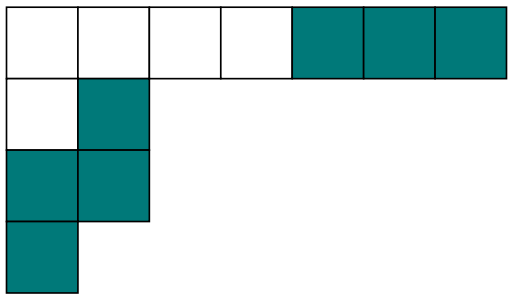
e.g. $\lambda = (7, 2^2, 1)$, $p = 3$



Xu's bijection

Truncated p-rim of λ: construct the p-rim, then remove the first node in each row (except the last row, if the size of the p-rim is not divisible by p).

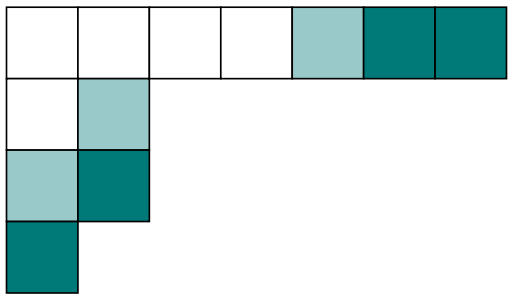
e.g. $\lambda = (7, 2^2, 1)$, $p = 3$



Xu's bijection

Truncated p-rim of λ: construct the p-rim, then remove the first node in each row (except the last row, if the size of the p-rim is not divisible by p).

e.g. $\lambda = (7, 2^2, 1)$, $p = 3$



Xu’s bijection

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition.

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

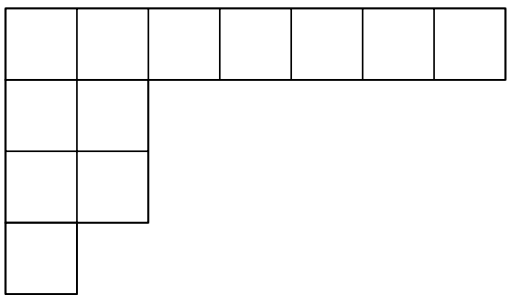
Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.

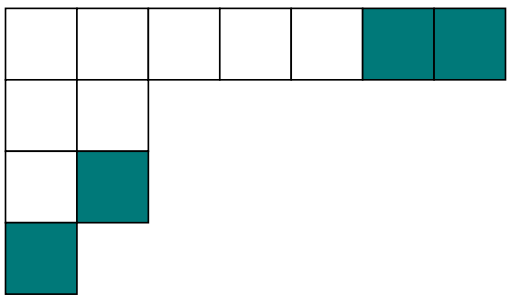


Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.

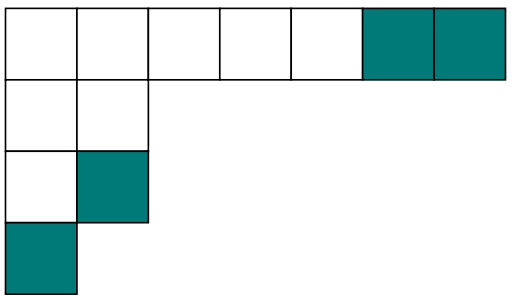


Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.



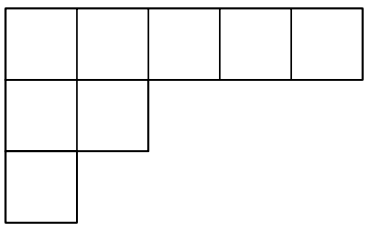
4

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.



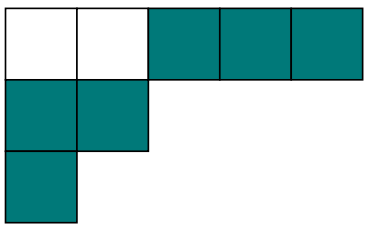
4

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.



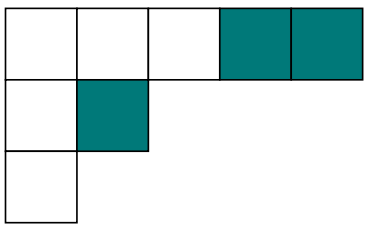
4

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.



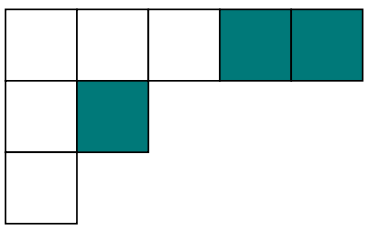
4

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.



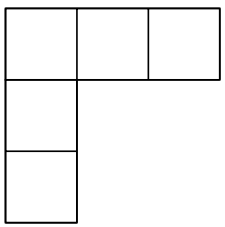
4 3

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.



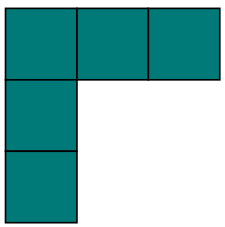
4 3

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.



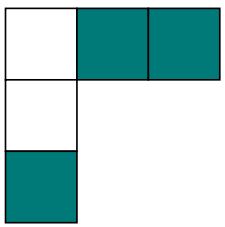
4 3

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.



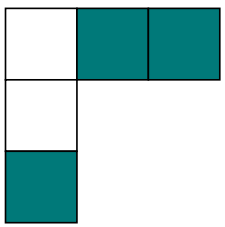
4 3

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.



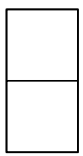
4 3 3

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.



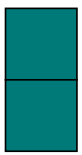
4 3 3

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.



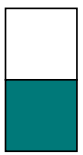
4 3 3

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.



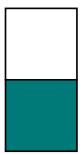
4 3 3

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.



4 3 3 1

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.

4 3 3 1

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.

4 3 3 1

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.

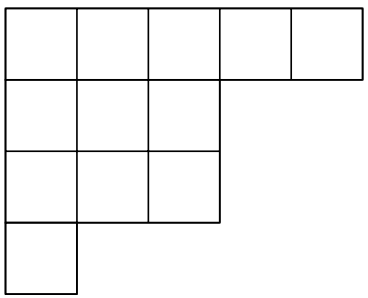
4 3 3 1 1

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.



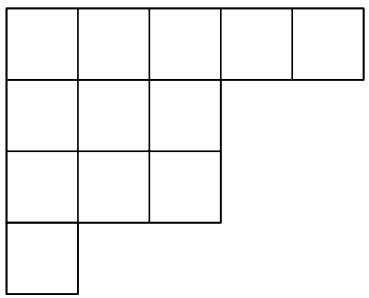
4 3 3 1 1

Xu's bijection

Xu's algorithm: Repeatedly remove the truncated p-rim until you reach the empty partition. At the ith step, record

$$m_i = \text{size of the truncated } p\text{-rim.}$$

Xu's bijection: $m_p(\lambda)$ is the partition whose columns have lengths $m_1, m_2, \ldots$.



$$4 \quad 3 \quad 3 \quad 1 \quad 1$$

$$m_3(7, 2^2, 1) = (5, 3^2, 1).$$

Kleshchev's bijection

Kleshchev's bijection

Robinson's restriction functors

$$e_i : \mathfrak{S}_n\text{-mod} \longrightarrow \mathfrak{S}_{n-1}\text{-mod} \qquad (i \in \mathbb{Z}/p\mathbb{Z})$$

such that for any module M,

$$M\downarrow_{\mathfrak{S}_{n-1}} = \bigoplus_{i\in\mathbb{Z}/p\mathbb{Z}} e_i M$$

Kleshchev's bijection

Robinson's restriction functors

$$e_i : \mathfrak{S}_n\text{-mod} \longrightarrow \mathfrak{S}_{n-1}\text{-mod} \qquad (i \in \mathbb{Z}/p\mathbb{Z})$$

such that for any module M,

$$M\downarrow_{\mathfrak{S}_{n-1}} = \bigoplus_{i \in \mathbb{Z}/p\mathbb{Z}} e_i M$$

$$e_i(M \otimes \mathrm{sgn}) \cong (e_{-i}M) \otimes \mathrm{sgn}.$$

Kleshchev's bijection

Robinson's restriction functors

$$e_i : \mathfrak{S}_n\text{-mod} \longrightarrow \mathfrak{S}_{n-1}\text{-mod} \qquad (i \in \mathbb{Z}/p\mathbb{Z})$$

such that for any module M,

$$M\downarrow_{\mathfrak{S}_{n-1}} = \bigoplus_{i \in \mathbb{Z}/p\mathbb{Z}} e_i M$$

$$e_i(M \otimes \mathrm{sgn}) \cong (e_{-i}M) \otimes \mathrm{sgn}.$$

Kleshchev's modular branching rules: $\mathrm{soc}(e_i \mathrm{D}^\lambda)$ is either 0 or simple

Kleshchev's bijection

Robinson's restriction functors

$$e_i : \mathfrak{S}_n\text{-mod} \longrightarrow \mathfrak{S}_{n-1}\text{-mod} \qquad (i \in \mathbb{Z}/p\mathbb{Z})$$

such that for any module M,

$$M\downarrow_{\mathfrak{S}_{n-1}} = \bigoplus_{i \in \mathbb{Z}/p\mathbb{Z}} e_i M$$

$$e_i(M \otimes \mathrm{sgn}) \cong (e_{-i}M) \otimes \mathrm{sgn}.$$

Kleshchev's modular branching rules: $\mathrm{soc}(e_i \mathrm{D}^\lambda)$ is either 0 or simple, and we can say combinatorially what happens.

Kleshchev's bijection

Kleshchev's bijection

Residue of node (r, c): residue of $c - r \pmod{p}$.

Kleshchev's bijection

Residue of node (r, c): residue of $c - r \pmod{p}$.

Given $i \in \mathbb{Z}/p\mathbb{Z}$, the i-signature of λ is obtained by working from top to bottom, writing

Kleshchev's bijection

Residue of node (r, c): residue of $c - r \pmod{p}$.

Given $i \in \mathbb{Z}/p\mathbb{Z}$, the i-signature of λ is obtained by working from top to bottom, writing

$+$ for each addable node of residue i

$-$ for each removable node of residue i.

Kleshchev's bijection

Residue of node (r, c): residue of $c - r \pmod{p}$.

Given $i \in \mathbb{Z}/p\mathbb{Z}$, the i-signature of λ is obtained by working from top to bottom, writing

$+$ for each addable node of residue i

$-$ for each removable node of residue i.

Reduced i-signature: successively delete pairs $+-$.

Kleshchev's bijection

Residue of node (r, c): residue of $c - r \pmod{p}$.

Given $i \in \mathbb{Z}/p\mathbb{Z}$, the i-signature of λ is obtained by working from top to bottom, writing

$+$ for each addable node of residue i

$-$ for each removable node of residue i.

Reduced i-signature: successively delete pairs $+-$.

Good node of residue i: removable node $\mathfrak{n}$ corresponding to the last $-$ in the reduced i-signature. Let $\tilde{e}_i\lambda = \lambda \setminus \mathfrak{n}$.

Kleshchev's bijection

Residue of node (r, c): residue of $c - r \pmod{p}$.

