On a Schur positivity conjecture in multiplicity-free cases

R. Biagioli, *V. Guerrini,* S. Rinaldi

University of Lyon, University of Siena

SLC 75, Bertinoro

Schur positivity

Definition. A symmetric function is *Schur positive* if it is a linear combination with nonnegative coefficients of the Schur functions.

Example

The product $s_\mu s_\nu$ of two Schur functions by means of the classical *Littlewood–Richardson rule* can be written as

$$s_\mu s_\nu = \sum_{\vartheta} c^{\vartheta}_{\mu,\nu}\, s_\vartheta,$$

where the $c^{\vartheta}_{\mu,\nu}$ are nonnegative integers called *Littlewood–Richardson coefficients*.

Schur positivity

In recent years there has been increasing interest in understanding the Schur-positivity of expressions of the form

$$s_\lambda s_\rho - s_\mu s_\nu. \tag{1}$$

Problem [Bergeron, McNamara, 2004]

Given a pair of partitions (μ, ν), which operations can we apply to this pair to yield another pair (λ, ρ) such that (1) is Schur-positive?

Necessary condition. The support of $s_\mu s_\nu$ is contained in the one of $s_\lambda s_\rho$.
Necessary and sufficient condition. For all partitions ϑ,

$$c^{\vartheta}_{\lambda,\rho} \geq c^{\vartheta}_{\mu,\nu}.$$

Fomin-Fulton-Li-Poon Conjecture

Definition. Let (μ, ν) be a pair of partitions having the same number of parts, allowing zero parts. The $*$-operation sends (μ, ν) into the pair (λ, ρ) defined for all k by

$$\lambda_k = \mu_k - k + \#\{l \mid \nu_l - l \geq \mu_k - k\},$$
$$\rho_k = \nu_k - k + 1 + \#\{j \mid \mu_j - j > \nu_k - k\}.$$

Conjecture [Fomin, Fulton, Li, and Poon, 2003]

The expression $s_\lambda s_\rho - s_\mu s_\nu$ is Schur positive, namely, for any partition ϑ,

$$c^{\vartheta}_{\lambda,\rho} \geq c^{\vartheta}_{\mu,\nu}.$$

The $*$-operation

The simplest case

Let $\mu = (a)$ and $\nu = (b)$, with $a > b$. Then,

- $\lambda_1 = a - 1 + \#\{l \,|\, \nu_l - l \geq a - 1\}$
- $\rho_1 = b + \#\{j \,|\, \mu_j - j > b - 1\}$,
- $((a), (b))^* = ((a-1), (b+1))$.

Fomin-Fulton-Li-Poon Conjecture is an instance of the Jacobi-Trudi identity

$$s_{a-1}s_{b+1} - s_a s_b = \det\begin{pmatrix} s_{a-1} & s_a \\ s_b & s_{b+1} \end{pmatrix} = s_{a-1,b+1}.$$

The $*$-operation

The simplest case

Let $\mu = (a)$ and $\nu = (b)$, with $a > b$. Then,

- $\lambda_1 = a - 1 + \#\{l \,|\, \nu_l - l \geq a - 1\}$
- $\rho_1 = b + \#\{j \,|\, \mu_j - j > b - 1\}$,
- $((a),(b))^* = ((a-1),(b+1))$.

Fomin-Fulton-Li-Poon Conjecture is an instance of the Jacobi-Trudi identity

$$s_{a-1}s_{b+1} - s_a s_b = \det \begin{pmatrix} s_{a-1} & s_a \\ s_b & s_{b+1} \end{pmatrix} = s_{a-1,b+1}.$$

The $*$-operation

Some properties of the $*$-operation:

- The partitions λ and ρ are such that $|\lambda| + |\rho| = |\mu| + |\nu|$.
- It is not commutative; in general $(\mu, \nu)^* \neq (\nu, \mu)^*$.
- Fixed points are characterized as the pairs (μ, ν) such that the sequence $\nu_1, \mu_1, \nu_2, \mu_2, \nu_3, \ldots$ is weakly decreasing.
- After applying the $*$-operation a finite number of times it is always reached a fixed point.
- It has an equivalent recursive definition obtained by Bergeron, Biagioli, Rosas.

François Bergeron, Riccardo Biagioli, and Mercedes H. Rosas. Inequalities between Littlewood-Richardson coefficients. J. Combin. Theory Ser. A (2006).

The $*$-operation and the Conjecture

Our goal is to prove the Conjecture for the class of pairs of partitions (μ, ν) such that $s_\mu s_\nu$ is *multiplicity-free*.

Definition. The product $s_\mu s_\nu$ is *multiplicity-free*, if $c^{\vartheta}_{\nu,\mu} \in \{0, 1\}$, for all partitions ϑ.

Example

An instance of the Pieri's rule:

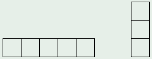
$$s_2\, s_{3,1} = s_{5,1} + s_{4,2} + s_{4,1,1} + s_{3,3} + s_{3,2,1}.$$

Stembridge's characterization

Theorem [Stembridge]

The product $s_\mu s_\nu$ is multiplicity-free if and only if

- *μ or ν is a one-line rectangle (Pieri's rule),or*



(5) (1^3)

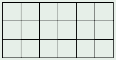
- *μ and ν are rectangles, or*
- *μ is a two-line rectangle and ν a fat hook or vice-versa, or*
- *μ is a rectangle and ν is a near-rectangle or vice-versa.*

Stembridge's characterization

Theorem [Stembridge]

The product $s_\mu s_\nu$ is multiplicity-free if and only if

- *μ or ν is a one-line rectangle (Pieri's rule),or*
- *μ and ν are rectangles, or*



(6^3)

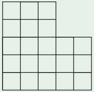
- *μ is a two-line rectangle and ν a fat hook or vice-versa, or*
- *μ is a rectangle and ν is a near-rectangle or vice-versa.*

Stembridge's characterization

Theorem [Stembridge]

The product $s_\mu s_\nu$ is multiplicity-free if and only if

- *μ or ν is a one-line rectangle (Pieri's rule),or*
- *μ and ν are rectangles, or*
- *μ is a two-line rectangle and ν a fat hook or vice-versa, or*



$(5^3, 3^2)$

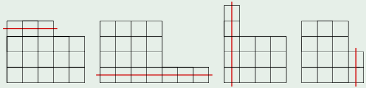
- *μ is a rectangle and ν is a near-rectangle or vice-versa.*

Stembridge's characterization

Theorem [Stembridge]

The product $s_\mu s_\nu$ is multiplicity-free if and only if

- μ *or* ν *is a one-line rectangle (Pieri's rule),or*
- μ *and* ν *are rectangles, or*
- μ *is a two-line rectangle and* ν *a fat hook or vice-versa, or*
- μ *is a rectangle and* ν *is a near-rectangle or vice-versa.*



$(5^3,3)$ $(7,4^3)$ $(4^3,1^2)$ $(4^2,3^2)$

Littlewood-Richardson coefficients

Littlewood-Richardson Rule. The Littlewood-Richardson coefficient $c^{\vartheta}_{\nu,\mu}$ is equal to the number of LR-fillings of shape ϑ/ν and type μ.

LR-filling

A *LR-filling* of shape ϑ/ν of type μ is a semistandard tableau of shape ϑ/ν such that the sequence of multiplicities of the integers $1, 2, \ldots$ that appear in its cells is μ and its reverse reading word is a lattice permutation.
The reverse reading word u is a *lattice permutation* if for any prefix v of u, $|v|_i \geq |v|_{i+1}$, for all i.

Littlewood-Richardson coefficients

Example

Set $\vartheta = (4, 3, 2, 1)$, $\nu = (2, 1)$ and $\mu = (3, 2, 2)$.

$$\begin{array}{|c|ccc} \hline 3 & & & \\ \hline 2 & \multicolumn{1}{c|}{3} & & \\ \hline & \multicolumn{1}{|c|}{1} & \multicolumn{1}{c|}{2} & \\ \hline & & \multicolumn{1}{|c|}{1} & \multicolumn{1}{c|}{1} \\ \hline \end{array} \qquad \begin{array}{|c|ccc} \hline 2 & & & \\ \hline 1 & \multicolumn{1}{c|}{3} & & \\ \hline & \multicolumn{1}{|c|}{2} & \multicolumn{1}{c|}{3} & \\ \hline & & \multicolumn{1}{|c|}{1} & \multicolumn{1}{c|}{1} \\ \hline \end{array}$$

1121323 | *1132312*

YES | NO

- $c^{\vartheta}_{\nu,\mu} = 2$.