Given $i \in \mathbb{Z}/p\mathbb{Z}$, the i-signature of λ is obtained by working from top to bottom, writing

$+$ for each addable node of residue i

$-$ for each removable node of residue i.

Reduced i-signature: successively delete pairs $+-$.

Good node of residue i: removable node $\mathfrak{n}$ corresponding to the last $-$ in the reduced i-signature. Let $\tilde{e}_i\lambda = \lambda \setminus \mathfrak{n}$.

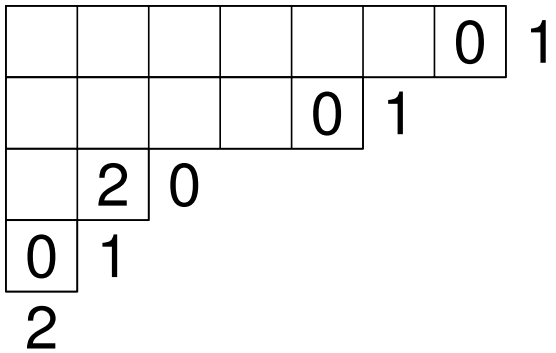
Kleshchev's modular branching rule:

$$\operatorname{soc}(e_i D^\lambda) = \begin{cases} D^{\tilde{e}_i\lambda} & \text{if } \lambda \text{ has a good } i\text{-node} \\ 0 & \text{otherwise.} \end{cases}$$

Kleshchev's bijection

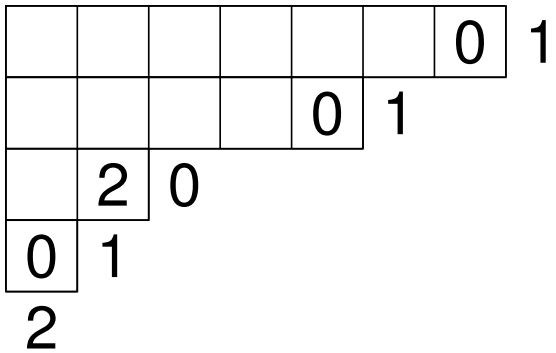
Kleshchev's bijection

e.g. $\lambda = (7, 5, 2, 1)$, $p = 3$



Kleshchev's bijection

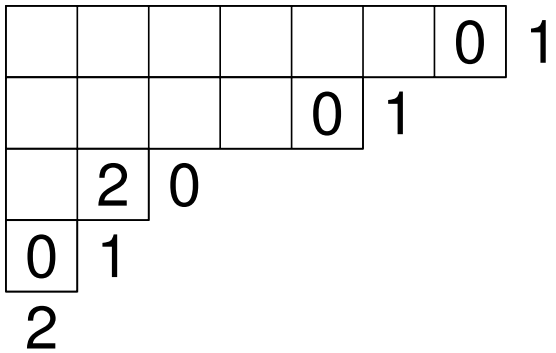
e.g. $\lambda = (7, 5, 2, 1)$, $p = 3$



0-signature $- - + -$

Kleshchev's bijection

e.g. $\lambda = (7, 5, 2, 1)$, $p = 3$

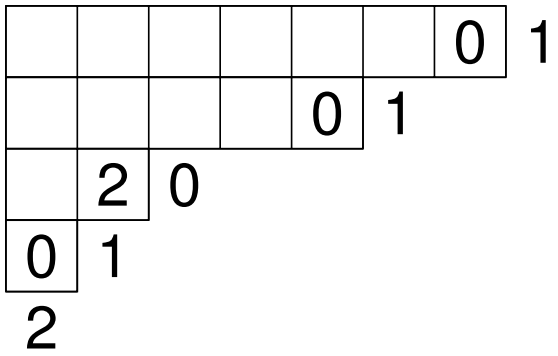


0-signature $- - + -$

reduced 0-signature $- -$

Kleshchev's bijection

e.g. $\lambda = (7, 5, 2, 1)$, $p = 3$



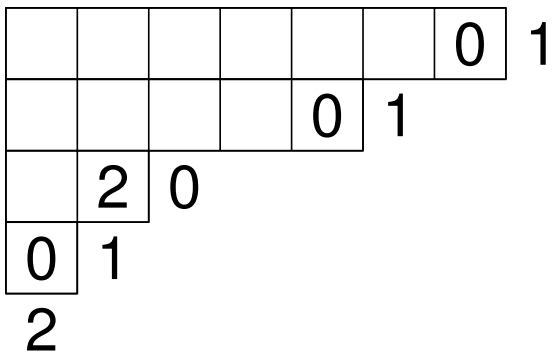
0-signature $- - + -$

reduced 0-signature $- -$

$$\mathrm{soc}(e_0 \mathrm{D}^{\lambda}) = \mathrm{D}^{(7,4,2,1)}$$

Kleshchev's bijection

e.g. $\lambda = (7, 5, 2, 1)$, $p = 3$



0-signature $- - + -$

reduced 0-signature $- -$

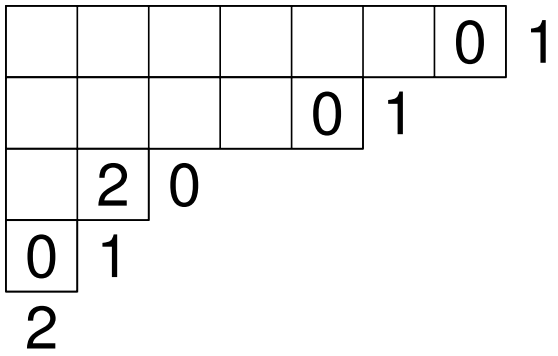
$$\mathrm{soc}(e_0 \mathrm{D}^\lambda) = \mathrm{D}^{(7,4,2,1)}$$

Kleshchev's bijection: Find $i_1, \ldots, i_n$ such that

$$\tilde{e}_{i_1} \ldots \tilde{e}_{i_n} \lambda = \varnothing.$$

Kleshchev's bijection

e.g. $\lambda = (7,5,2,1)$, $p = 3$



0-signature $- - + -$

reduced 0-signature $- -$

$$\mathrm{soc}(e_0 \mathrm{D}^{\lambda}) = \mathrm{D}^{(7,4,2,1)}$$

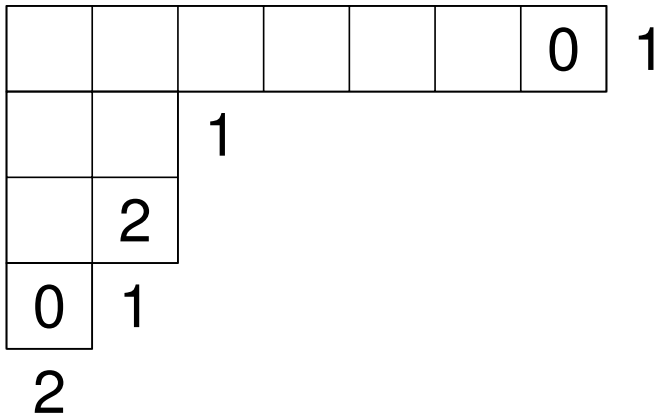
Kleshchev's bijection: Find $i_1, \ldots, i_n$ such that

$$\tilde{e}_{i_1} \ldots \tilde{e}_{i_n} \lambda = \varnothing.$$

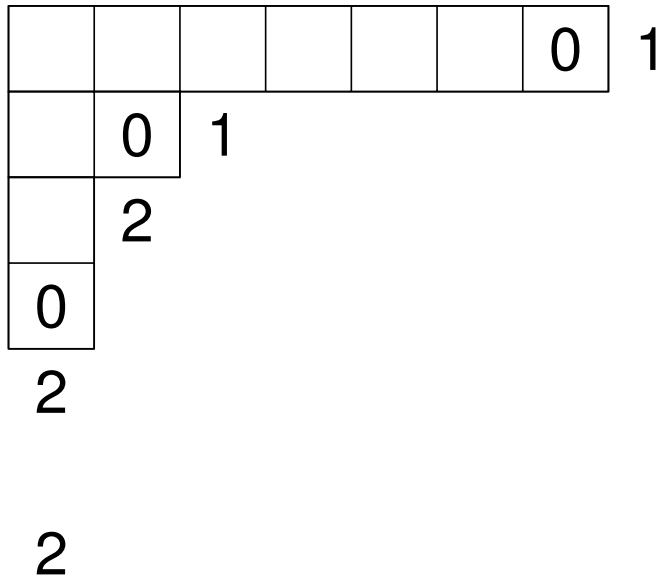
Then $m_p(\lambda)$ is the partition such that

$$\tilde{e}_{-i_1} \ldots \tilde{e}_{-i_n} m_p(\lambda) = \varnothing.$$

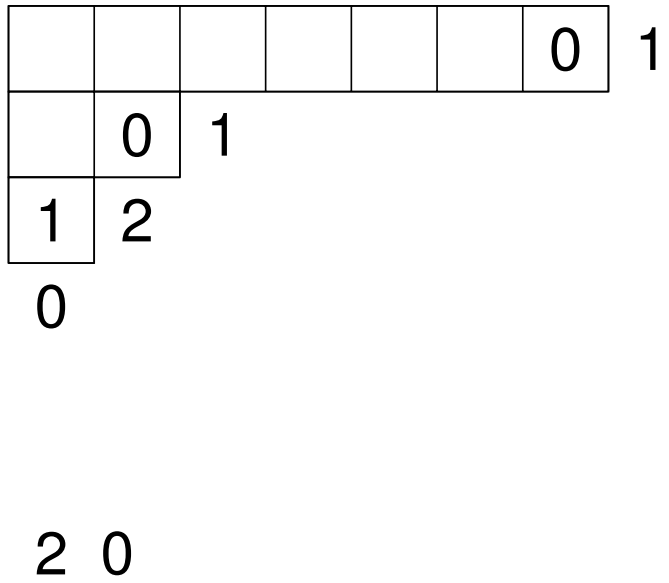
Kleshchev's bijection – example



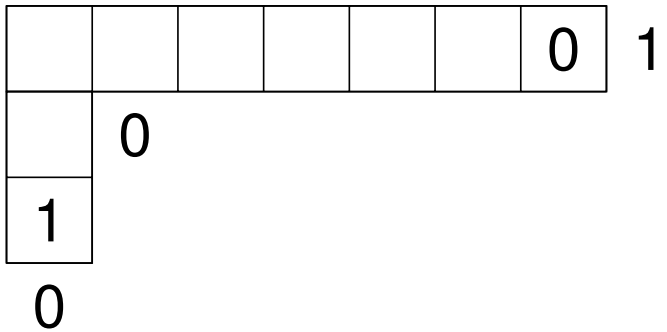
Kleshchev's bijection – example



Kleshchev's bijection – example

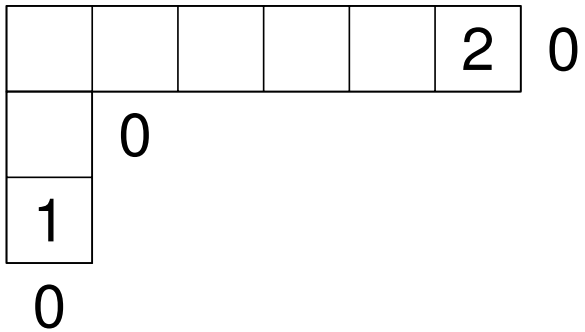


Kleshchev's bijection – example



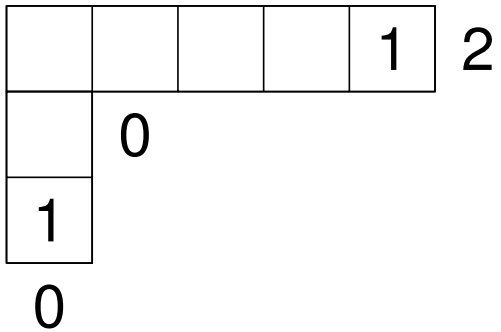
2 0 0

Kleshchev's bijection – example



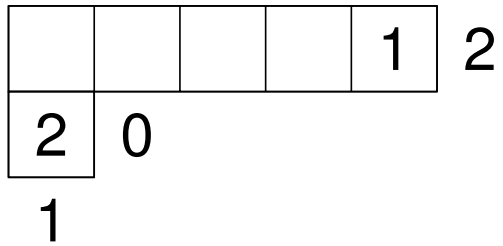
2 0 0 0

Kleshchev's bijection – example



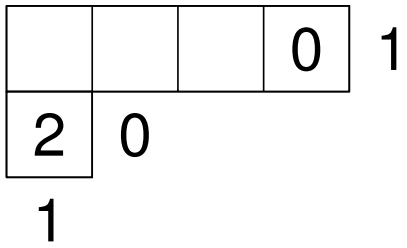
2 0 0 0 2

Kleshchev's bijection – example



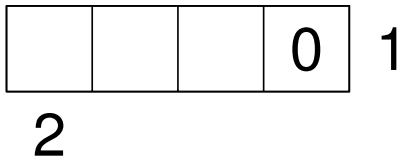
2 0 0 0 2 1

Kleshchev's bijection – example



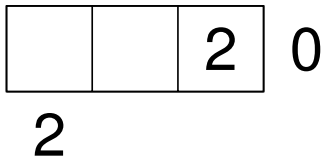
2 0 0 0 2 1 1

Kleshchev's bijection – example



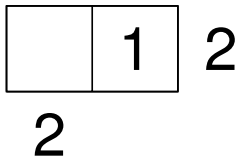
2 0 0 0 2 1 1 2

Kleshchev's bijection – example



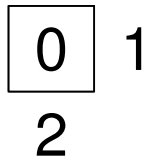
2 0 0 0 2 1 1 2 0

Kleshchev's bijection – example



2 0 0 0 2 1 1 2 0 2

Kleshchev's bijection – example



2 0 0 0 2 1 1 2 0 2 1

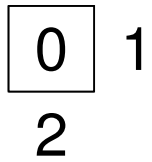
Kleshchev's bijection – example

2 0 0 0 2 1 1 2 0 2 1 0

Kleshchev's bijection – example

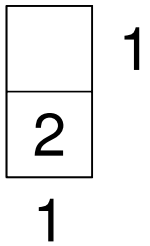
1 0 0 0 1 2 2 1 0 1 2 0

Kleshchev's bijection – example



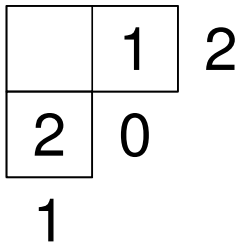
1 0 0 0 1 2 2 1 0 1 2

Kleshchev's bijection – example



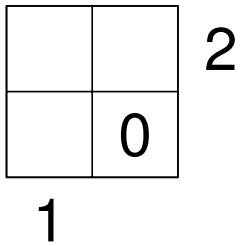
1 0 0 0 1 2 2 1 0 1

Kleshchev's bijection – example



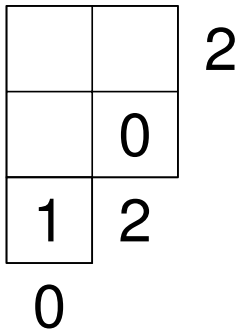
1 0 0 0 1 2 2 1 0

Kleshchev's bijection – example



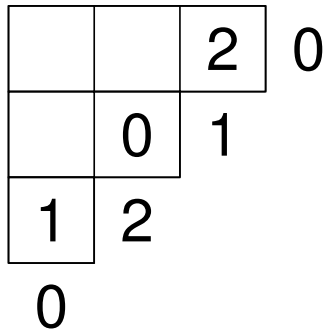
1 0 0 0 1 2 2 1

Kleshchev's bijection – example



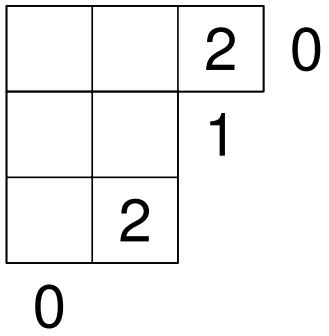
1 0 0 0 1 2 2

Kleshchev's bijection – example



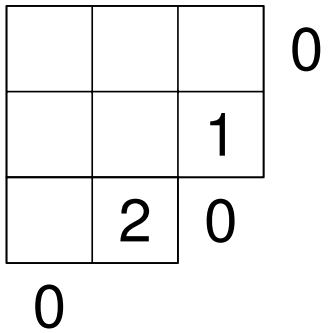
1 0 0 0 1 2

Kleshchev's bijection – example



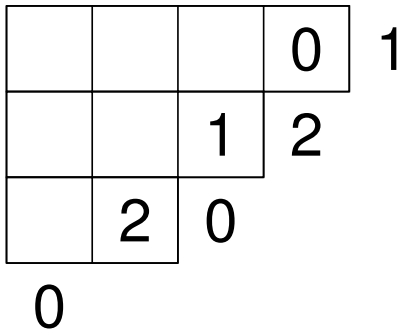
1 0 0 0 1

Kleshchev's bijection – example



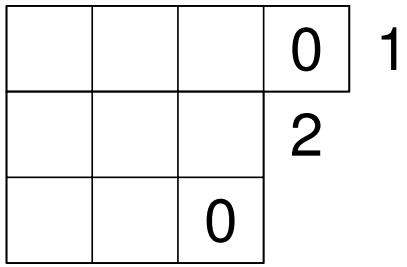
1 0 0 0

Kleshchev's bijection – example

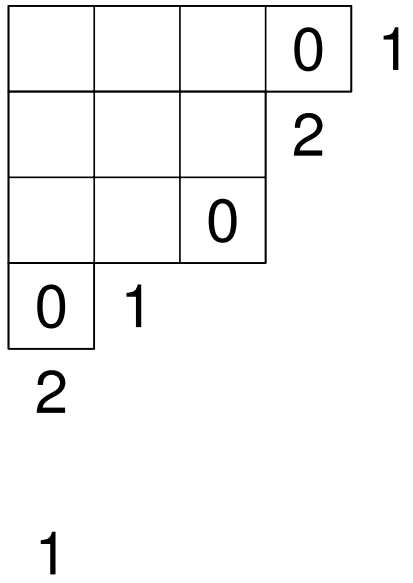


1 0 0

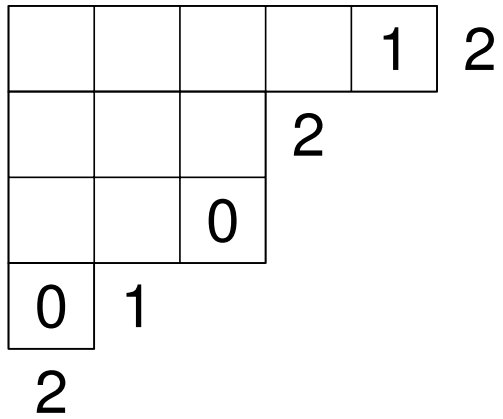
Kleshchev's bijection – example



Kleshchev's bijection – example



Kleshchev's bijection – example



$$m_3(7, 2^2, 1) = (5, 3^2, 1).$$

Brundan–Kujawa–Serganova bijection

Brundan–Kujawa–Serganova bijection

Signed composition: sequence of integers which is eventually zero.

Brundan–Kujawa–Serganova bijection

Signed composition: sequence of integers which is eventually zero.

Resolve a signed composition by making moves as follows:

Brundan–Kujawa–Serganova bijection

Signed composition: sequence of integers which is eventually zero.

Resolve a signed composition by making moves as follows: whenever $\lambda_i < 0 \leqslant \lambda_{i+1}$, replace λ_i, λ_{i+1} with

$$\begin{array}{ll} \lambda_{i+1}, \lambda_i & \text{if } \lambda_{i+1} \equiv \lambda_i \pmod{p} \\ \lambda_{i+1}+1, \lambda_i+1 & \text{if } \lambda_{i+1} \not\equiv \lambda_i \pmod{p}. \end{array}$$

Brundan–Kujawa–Serganova bijection

Signed composition: sequence of integers which is eventually zero.

Resolve a signed composition by making moves as follows: whenever $\lambda_i < 0 \leqslant \lambda_{i+1}$, replace λ_i, λ_{i+1} with

$$\begin{array}{ll} \lambda_{i+1}, \lambda_i & \text{if } \lambda_{i+1} \equiv \lambda_i \pmod{p} \\ \lambda_{i+1}+1, \lambda_i+1 & \text{if } \lambda_{i+1} \not\equiv \lambda_i \pmod{p}. \end{array}$$

e.g. $(\ldots, -3, 4 \ldots) \mapsto (\ldots, 5, -2, \ldots)$ if $p \neq 7$.

Brundan–Kujawa–Serganova bijection

Signed composition: sequence of integers which is eventually zero.

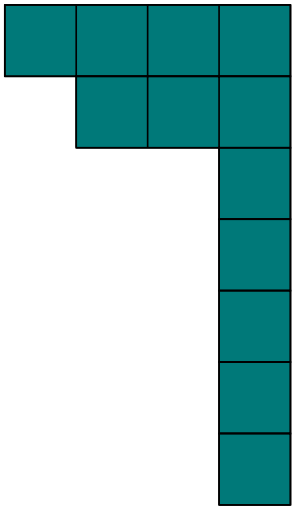
Resolve a signed composition by making moves as follows: whenever $\lambda_i < 0 \leqslant \lambda_{i+1}$, replace λ_i, λ_{i+1} with

$$\begin{array}{ll} \lambda_{i+1}, \lambda_i & \text{if } \lambda_{i+1} \equiv \lambda_i \pmod{p} \\ \lambda_{i+1}+1, \lambda_i+1 & \text{if } \lambda_{i+1} \not\equiv \lambda_i \pmod{p}. \end{array}$$

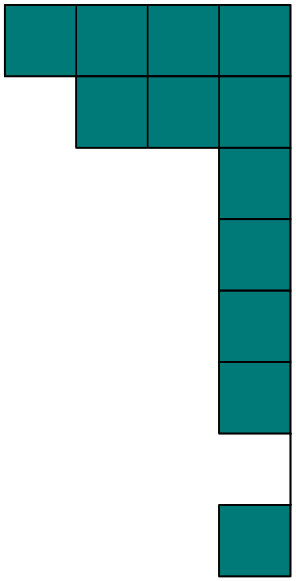
e.g. $(\dots, -3, 4 \dots) \mapsto (\dots, 5, -2, \dots)$ if $p \neq 7$.