Littlewood-Richardson coefficients

Example

Set $\vartheta = (4,3,2,1)$, $\nu = (2,1)$ and $\mu = (3,2,2)$.

$$
\begin{array}{|c|c|c|c|}
\hline
3 \\
\cline{1-1}
2 & 3 \\
\cline{1-3}
 & 1 & 2 \\
\cline{2-4}
 & & 1 & 1 \\
\hline
\end{array}
\qquad\qquad
\begin{array}{|c|c|c|c|}
\hline
2 \\
\cline{1-1}
1 & 3 \\
\cline{1-3}
 & 2 & 3 \\
\cline{2-4}
 & & 1 & 1 \\
\hline
\end{array}
$$

1121323   *1132312*

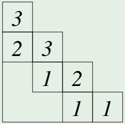
YES   NO

- $c^{\vartheta}_{\nu,\mu} = 2$.

Littlewood-Richardson coefficients

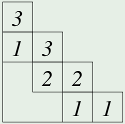
Example

Set $\vartheta = (4,3,2,1)$, $\nu = (2,1)$ and $\mu = (3,2,2)$.



1121323

YES



1122313

YES

- $c^{\vartheta}_{\nu,\mu} = 2$.

Littlewood-Richardson coefficients

Example

Set $\vartheta = (4,3,2,1)$, $\nu = (2,1)$ and $\mu = (3,2,2)$.

$$
\begin{array}{|c|c|c|c|}
\hline
3 & & & \\
\hline
2 & 3 & & \\
\hline
 & 1 & 2 & \\
\hline
 & & 1 & 1 \\
\hline
\end{array}
\qquad\qquad
\begin{array}{|c|c|c|c|}
\hline
3 & & & \\
\hline
1 & 3 & & \\
\hline
 & 2 & 2 & \\
\hline
 & & 1 & 1 \\
\hline
\end{array}
$$

1121323   *1122313*

YES   YES

- $c^{\vartheta}_{\nu,\mu} = 2$.

Our method

- To prove that $c^{\vartheta}_{\rho,\lambda} \geq c^{\vartheta}_{\nu,\mu}$ for all the multiplicity-free pairs of partitions (μ, ν), we need to provide a semi standard tableau of shape ϑ/ρ and type λ for each partition ϑ such that $c^{\vartheta}_{\nu,\mu} = 1$.
- Our aim is to define for each of the Stembridge's cases an algorithm that constructs a LR-filling of shape ϑ/ρ and type λ from the LR-filling of shape ϑ/ν and type μ modifying and moving entries of its cells.
- To calculate $(\lambda, \rho) = (\mu, \nu)^*$, we use the recursive description given by Bergeron, Biagioli and Rosas.

Our method

- To prove that $c^{\vartheta}_{\rho,\lambda} \geq c^{\vartheta}_{\nu,\mu}$ for all the multiplicity-free pairs of partitions (μ, ν), we need to provide a semi standard tableau of shape ϑ/ρ and type λ for each partition ϑ such that $c^{\vartheta}_{\nu,\mu} = 1$.
- Our aim is to define for each of the Stembridge's cases an algorithm that constructs a LR-filling of shape ϑ/ρ and type λ from the LR-filling of shape ϑ/ν and type μ modifying and moving entries of its cells.
- To calculate $(\lambda, \rho) = (\mu, \nu)^*$, we use the recursive description given by Bergeron, Biagioli and Rosas.

Our method

- To prove that $c^{\vartheta}_{\rho,\lambda} \geq c^{\vartheta}_{\nu,\mu}$ for all the multiplicity-free pairs of partitions (μ, ν), we need to provide a semi standard tableau of shape ϑ/ρ and type λ for each partition ϑ such that $c^{\vartheta}_{\nu,\mu} = 1$.
- Our aim is to define for each of the Stembridge's cases an algorithm that constructs a LR-filling of shape ϑ/ρ and type λ from the LR-filling of shape ϑ/ν and type μ modifying and moving entries of its cells.
- To calculate $(\lambda, \rho) = (\mu, \nu)^*$, we use the recursive description given by Bergeron, Biagioli and Rosas.

Recursive definition

- Start from the pair $(0, \nu)$ and calculate $(0, \nu)^* = (\overline{\nu}, \underline{\nu})$.
- Then, the pair $(\lambda, \rho) = (\mu, \nu)^*$, with $\mu \neq 0$, is obtained making $(\overline{\nu}, \underline{\nu})$ grow according to the recursive definition.

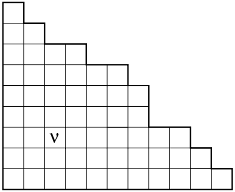
Lemma.

Let ν be any partition. Then

$$\begin{aligned} \underline{\nu} &= \rho(0, \nu) = (\nu_1, \nu_2 - 1, \nu_3 - 2, \ldots, \nu_\kappa - (\kappa - 1)), \\ \overline{\nu}' &= \lambda'(0, \nu) = (\nu_1' - 1, \nu_2' - 2, \ldots, \nu_\kappa' - \kappa), \end{aligned}$$

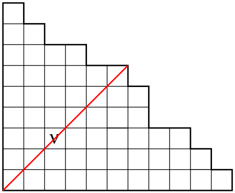
where $\kappa = \max\{i \,|\, \nu_i \geq i\}$.

Recursive definition



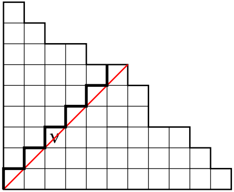
- The Conjecture holds for the pair (μ, ν) if and only if it holds for (ν', μ').
- $c^{\nu}_{\underline{\nu},\overline{\nu}} = 1 = c^{\nu}_{\nu,0}$ and so, the Conjecture holds for the pair $(0, \nu)$.

Recursive definition



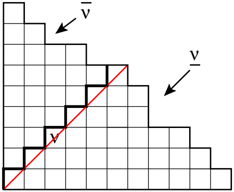
- The Conjecture holds for the pair (μ, ν) if and only if it holds for (ν', μ').
- $c^{\nu}_{\underline{\nu},\overline{\nu}} = 1 = c^{\nu}_{\nu,0}$ and so, the Conjecture holds for the pair $(0, \nu)$.

Recursive definition



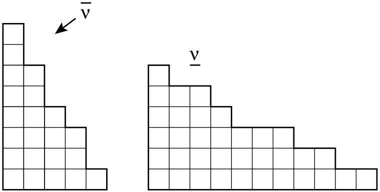
- The Conjecture holds for the pair (μ, ν) if and only if it holds for (ν', μ').
- $c^{\nu}_{\nu,\overline{\nu}} = 1 = c^{\nu}_{\nu,0}$ and so, the Conjecture holds for the pair $(0, \nu)$.

Recursive definition



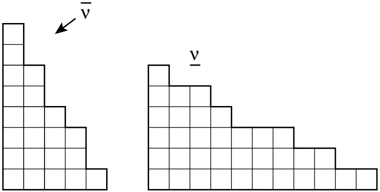
- The Conjecture holds for the pair (μ, ν) if and only if it holds for (ν', μ').
- $c^{\nu}_{\underline{\nu},\overline{\nu}} = 1 = c^{\nu}_{\nu,0}$ and so, the Conjecture holds for the pair $(0, \nu)$.

Recursive definition



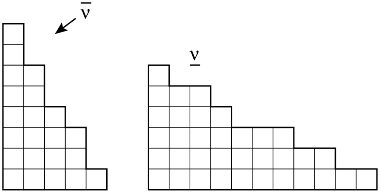
- The Conjecture holds for the pair (μ, ν) if and only if it holds for (ν', μ').
- $c^{\nu}_{\underline{\nu},\overline{\nu}} = 1 = c^{\nu}_{\nu,0}$ and so, the Conjecture holds for the pair $(0, \nu)$.

Recursive definition



- The Conjecture holds for the pair (μ, ν) if and only if it holds for (ν', μ').
- $c^{\nu}_{\underline{\nu},\overline{\nu}} = 1 = c^{\nu}_{\nu,0}$ and so, the Conjecture holds for the pair $(0, \nu)$.