BKS bijection: $m_p(\lambda)'$ is the resolution of $-(\lambda')$.

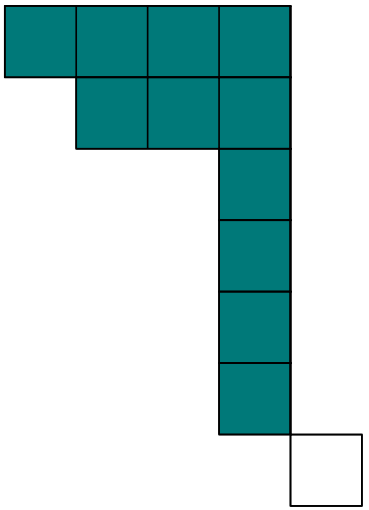
BKS bijection – example



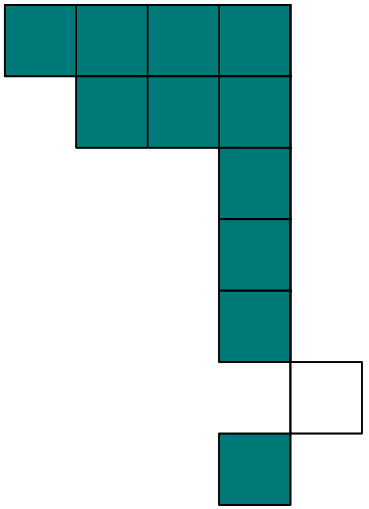
BKS bijection – example



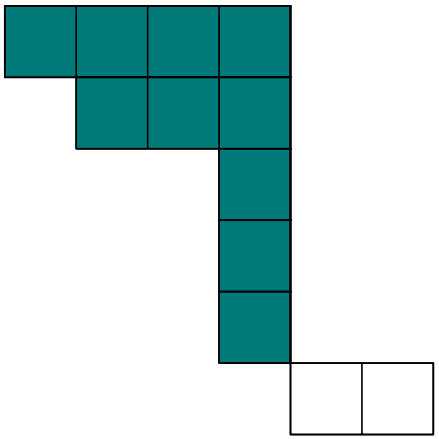
BKS bijection – example



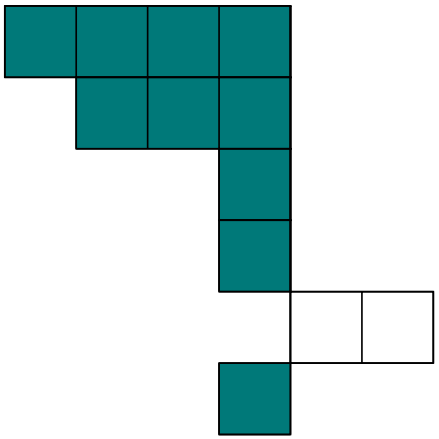
BKS bijection – example



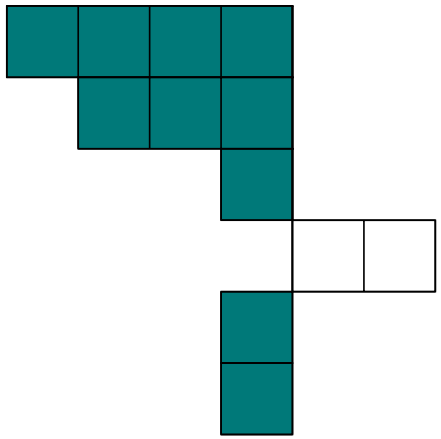
BKS bijection – example



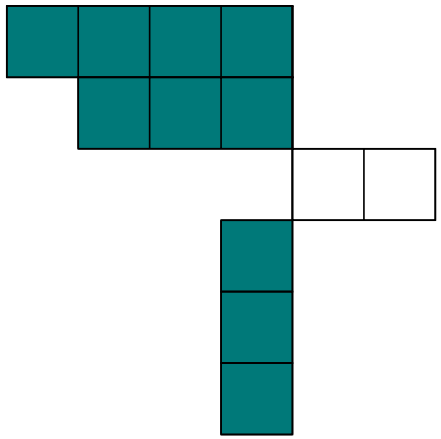
BKS bijection – example



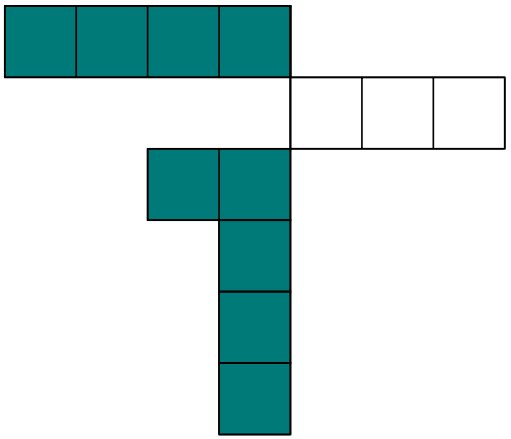
BKS bijection – example



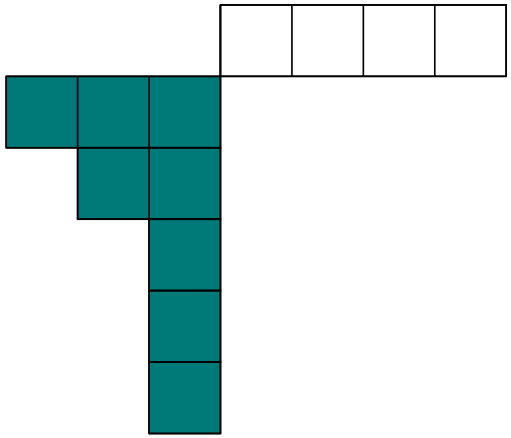
BKS bijection – example



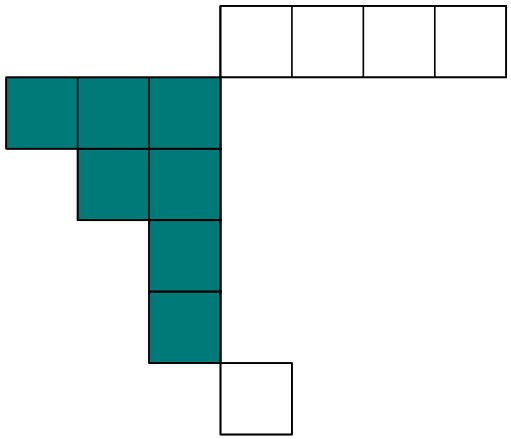
BKS bijection – example



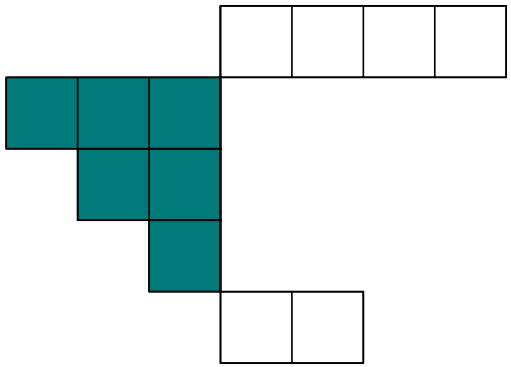
BKS bijection – example



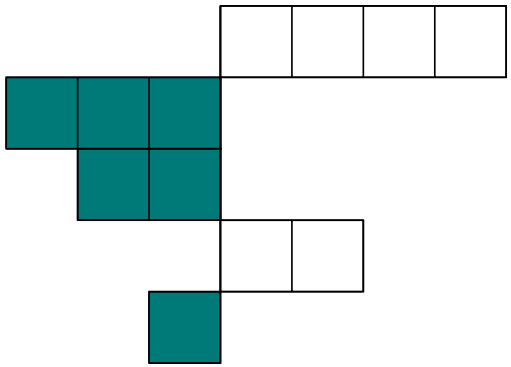
BKS bijection – example



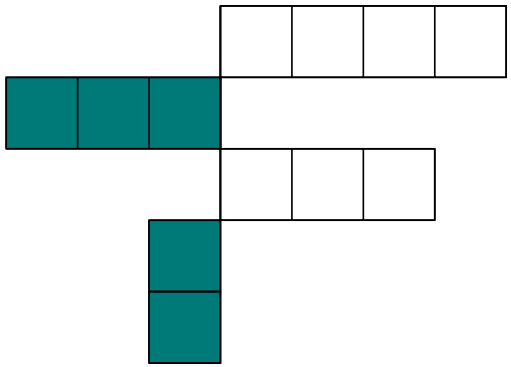
BKS bijection – example



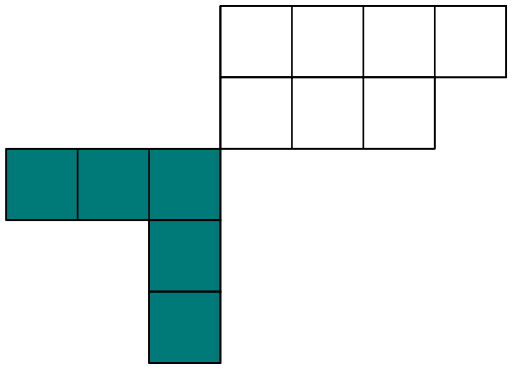
BKS bijection – example



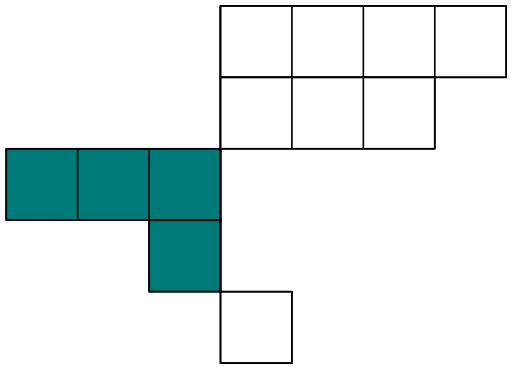
BKS bijection – example



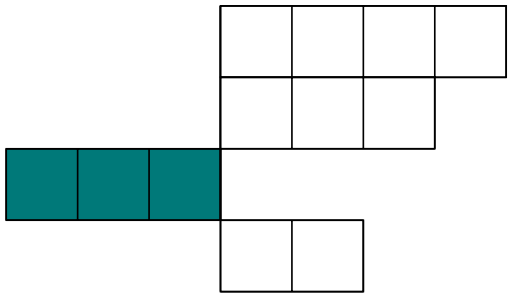
BKS bijection – example



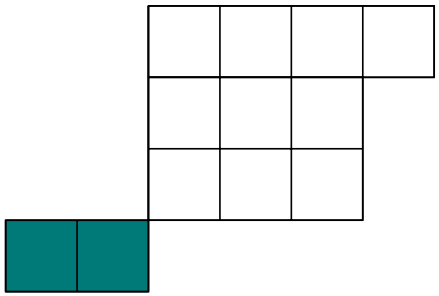
BKS bijection – example



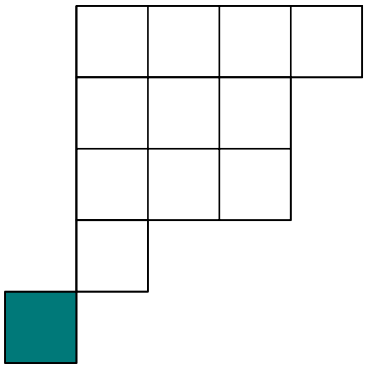
BKS bijection – example



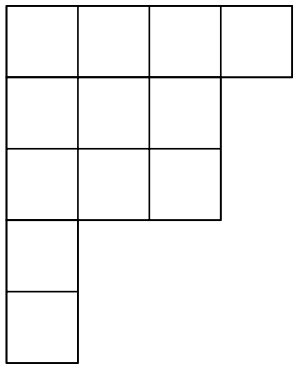
BKS bijection – example



BKS bijection – example



BKS bijection – example



Crystals

Crystals

Recall that $\tilde{e}_i\lambda$ is obtained by removing the good i-node from λ.