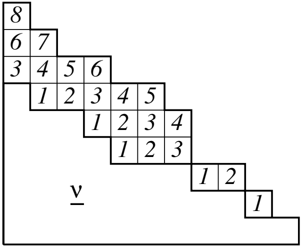
Recursive definition



- The Conjecture holds for the pair (μ, ν) if and only if it holds for (ν', μ').
- $c^{\nu}_{\underline{\nu},\overline{\nu}} = 1 = c^{\nu}_{\nu,0}$ and so, the Conjecture holds for the pair $(\overline{0}, \nu)$.

N-filling

The unique LR-filling of shape $\nu/\underline{\nu}$ and type $\overline{\nu}$, called *natural filling* (briefly N-filling).



General solution method

1. Construct a tableau of shape $\vartheta/\underline{\nu}$, where the cells of ϑ/ν are filled with its unique LR-filling of type μ and those of $\nu/\underline{\nu}$ with the N-filling.
2. Remove the cells of $\rho/\underline{\nu}$ and obtain a new tableau of shape ϑ/ρ, called *initial tableau* and denoted with T^I.
3. Define the tableau $T^{(0)}$ starting from T^I in a way such that its type is λ.
4. A shape-by-shape algorithm is defined to move the entries of $T^{(0)}$ so that the *final tableau* T^F is semi standard and has a lattice permutation as reverse reading word.

Stembridge's characterization

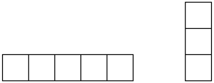
Theorem [Stembridge]

The product $s_\mu s_\nu$ is multiplicity-free if and only if

- μ *or* ν *is a one-line rectangle (Pieri's rule),or*
- μ *and* ν *are rectangles, or*
- μ *is a two-line rectangle and* ν *a fat hook or vice-versa, or*
- μ *is a rectangle and* ν *is a near-rectangle or vice-versa.*

Pieri's rule case

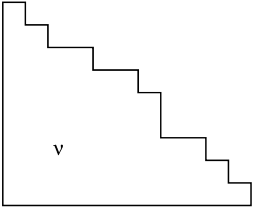
- μ or ν is a one-line rectangle (n) or (1^n)



(5) (1^3)

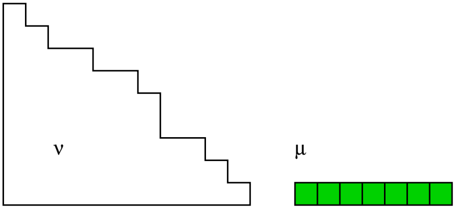
- We prove the validity of the Conjecture for both cases $((n), \nu)$ and $((1^n), \nu)$, where ν is any partition.
- The Conjecture holds for the pairs $(\mu, (1^n))$ and $(\mu, (n))$, where μ is any partition, if and only if it holds for $((n), \nu)$ and $((1^n), \nu)$, respectively.

Pieri's rule case: $\mu = (n)$



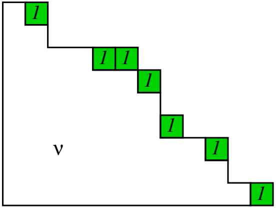
- Set $\nu = (11, 10, 9, 7, 7, 6, 4, 2, 1)$.
- Let $\mu = (7)$ be a one-part partition and ϑ be such that $c^{\vartheta}_{\nu,\mu} = 1$.
- The unique LR-filling of shape ϑ/ν and type μ is a horizontal strip.

Pieri's rule case: $\mu = (n)$



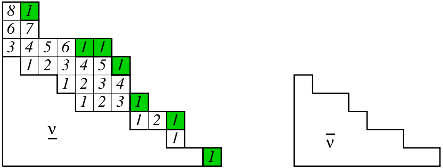
- Set $\nu = (11, 10, 9, 7, 7, 6, 4, 2, 1)$.
- Let $\mu = (7)$ be a one-part partition and ϑ be such that $c^{\vartheta}_{\nu,\mu} = 1$.
- The unique LR-filling of shape ϑ/ν and type μ is a horizontal strip.

Pieri's rule case: $\mu = (n)$



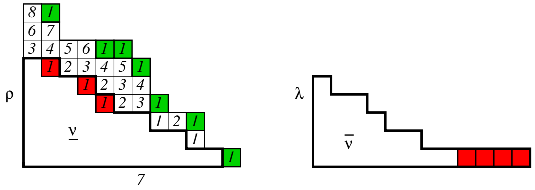
- Set $\nu = (11, 10, 9, 7, 7, 6, 4, 2, 1)$.
- Let $\mu = (7)$ be a one-part partition and ϑ be such that $c^{\vartheta}_{\nu,\mu} = 1$.
- The unique LR-filling of shape ϑ/ν and type μ is a horizontal strip.