Crystals

Recall that $\tilde{e}_i\lambda$ is obtained by removing the good i-node from λ.

Define a labelled directed graph B_{reg}:

Crystals

Recall that $\tilde{e}_i\lambda$ is obtained by removing the good i-node from λ.

Define a labelled directed graph B_{reg}:

vertices: p-regular partitions

Crystals

Recall that $\tilde{e}_i\lambda$ is obtained by removing the good i-node from λ.

Define a labelled directed graph B_{reg}:

vertices: p-regular partitions

arrows: $\lambda \xrightarrow{i} \mu$ when $\lambda = \tilde{e}_i\mu$, for $i \in \mathbb{Z}/p\mathbb{Z}$.

Crystals

Recall that $\tilde{e}_i\lambda$ is obtained by removing the good i-node from λ.

Define a labelled directed graph B_{reg}:

vertices: p-regular partitions

arrows: $\lambda \xrightarrow{i} \mu$ when $\lambda = \tilde{e}_i\mu$, for $i \in \mathbb{Z}/p\mathbb{Z}$.

B_{reg} is isomorphic to the crystal graph for the highest-weight crystal $B(\Lambda_0)$ for $\widehat{\mathfrak{sl}}_p$ (Misra–Miwa 1990). Kleshchev's bijection says that m_p is a "signed automorphism" of this graph, i.e.

$$\lambda \xrightarrow{i} \mu \qquad \Longleftrightarrow \qquad m_p(\lambda) \xrightarrow{-i} m_p(\mu).$$

Crystals

$$\lambda \xrightarrow{i} \mu \qquad \Longleftrightarrow \qquad m_p(\lambda) \xrightarrow{-i} m_p(\mu).$$

Crystals

$$\lambda \xrightarrow{i} \mu \qquad \Longleftrightarrow \qquad m_p(\lambda) \xrightarrow{-i} m_p(\mu).$$

Now change the definition of i-signature by reading from bottom to top.

Crystals

$$\lambda \xrightarrow{i} \mu \qquad \Longleftrightarrow \qquad m_p(\lambda) \xrightarrow{-i} m_p(\mu).$$

Now change the definition of i-signature by reading from bottom to top. Get a new family of functions $\tilde{e}_i$, and a graph B_{rest}:

Crystals

$$\lambda \xrightarrow{i} \mu \qquad \Longleftrightarrow \qquad m_p(\lambda) \xrightarrow{-i} m_p(\mu).$$

Now change the definition of i-signature by reading from bottom to top. Get a new family of functions $\tilde{e}_i$, and a graph B_{rest}:

vertices: p-restricted partitions

Crystals

$$\lambda \xrightarrow{i} \mu \qquad \Longleftrightarrow \qquad m_p(\lambda) \xrightarrow{-i} m_p(\mu).$$

Now change the definition of i-signature by reading from bottom to top. Get a new family of functions $\tilde{e}_i$, and a graph B_{rest}:

vertices: p-restricted partitions ($\lambda_r - \lambda_{r+1} < p$ for all r)

Crystals

$$\lambda \xrightarrow{i} \mu \qquad \Longleftrightarrow \qquad m_p(\lambda) \xrightarrow{-i} m_p(\mu).$$

Now change the definition of i-signature by reading from bottom to top. Get a new family of functions $\tilde{e}_i$, and a graph B_{rest}:

vertices: p-restricted partitions ($\lambda_r - \lambda_{r+1} < p$ for all r)

arrows: $\lambda \xrightarrow{i} \mu$ when $\lambda = \tilde{e}_i \mu$.

Crystals

$$\lambda \xrightarrow{i} \mu \qquad \Longleftrightarrow \qquad m_p(\lambda) \xrightarrow{-i} m_p(\mu).$$

Now change the definition of i-signature by reading from bottom to top. Get a new family of functions $\tilde{e}_i$, and a graph B_{rest}:

vertices: p-restricted partitions ($\lambda_r - \lambda_{r+1} < p$ for all r)

arrows: $\lambda \xrightarrow{i} \mu$ when $\lambda = \tilde{e}_i \mu$.

B_{rest} is also isomorphic to $B(\Lambda_0)$.

Crystals

$$\lambda \xrightarrow{i} \mu \qquad \Longleftrightarrow \qquad m_p(\lambda) \xrightarrow{-i} m_p(\mu).$$

Now change the definition of i-signature by reading from bottom to top. Get a new family of functions $\tilde{e}_i$, and a graph B_{rest}:

vertices: p-restricted partitions ($\lambda_r - \lambda_{r+1} < p$ for all r)

arrows: $\lambda \xrightarrow{i} \mu$ when $\lambda = \tilde{e}_i \mu$.

B_{rest} is also isomorphic to $B(\Lambda_0)$.

Clearly the map $\lambda \mapsto \lambda'$ is a signed isomorphism $B_{\text{rest}} \to B_{\text{reg}}$.

$$\lambda \xrightarrow{i} \mu \qquad \Longleftrightarrow \qquad m_p(\lambda) \xrightarrow{-i} m_p(\mu).$$

Now change the definition of i-signature by reading from bottom to top. Get a new family of functions $\tilde{e}_i$, and a graph B_{rest}:

vertices: p-restricted partitions ($\lambda_r - \lambda_{r+1} < p$ for all r)

arrows: $\lambda \xrightarrow{i} \mu$ when $\lambda = \tilde{e}_i \mu$.

B_{rest} is also isomorphic to $B(\Lambda_0)$.

Clearly the map $\lambda \mapsto \lambda'$ is a signed isomorphism $B_{\text{rest}} \to B_{\text{reg}}$. Hence the map $\lambda \mapsto m_p(\lambda')$ is an isomorphism $B_{\text{rest}} \to B_{\text{reg}}$.

More crystals

More crystals

If $p \geqslant 3$, then B_{reg} and B_{rest} are members of an infinite family of crystal graphs.

More crystals

If $p \geqslant 3$, then B_{reg} and B_{rest} are members of an infinite family of crystal graphs.
Suppose (r, c) is a node of λ.

More crystals

If $p \geqslant 3$, then B_{reg} and B_{rest} are members of an infinite family of crystal graphs.
Suppose (r, c) is a node of λ.

arm length: $a_{(r,c)} = \lambda_r - c$

leg length: $l_{(r,c)} = \lambda'_c - r$

hook length: $h_{(r,c)} = a_{(r,c)} + l_{(r,c)} + 1$.

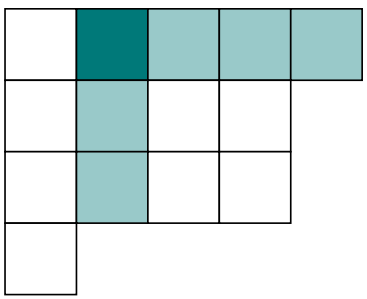
More crystals

If $p \geqslant 3$, then B_{reg} and B_{rest} are members of an infinite family of crystal graphs.
Suppose (r, c) is a node of λ.

arm length: $a_{(r,c)} = \lambda_r - c$

leg length: $l_{(r,c)} = \lambda'_c - r$

hook length: $h_{(r,c)} = a_{(r,c)} + l_{(r,c)} + 1$.

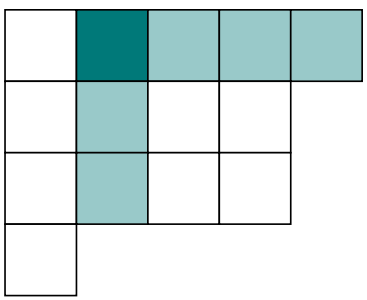


$a_{(1,2)} = 3$
$l_{(1,2)} = 2$
$h_{(1,2)} = 6$.

More crystals

If $p \geqslant 3$, then B_{reg} and B_{rest} are members of an infinite family of crystal graphs.
Suppose (r, c) is a node of λ.

arm length: $a_{(r,c)} = \lambda_r - c$
leg length: $l_{(r,c)} = \lambda'_c - r$
hook length: $h_{(r,c)} = a_{(r,c)} + l_{(r,c)} + 1$.



$a_{(1,2)} = 3$
$l_{(1,2)} = 2$
$h_{(1,2)} = 6$.

Arm sequence: Sequence $A = (A_1, A_2, \dots)$ such that

- $r - 1 \leqslant A_r \leqslant (p-1)r$
- $A_{r+s} \in \{A_r + A_s, A_r + A_s + 1\}$.

More crystals

Arm sequence: Sequence $A = (A_1, A_2, \dots)$ such that

- $r - 1 \leqslant A_r \leqslant (p-1)r$
- $A_{r+s} \in \{A_r + A_s, A_r + A_s + 1\}$.

More crystals

Arm sequence: Sequence $A = (A_1, A_2, \dots)$ such that

- $r - 1 \leqslant A_r \leqslant (p-1)r$
- $A_{r+s} \in \{A_r + A_s, A_r + A_s + 1\}$.

Say λ is A-regular if it has no node with hook length pr and arm length A_r for any r.

More crystals

Arm sequence: Sequence $A = (A_1, A_2, \dots)$ such that

- $r - 1 \leqslant A_r \leqslant (p-1)r$
- $A_{r+s} \in \{A_r + A_s, A_r + A_s + 1\}$.

Say λ is A-regular if it has no node with hook length pr and arm length A_r for any r.
For an appropriate redefinition of i-signature, get a graph B_A:

More crystals

Arm sequence: Sequence $A = (A_1, A_2, \dots)$ such that

- $r-1 \leqslant A_r \leqslant (p-1)r$
- $A_{r+s} \in \{A_r + A_s, A_r + A_s + 1\}$.

Say λ is A-regular if it has no node with hook length pr and arm length A_r for any r.
For an appropriate redefinition of i-signature, get a graph B_A:

vertices: A-regular partitions

More crystals

Arm sequence: Sequence $A = (A_1, A_2, \dots)$ such that

- $r-1 \leqslant A_r \leqslant (p-1)r$
- $A_{r+s} \in \{A_r + A_s, A_r + A_s + 1\}$.

Say λ is A-regular if it has no node with hook length pr and arm length A_r for any r.
For an appropriate redefinition of i-signature, get a graph B_A:

vertices: A-regular partitions

arrows: $\lambda \xrightarrow{i} \mu$ when $\lambda = \tilde{e}_i \mu$.

More crystals

Arm sequence: Sequence $A = (A_1, A_2, \dots)$ such that

- $r-1 \leqslant A_r \leqslant (p-1)r$
- $A_{r+s} \in \{A_r + A_s, A_r + A_s + 1\}$.

Say λ is A-regular if it has no node with hook length pr and arm length A_r for any r.
For an appropriate redefinition of i-signature, get a graph B_A:

vertices: A-regular partitions

arrows: $\lambda \xrightarrow{i} \mu$ when $\lambda = \tilde{e}_i \mu$.

($B_{\text{reg}} = B_{(0,1,2,\dots)}$, $B_{\text{rest}} = B_{(p-1,2(p-1),3(p-1),\dots)}$.)

More crystals

Arm sequence: Sequence $A = (A_1, A_2, \dots)$ such that

- $r-1 \leqslant A_r \leqslant (p-1)r$
- $A_{r+s} \in \{A_r + A_s, A_r + A_s + 1\}$.

Say λ is A-regular if it has no node with hook length pr and arm length A_r for any r.
For an appropriate redefinition of i-signature, get a graph B_A:

vertices: A-regular partitions

arrows: $\lambda \xrightarrow{i} \mu$ when $\lambda = \tilde{e}_i \mu$.

($B_{\text{reg}} = B_{(0,1,2,\dots)}$, $B_{\text{rest}} = B_{(p-1,2(p-1),3(p-1),\dots)}$.)

Theorem (F. 2010): B_A is isomorphic to the crystal graph $B(\Lambda_0)$.

p-regularisation

p-regularisation

James (1976) defined a function

$$\mathrm{reg} : \{\text{partitions of } n\} \longrightarrow \{p\text{-regular partitions of } n\}\,.$$

p-regularisation

James (1976) defined a function

$$\mathrm{reg} : \{\text{partitions of } n\} \longrightarrow \{p\text{-regular partitions of } n\}\,.$$

Ladder in $\mathbb{N}^2$: a line of gradient $p-1$.

p-regularisation

James (1976) defined a function

$$\mathrm{reg} : \{\text{partitions of } n\} \longrightarrow \{p\text{-regular partitions of } n\}\,.$$

Ladder in $\mathbb{N}^2$: a line of gradient $p-1$. λ^{reg} is obtained from λ by moving the nodes to the highest positions in their ladders.

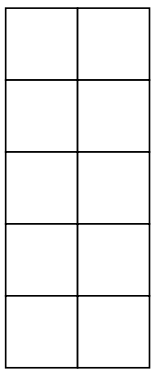
p-regularisation

James (1976) defined a function

$$\mathrm{reg} : \{\text{partitions of } n\} \longrightarrow \{p\text{-regular partitions of } n\}.$$

Ladder in $\mathbb{N}^2$: a line of gradient $p-1$. λ^{reg} is obtained from λ by moving the nodes to the highest positions in their ladders.

e.g. $\lambda = (2^5)$, $p = 3$.



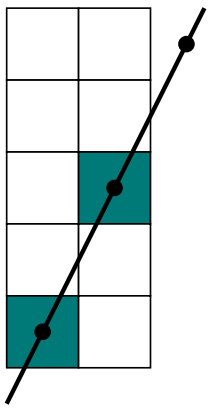
p-regularisation

James (1976) defined a function

$$\mathrm{reg} : \{\text{partitions of } n\} \longrightarrow \{p\text{-regular partitions of } n\}.$$

Ladder in $\mathbb{N}^2$: a line of gradient $p-1$. λ^{reg} is obtained from λ by moving the nodes to the highest positions in their ladders.

e.g. $\lambda = (2^5)$, $p = 3$.



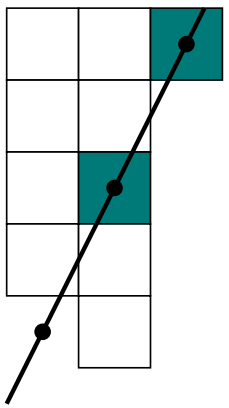
p-regularisation

James (1976) defined a function

$$\mathrm{reg} : \{\text{partitions of } n\} \longrightarrow \{p\text{-regular partitions of } n\}.$$

Ladder in $\mathbb{N}^2$: a line of gradient $p-1$. λ^{reg} is obtained from λ by moving the nodes to the highest positions in their ladders.

e.g. $\lambda = (2^5)$, $p = 3$.



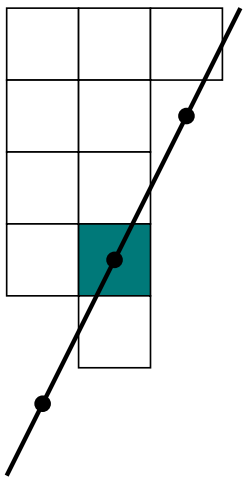
p-regularisation

James (1976) defined a function

$$\mathrm{reg} : \{\text{partitions of } n\} \longrightarrow \{p\text{-regular partitions of } n\}.$$

Ladder in $\mathbb{N}^2$: a line of gradient $p-1$. λ^{reg} is obtained from λ by moving the nodes to the highest positions in their ladders.

e.g. $\lambda = (2^5)$, $p = 3$.



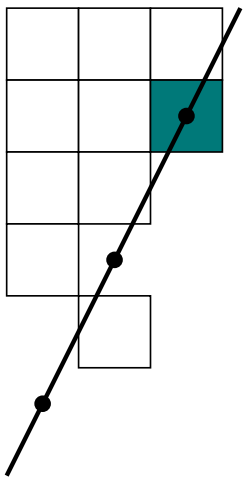
p-regularisation

James (1976) defined a function

$$\mathrm{reg} : \{\text{partitions of } n\} \longrightarrow \{p\text{-regular partitions of } n\}.$$

Ladder in $\mathbb{N}^2$: a line of gradient $p-1$. λ^{reg} is obtained from λ by moving the nodes to the highest positions in their ladders.

e.g. $\lambda = (2^5)$, $p = 3$.



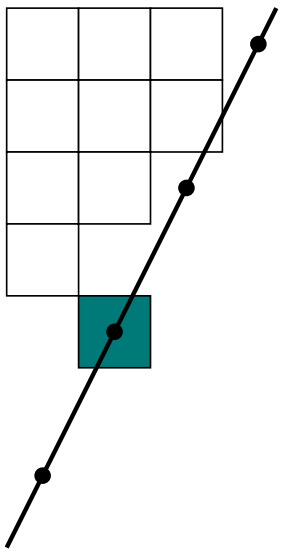
p-regularisation

James (1976) defined a function

$$\mathrm{reg} : \{\text{partitions of } n\} \longrightarrow \{p\text{-regular partitions of } n\}.$$

Ladder in $\mathbb{N}^2$: a line of gradient $p-1$. λ^{reg} is obtained from λ by moving the nodes to the highest positions in their ladders.

e.g. $\lambda = (2^5)$, $p = 3$.



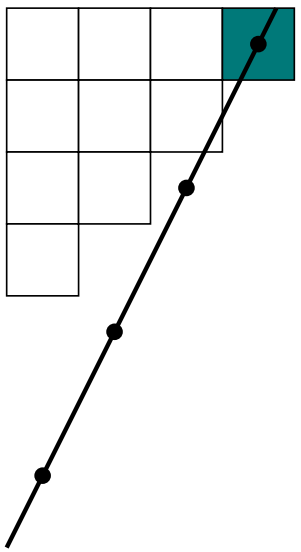
p-regularisation

James (1976) defined a function

$$\mathrm{reg} : \{\text{partitions of } n\} \longrightarrow \{p\text{-regular partitions of } n\}.$$

Ladder in $\mathbb{N}^2$: a line of gradient $p-1$. λ^{reg} is obtained from λ by moving the nodes to the highest positions in their ladders.

e.g. $\lambda = (2^5)$, $p = 3$.



p-regularisation

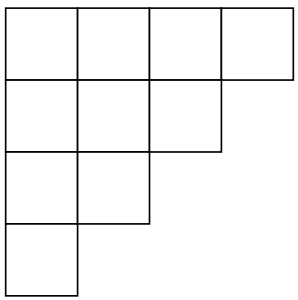
James (1976) defined a function

$$\mathrm{reg} : \{\text{partitions of } n\} \longrightarrow \{p\text{-regular partitions of } n\}.$$

Ladder in $\mathbb{N}^2$: a line of gradient $p-1$. λ^{reg} is obtained from λ by moving the nodes to the highest positions in their ladders.

e.g. $\lambda = (2^5)$, $p = 3$.

$(2^5)^{\mathrm{reg}} = (4, 3, 2, 1)$.



p-regularisation

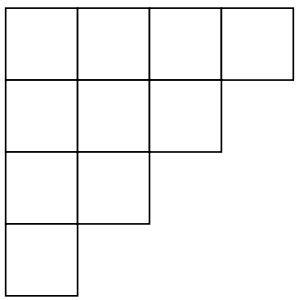
James (1976) defined a function

$$\mathrm{reg} : \{\text{partitions of } n\} \longrightarrow \{p\text{-regular partitions of } n\}.$$

Ladder in $\mathbb{N}^2$: a line of gradient $p-1$. λ^{reg} is obtained from λ by moving the nodes to the highest positions in their ladders.

e.g. $\lambda = (2^5)$, $p = 3$.

$(2^5)^{\mathrm{reg}} = (4,3,2,1)$.



Theorem (James): $[\mathrm{S}^\lambda : \mathrm{D}^{\lambda^{\mathrm{reg}}}] = 1$, and if $[\mathrm{S}^\lambda : \mathrm{D}^\mu] > 0$ then $\mu \trianglerighteq \lambda^{\mathrm{reg}}$.

p-regularisation

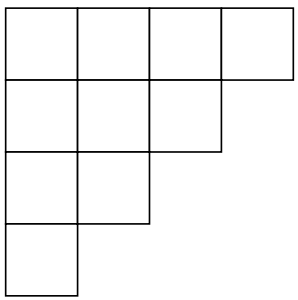
James (1976) defined a function

$$\mathrm{reg} : \{\text{partitions of } n\} \longrightarrow \{p\text{-regular partitions of } n\}.$$

Ladder in $\mathbb{N}^2$: a line of gradient $p-1$. λ^{reg} is obtained from λ by moving the nodes to the highest positions in their ladders.

e.g. $\lambda = (2^5)$, $p = 3$.

$(2^5)^{\mathrm{reg}} = (4,3,2,1)$.



Theorem (James): $[S^\lambda : D^{\lambda^{\mathrm{reg}}}] = 1$, and if $[S^\lambda : D^\mu] > 0$ then $\mu \trianglerighteq \lambda^{\mathrm{reg}}$.

(Dominance order: $\lambda \trianglerighteq \mu$ if $\lambda_1 + \cdots + \lambda_r \geqslant \mu_1 + \cdots + \mu_r$ for all r.)

Crystal isomorphisms

Crystal isomorphisms

Recall $B_{\mathrm{reg}} = B_{(0,1,2,\dots)}$.