Pieri's rule case: $\mu = (n)$



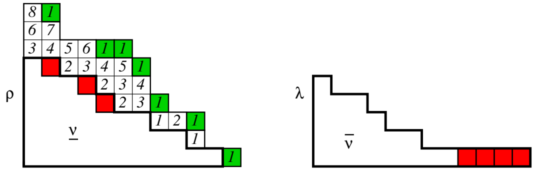
- Construct a tableau of shape $\vartheta/\underline{\nu}$, where the cells of ϑ/ν are filled with its unique LR-filling of type μ and those of $\nu/\underline{\nu}$ with the N-filling.
- The partition ρ is obtained making $\underline{\nu}$ grow in each corner up to column $n = 7$ and the type λ is equal to $\overline{\nu}$ plus a certain number of 1 entries, precisely $n - |\rho/\underline{\nu}|$.

Pieri's rule case: $\mu = (n)$



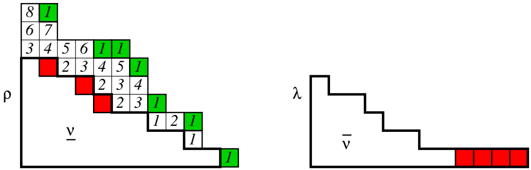
- Construct a tableau of shape $\vartheta/\underline{\nu}$, where the cells of ϑ/ν are filled with its unique LR-filling of type μ and those of $\nu/\underline{\nu}$ with the N-filling.
- The partition ρ is obtained making $\underline{\nu}$ grow in each corner up to column $n = 7$ and the type λ is equal to $\overline{\nu}$ plus a certain number of 1 entries, precisely $n - |\rho/\underline{\nu}|$.

Pieri's rule case: $\mu = (n)$



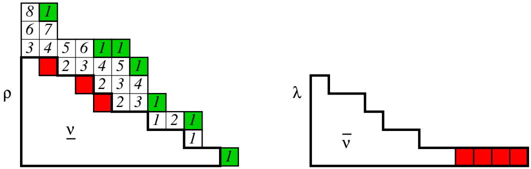
- Remove cells of $\rho/\underline{\nu}$.
- The tableau $T^{(0)}$ is set to be equal to T^{I}.
- Rearrange the entries of $T^{(0)}$ to obtain a semistandard tableau. In this case, the algorithm consists in constructing a sequence of hooks that insert the 1 entries of the horizontal strip in the tableau $T^{(0)}$.

Pieri's rule case: $\mu = (n)$



- Remove cells of $\rho/\underline{\nu}$.
- The tableau $T^{(0)}$ is set to be equal to T^{I}.
- Rearrange the entries of $T^{(0)}$ to obtain a semistandard tableau. In this case, the algorithm consists in constructing a sequence of hooks that insert the 1 entries of the horizontal strip in the tableau $T^{(0)}$.

Pieri's rule case: $\mu = (n)$

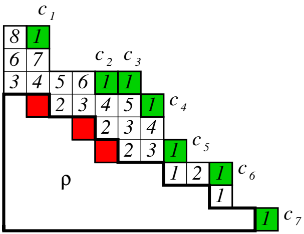


- Remove cells of $\rho/\underline{\nu}$.
- The tableau $T^{(0)}$ is set to be equal to T^{I}.
- Rearrange the entries of $T^{(0)}$ to obtain a semistandard tableau. In this case, the algorithm consists in constructing a sequence of hooks that insert the 1 entries of the horizontal strip in the tableau $T^{(0)}$.

Pieri's rule case: $\mu = (n)$

Algorithm

- Number cells of the horizontal strip left-to-right.
- Construct a sequence of hooks, where the vertex cell of each hook is determined by the previous one or by ρ, and reorder every hook.

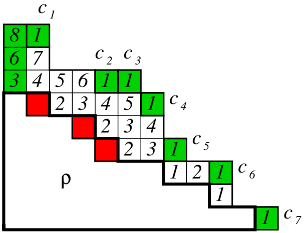


- After the rearrangement of the last hook, we obtain the final tableau T^F, which is semistandard and its reverse reading word is a lattice permutation.