Crystal isomorphisms

Recall $B_{\mathrm{reg}} = B_{(0,1,2,\dots)}$.

The crystal $B_{(1,2,3\dots)}$ was introduced by Berg (called the ladder crystal).

Crystal isomorphisms

Recall $B_{\mathrm{reg}} = B_{(0,1,2,\dots)}$.

The crystal $B_{(1,2,3\dots)}$ was introduced by Berg (called the ladder crystal).

Theorem (Berg 2010): The isomorphism $B_{(1,2,3,\dots)} \to B_{(0,1,2,\dots)}$ is given by $\lambda \mapsto \lambda^{\mathrm{reg}}$.

Crystal isomorphisms

Recall $B_{\mathrm{reg}} = B_{(0,1,2,\dots)}$.

The crystal $B_{(1,2,3\dots)}$ was introduced by Berg (called the ladder crystal).

Theorem (Berg 2010): The isomorphism $B_{(1,2,3,\dots)} \to B_{(0,1,2,\dots)}$ is given by $\lambda \mapsto \lambda^{\mathrm{reg}}$.

To generalise this, we need to generalise regularisation ...

(p, c)-regularisation

(p, c)-regularisation

Take $c \in \{1, \ldots, p-1\}$.

(p, c)-regularisation

Take $c \in \{1, \ldots, p-1\}$. (p, c)-ladder: line of gradient $\frac{p-c}{c}$.

(p, c)-regularisation

Take $c \in \{1, \ldots, p-1\}$. (p, c)-ladder: line of gradient $\frac{p-c}{c}$.

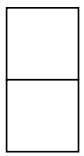
We want to define (p, c)-regularisation.

(p, c)-regularisation

Take $c \in \{1, \ldots, p-1\}$. (p, c)-ladder: line of gradient $\frac{p-c}{c}$.

We want to define (p, c)-regularisation.

e.g. $p = 3$, $c = 2$:

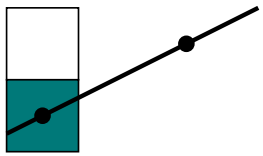


(p, c)-regularisation

Take $c \in \{1, \ldots, p-1\}$. (p, c)-ladder: line of gradient $\frac{p-c}{c}$.

We want to define (p, c)-regularisation.

e.g. $p = 3$, $c = 2$:

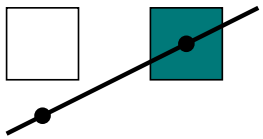


(p, c)-regularisation

Take $c \in \{1, \ldots, p-1\}$. (p, c)-ladder: line of gradient $\frac{p-c}{c}$.

We want to define (p, c)-regularisation.

e.g. $p = 3$, $c = 2$:

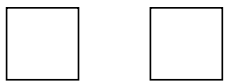


(p, c)-regularisation

Take $c \in \{1, \ldots, p-1\}$. (p, c)-ladder: line of gradient $\frac{p-c}{c}$.

We want to define (p, c)-regularisation.

e.g. $p = 3$, $c = 2$:

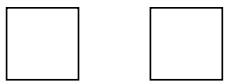


(p, c)-regularisation

Take $c \in \{1, \ldots, p-1\}$. (p, c)-ladder: line of gradient $\frac{p-c}{c}$.

We want to define (p, c)-regularisation.

e.g. $p = 3$, $c = 2$:



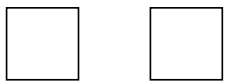
Say that λ, μ are (p, c)-equivalent if they have the same number of nodes in each (p, c)-ladder.

(p, c)-regularisation

Take $c \in \{1, \ldots, p-1\}$. (p, c)-ladder: line of gradient $\frac{p-c}{c}$.

We want to define (p, c)-regularisation.

e.g. $p = 3$, $c = 2$:



Say that λ, μ are (p, c)-equivalent if they have the same number of nodes in each (p, c)-ladder.

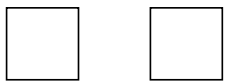
Theorem (Millan Berdasco 2019): Each (p, c)-equivalence class contains a unique most dominant partition.

(p, c)-regularisation

Take $c \in \{1, \ldots, p-1\}$. (p, c)-ladder: line of gradient $\frac{p-c}{c}$.

We want to define (p, c)-regularisation.

e.g. $p = 3$, $c = 2$:



Say that λ, μ are (p, c)-equivalent if they have the same number of nodes in each (p, c)-ladder.

Theorem (Millan Berdasco 2019): Each (p, c)-equivalence class contains a unique most dominant partition.

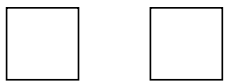
(Case $c = p-1$ is due to Berg.)

(p, c)-regularisation

Take $c \in \{1, \ldots, p-1\}$. (p, c)-ladder: line of gradient $\frac{p-c}{c}$.

We want to define (p, c)-regularisation.

e.g. $p = 3$, $c = 2$:



Say that λ, μ are (p, c)-equivalent if they have the same number of nodes in each (p, c)-ladder.

Theorem (Millan Berdasco 2019): Each (p, c)-equivalence class contains a unique most dominant partition.

(Case $c = p-1$ is due to Berg.)

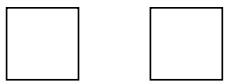
Define the (p, c)-regularisation of λ to be the most dominant partition in the (p, c)-class containing λ.

(p, c)-regularisation

Take $c \in \{1, \ldots, p-1\}$. (p, c)-ladder: line of gradient $\frac{p-c}{c}$.

We want to define (p, c)-regularisation.

e.g. $p = 3$, $c = 2$:



Say that λ, μ are (p, c)-equivalent if they have the same number of nodes in each (p, c)-ladder.

Theorem (Millan Berdasco 2019): Each (p, c)-equivalence class contains a unique most dominant partition.

(Case $c = p-1$ is due to Berg.)

Define the (p, c)-regularisation of λ to be the most dominant partition in the (p, c)-class containing λ.

(James-type decomposition number theorem.)

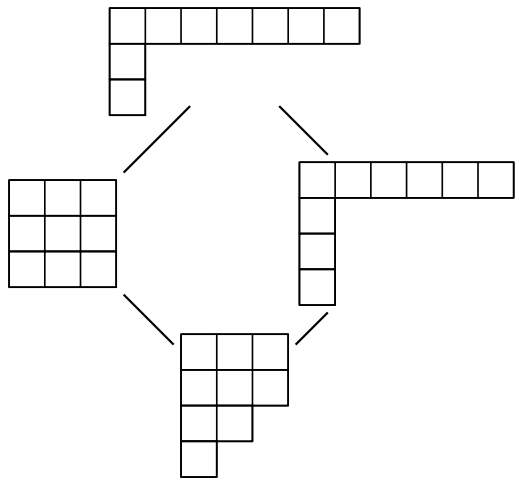
(p, c)-regularisation

Theorem (Millan Berdasco 2019): Each (p, c)-equivalence class of λ contains a unique most dominant partition.

(p, c)-regularisation

Theorem (Millan Berdasco 2019): Each (p, c)-equivalence class of λ contains a unique most dominant partition.

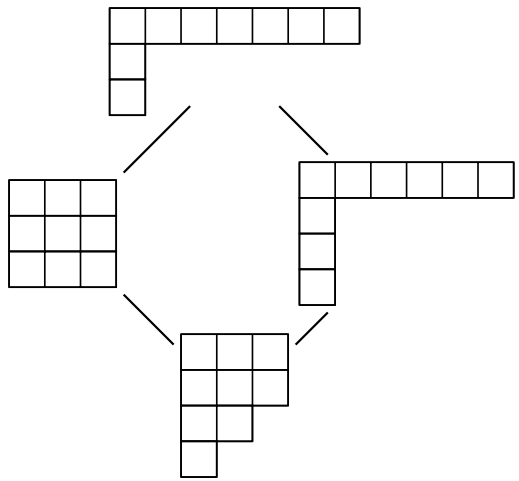
e.g. $p = 3$, $c = 2$, $\lambda = (3^3)$.



(p, c)-regularisation

Theorem (Millan Berdasco 2019): Each (p, c)-equivalence class of λ contains a unique most dominant partition.

e.g. $p = 3$, $c = 2$, $\lambda = (3^3)$.

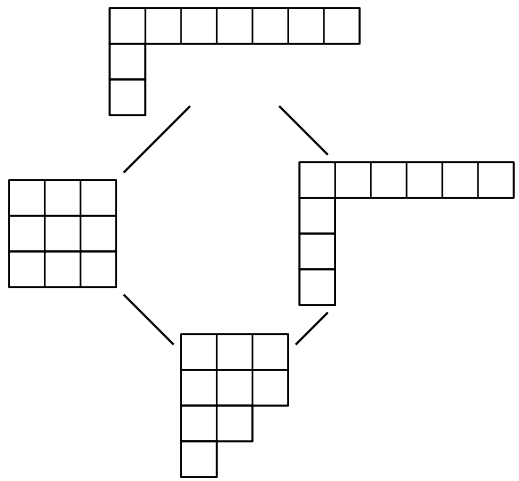


Millan Berdasco gives an algorithm to compute the (p, c)-regularisation using the abacus.

(p, c)-regularisation

Theorem (Millan Berdasco 2019): Each (p, c)-equivalence class of λ contains a unique most dominant partition.

e.g. $p = 3$, $c = 2$, $\lambda = (3^3)$.



Millan Berdasco gives an algorithm to compute the (p, c)-regularisation using the abacus.

Theorem (Millan Berdasco 2019): Let $A_c^\flat = (c-1, 2c-1, 3c-1, \dots)$. Then the (p, c)-regularisation of λ is the unique $A_c^\flat$-regular partition in the same (p, c)-class as λ.

(p, c)-regularisation and crystals

$$A_c^\flat = (c-1, 2c-1, 3c-1, \dots)$$

(p, c)-regularisation and crystals

$$A_c^{\flat} = (c-1, 2c-1, 3c-1, \dots)$$
$$A_c^{\sharp} = (c, 2c, 3c, \dots).$$

(p, c)-regularisation and crystals

$$A_c^\flat = (c-1, 2c-1, 3c-1, \dots)$$
$$A_c^\sharp = (c, 2c, 3c, \dots).$$

Theorem (F. 2020): The isomorphism $B_{A_c^\sharp} \to B_{A_c^\flat}$ is given by (p, c)-regularisation.

(p, c)-regularisation and crystals

$$A_c^\flat = (c-1, 2c-1, 3c-1, \dots)$$
$$A_c^\sharp = (c, 2c, 3c, \dots).$$

Theorem (F. 2020): The isomorphism $B_{A_c^\sharp} \to B_{A_c^\flat}$ is given by (p, c)-regularisation.

More generally, take any rational $c \in [1, p-1]$. Define

$$A_c^\flat = (\lceil c-1 \rceil, \lceil 2c-1 \rceil, \lceil 3c-1 \rceil, \dots)$$
$$A_c^\sharp = (\lfloor c \rfloor, \lfloor 2c \rfloor, \lfloor 3c \rfloor, \dots)$$

(p, c)-regularisation and crystals

$$A_c^\flat = (c-1, 2c-1, 3c-1, \dots)$$
$$A_c^\sharp = (c, 2c, 3c, \dots).$$

Theorem (F. 2020): The isomorphism $B_{A_c^\sharp} \to B_{A_c^\flat}$ is given by (p, c)-regularisation.
More generally, take any rational $c \in [1, p-1]$. Define

$$A_c^\flat = (\lceil c-1 \rceil, \lceil 2c-1 \rceil, \lceil 3c-1 \rceil, \dots)$$
$$A_c^\sharp = (\lfloor c \rfloor, \lfloor 2c \rfloor, \lfloor 3c \rfloor, \dots)$$

and define (p, c)-regularisation to be (pd, cd)-regularisation, where d is the denominator of c.

(p, c)-regularisation and crystals

$$A_c^\flat = (c-1, 2c-1, 3c-1, \dots)$$
$$A_c^\sharp = (c, 2c, 3c, \dots).$$

Theorem (F. 2020): The isomorphism $B_{A_c^\sharp} \to B_{A_c^\flat}$ is given by (p, c)-regularisation.

More generally, take any rational $c \in [1, p-1]$. Define

$$A_c^\flat = (\lceil c-1 \rceil, \lceil 2c-1 \rceil, \lceil 3c-1 \rceil, \dots)$$
$$A_c^\sharp = (\lfloor c \rfloor, \lfloor 2c \rfloor, \lfloor 3c \rfloor, \dots)$$

and define (p, c)-regularisation to be (pd, cd)-regularisation, where d is the denominator of c.

$p = 3$, $c = \frac{3}{2}$: the isomorphism

$$B_{(1,3,4,6,7,9,\dots)} \longrightarrow B_{(1,2,4,5,7,8,\dots)}.$$

is given by $(3, \frac{3}{2})$-regularisation $(= (6, 3)$-regularisation).

A new bijection

A new bijection

Recall: $m_p(\lambda)$ obtained by replacing λ with λ', and then applying the isomorphism $B_{\text{rest}} \to B_{\text{reg}}$.

A new bijection

Recall: $m_p(\lambda)$ obtained by replacing λ with λ', and then applying the isomorphism $B_{\text{rest}} \to B_{\text{reg}}$. We construct this isomorphism as a composition of isomorphisms between crystals B_A.

A new bijection

Recall: $m_p(\lambda)$ obtained by replacing λ with λ', and then applying the isomorphism $B_{\text{rest}} \to B_{\text{reg}}$. We construct this isomorphism as a composition of isomorphisms between crystals B_A.

Informally: Start with λ', and apply (p, c)-regularisation for all rational numbers $c \in [1, p-1]$ in decreasing order.

A new bijection

Recall: $m_p(\lambda)$ obtained by replacing λ with λ', and then applying the isomorphism $B_{\text{rest}} \to B_{\text{reg}}$. We construct this isomorphism as a composition of isomorphisms between crystals B_A.

Informally: Start with λ', and apply (p, c)-regularisation for all rational numbers $c \in [1, p-1]$ in decreasing order.

Formally:

A new bijection

Recall: $m_p(\lambda)$ obtained by replacing λ with λ', and then applying the isomorphism $B_{\text{rest}} \to B_{\text{reg}}$. We construct this isomorphism as a composition of isomorphisms between crystals B_A.

Informally: Start with λ', and apply (p, c)-regularisation for all rational numbers $c \in [1, p-1]$ in decreasing order.

Formally:

1. Let $\mu = \lambda'$, and set $b = p-1$.

A new bijection

Recall: $m_p(\lambda)$ obtained by replacing λ with λ', and then applying the isomorphism $B_{\text{rest}} \to B_{\text{reg}}$. We construct this isomorphism as a composition of isomorphisms between crystals B_A.

Informally: Start with λ', and apply (p, c)-regularisation for all rational numbers $c \in [1, p-1]$ in decreasing order.

Formally:

1. Let $\mu = \lambda'$, and set $b = p-1$.
2. Let c be the largest rational number in $[1, b]$ such that μ is not $A_c^\flat$-regular.

A new bijection

Recall: $m_p(\lambda)$ obtained by replacing λ with λ', and then applying the isomorphism $B_{\text{rest}} \to B_{\text{reg}}$. We construct this isomorphism as a composition of isomorphisms between crystals B_A.

Informally: Start with λ', and apply (p, c)-regularisation for all rational numbers $c \in [1, p-1]$ in decreasing order.

Formally:

1. Let $\mu = \lambda'$, and set $b = p-1$.
2. Let c be the largest rational number in $[1, b]$ such that μ is not $A_c^\flat$-regular. Replace μ with its (p, c)-regularisation, and replace b with c.

A new bijection

Recall: $m_p(\lambda)$ obtained by replacing λ with λ', and then applying the isomorphism $B_{\text{rest}} \to B_{\text{reg}}$. We construct this isomorphism as a composition of isomorphisms between crystals B_A.

Informally: Start with λ', and apply (p, c)-regularisation for all rational numbers $c \in [1, p-1]$ in decreasing order.

Formally:

1. Let $\mu = \lambda'$, and set $b = p-1$.
2. Let c be the largest rational number in $[1, b]$ such that μ is not $A_c^\flat$-regular. Replace μ with its (p, c)-regularisation, and replace b with c.
3. Repeat step 2 until μ is $A_c^\flat$-regular for all $c \in [1, b]$.

A new bijection

Recall: $m_p(\lambda)$ obtained by replacing λ with λ', and then applying the isomorphism $B_{\text{rest}} \to B_{\text{reg}}$. We construct this isomorphism as a composition of isomorphisms between crystals B_A.

Informally: Start with λ', and apply (p, c)-regularisation for all rational numbers $c \in [1, p-1]$ in decreasing order.

Formally:

1. Let $\mu = \lambda'$, and set $b = p-1$.
2. Let c be the largest rational number in $[1, b]$ such that μ is not $A_c^\flat$-regular. Replace μ with its (p, c)-regularisation, and replace b with c.
3. Repeat step 2 until μ is $A_c^\flat$-regular for all $c \in [1, b]$.
4. Output μ.

New bijection – example

New bijection – example

$p = 3$

If we discard partitions of size bigger than 11, then there are six crystals B_A:

$$B_{\text{rest}} = B_{(2,4,6,\dots)}$$

$$B_{(1,3,5,\dots)}$$

$$B_{(1,3,4,\dots)}$$

$$B_{(1,2,4,\dots)}$$

$$B_{(1,2,3,\dots)}$$

$$B_{\text{reg}} = B_{(0,1,2,\dots)}$$

New bijection – example

$p = 3$

If we discard partitions of size bigger than 11, then there are six crystals B_A:

$$
\begin{aligned}
B_{\text{rest}} &= B_{(2,4,6,\dots)} \\
&\quad \downarrow (3,2)\text{-regularisation} \\
&B_{(1,3,5,\dots)} \\
&\quad \downarrow (9,5)\text{-regularisation} \\
&B_{(1,3,4,\dots)} \\
&\quad \downarrow (6,3)\text{-regularisation} \\
&B_{(1,2,4,\dots)} \\
&\quad \downarrow (9,4)\text{-regularisation} \\
&B_{(1,2,3,\dots)} \\
&\quad \downarrow (3,1)\text{-regularisation} \\
B_{\text{reg}} &= B_{(0,1,2,\dots)}
\end{aligned}
$$

Regularisation and the Mullineux map

Regularisation and the Mullineux map

$$m_p(3) = \begin{cases} (3) & (p = 2) \\ (2,1) & (p = 3) \\ (1^3) & (p > 3) \end{cases}$$

Regularisation and the Mullineux map

$$m_p(3) = \begin{cases} (3) & (p = 2) \\ (2,1) & (p = 3) \\ (1^3) & (p > 3) \end{cases}$$

So $m_p(3) = (1^3)^{\mathrm{reg}}$ for all p.

Regularisation and the Mullineux map

$$m_p(3) = \begin{cases} (3) & (p = 2) \\ (2,1) & (p = 3) \\ (1^3) & (p > 3) \end{cases}$$

So $m_p(3) = (1^3)^{\mathrm{reg}}$ for all p.
For which λ is it true that $m_p(\lambda) = (\lambda')^{\mathrm{reg}}$?

Regularisation and the Mullineux map

$$m_p(3) = \begin{cases} (3) & (p = 2) \\ (2,1) & (p = 3) \\ (1^3) & (p > 3) \end{cases}$$

So $m_p(3) = (1^3)^{\mathrm{reg}}$ for all p.
For which λ is it true that $m_p(\lambda) = (\lambda')^{\mathrm{reg}}$?

Say that the (r, c)-hook of λ is shallow if

$$\frac{a_{(r,c)}}{l_{(r,c)}} > p - 1.$$

Regularisation and the Mullineux map

$$m_p(3) = \begin{cases} (3) & (p = 2) \\ (2,1) & (p = 3) \\ (1^3) & (p > 3) \end{cases}$$

So $m_p(3) = (1^3)^{\mathrm{reg}}$ for all p.
For which λ is it true that $m_p(\lambda) = (\lambda')^{\mathrm{reg}}$?

Say that the (r, c)-hook of λ is shallow if

$$\frac{a_{(r,c)}}{l_{(r,c)}} > p - 1.$$

Theorem (Bessenrodt–Olsson–Xu): $m_p(\lambda) \trianglerighteq (\lambda')^{\mathrm{reg}}$, with equality if and only if every hook of length divisible by p is shallow.

Regularisation and the Mullineux map

Regularisation and the Mullineux map

More generally, take λ not necessarily p-regular.

Regularisation and the Mullineux map

More generally, take λ not necessarily p-regular.

When is $m_p(\lambda^{\mathrm{reg}}) = (\lambda')^{\mathrm{reg}}$?

Regularisation and the Mullineux map

More generally, take λ not necessarily p-regular.

When is $m_p(\lambda^{\mathrm{reg}}) = (\lambda')^{\mathrm{reg}}$?

Say that the (r, c)-hook of λ is steep if

$$\frac{l_{(r,c)}}{a_{(r,c)}} > p-1.$$

Regularisation and the Mullineux map

More generally, take λ not necessarily p-regular.

When is $m_p(\lambda^{\mathrm{reg}}) = (\lambda')^{\mathrm{reg}}$?

Say that the (r, c)-hook of λ is steep if

$$\frac{l_{(r,c)}}{a_{(r,c)}} > p-1.$$

Theorem (F. 2008): $m_p(\lambda^{\mathrm{reg}}) \trianglerighteq (\lambda')^{\mathrm{reg}}$, with equality if and only if every hook of length divisible by p is either steep or shallow.

Regularisation and the Mullineux map

More generally, take λ not necessarily p-regular.

When is $m_p(\lambda^{\mathrm{reg}}) = (\lambda')^{\mathrm{reg}}$?

Say that the (r, c)-hook of λ is steep if

$$\frac{l_{(r,c)}}{a_{(r,c)}} > p-1.$$

Theorem (F. 2008): $m_p(\lambda^{\mathrm{reg}}) \trianglerighteq (\lambda')^{\mathrm{reg}}$, with equality if and only if every hook of length divisible by p is either steep or shallow.

New algorithm gives a much easier proof of this.