Pieri's rule case: $\mu = (n)$

Algorithm

- Number cells of the horizontal strip left-to-right.
- Construct a sequence of hooks, where the vertex cell of each hook is determined by the previous one or by ρ, and reorder every hook.

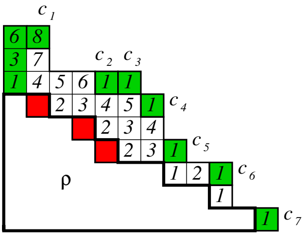


- After the rearrangement of the last hook, we obtain the final tableau T^F, which is semistandard and its reverse reading word is a lattice permutation.

Pieri's rule case: $\mu = (n)$

Algorithm

- Number cells of the horizontal strip left-to-right.
- Construct a sequence of hooks, where the vertex cell of each hook is determined by the previous one or by ρ, and reorder every hook.

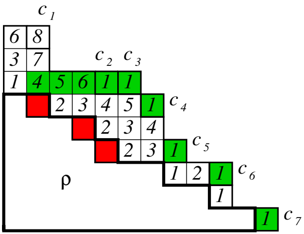


- After the rearrangement of the last hook, we obtain the final tableau T^F, which is semistandard and its reverse reading word is a lattice permutation.

Pieri's rule case: $\mu = (n)$

Algorithm

- Number cells of the horizontal strip left-to-right.
- Construct a sequence of hooks, where the vertex cell of each hook is determined by the previous one or by ρ, and reorder every hook.

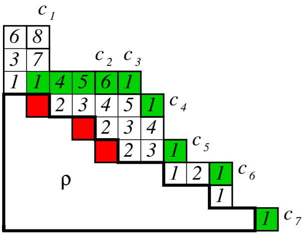


- After the rearrangement of the last hook, we obtain the final tableau T^F, which is semistandard and its reverse reading word is a lattice permutation.

Pieri's rule case: $\mu = (n)$

Algorithm

- Number cells of the horizontal strip left-to-right.
- Construct a sequence of hooks, where the vertex cell of each hook is determined by the previous one or by ρ, and reorder every hook.

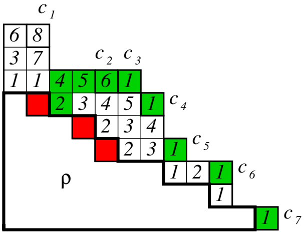


- After the rearrangement of the last hook, we obtain the final tableau T^F, which is semistandard and its reverse reading word is a lattice permutation.

Pieri's rule case: $\mu = (n)$

Algorithm

- Number cells of the horizontal strip left-to-right.
- Construct a sequence of hooks, where the vertex cell of each hook is determined by the previous one or by ρ, and reorder every hook.

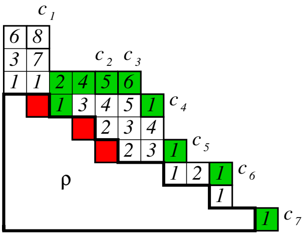


- After the rearrangement of the last hook, we obtain the final tableau T^F, which is semistandard and its reverse reading word is a lattice permutation.

Pieri's rule case: $\mu = (n)$

Algorithm

- Number cells of the horizontal strip left-to-right.
- Construct a sequence of hooks, where the vertex cell of each hook is determined by the previous one or by ρ, and reorder every hook.

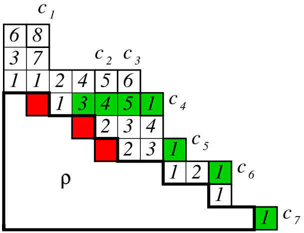


- After the rearrangement of the last hook, we obtain the final tableau T^F, which is semistandard and its reverse reading word is a lattice permutation.

Pieri's rule case: $\mu = (n)$

Algorithm

- Number cells of the horizontal strip left-to-right.
- Construct a sequence of hooks, where the vertex cell of each hook is determined by the previous one or by ρ, and reorder every hook.

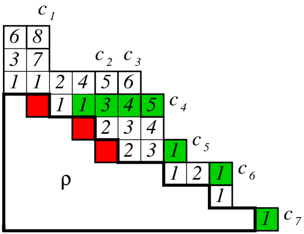


- After the rearrangement of the last hook, we obtain the final tableau T^F, which is semistandard and its reverse reading word is a lattice permutation.

Pieri's rule case: $\mu = (n)$

Algorithm

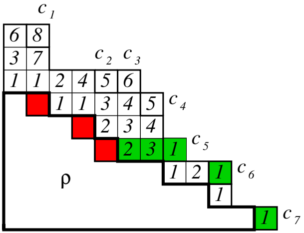
- Number cells of the horizontal strip left-to-right.
- Construct a sequence of hooks, where the vertex cell of each hook is determined by the previous one or by ρ, and reorder every hook.



- After the rearrangement of the last hook, we obtain the final tableau T^F, which is semistandard and its reverse reading word is a lattice permutation.

Pieri's rule case: $\mu = (n)$
Algorithm

- Number cells of the horizontal strip left-to-right.
- Construct a sequence of hooks, where the vertex cell of each hook is determined by the previous one or by ρ, and reorder every hook.

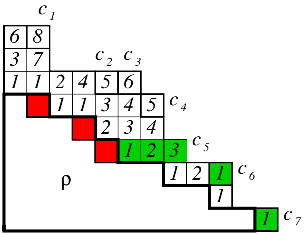


- After the rearrangement of the last hook, we obtain the final tableau T^F, which is semistandard and its reverse reading word is a lattice permutation.

Pieri's rule case: $\mu = (n)$

Algorithm

- Number cells of the horizontal strip left-to-right.
- Construct a sequence of hooks, where the vertex cell of each hook is determined by the previous one or by ρ, and reorder every hook.

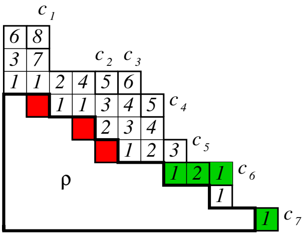


- After the rearrangement of the last hook, we obtain the final tableau T^F, which is semistandard and its reverse reading word is a lattice permutation.

Pieri's rule case: $\mu = (n)$

Algorithm

- Number cells of the horizontal strip left-to-right.
- Construct a sequence of hooks, where the vertex cell of each hook is determined by the previous one or by ρ, and reorder every hook.

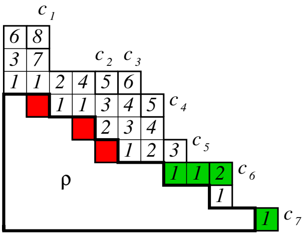


- After the rearrangement of the last hook, we obtain the final tableau T^F, which is semistandard and its reverse reading word is a lattice permutation.

Pieri's rule case: $\mu = (n)$

Algorithm

- Number cells of the horizontal strip left-to-right.
- Construct a sequence of hooks, where the vertex cell of each hook is determined by the previous one or by ρ, and reorder every hook.

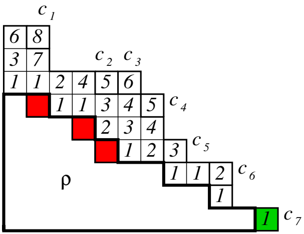


- After the rearrangement of the last hook, we obtain the final tableau T^F, which is semistandard and its reverse reading word is a lattice permutation.

Pieri's rule case: $\mu = (n)$

Algorithm

- Number cells of the horizontal strip left-to-right.
- Construct a sequence of hooks, where the vertex cell of each hook is determined by the previous one or by ρ, and reorder every hook.

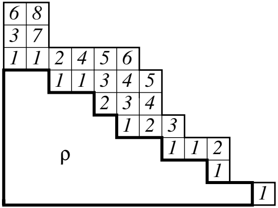


- After the rearrangement of the last hook, we obtain the final tableau T^F, which is semistandard and its reverse reading word is a lattice permutation.

Pieri's rule case: $\mu = (n)$

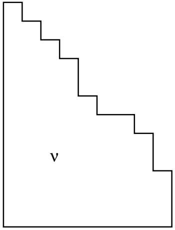
Algorithm

- Number cells of the horizontal strip left-to-right.
- Construct a sequence of hooks, where the vertex cell of each hook is determined by the previous one or by ρ, and reorder every hook.



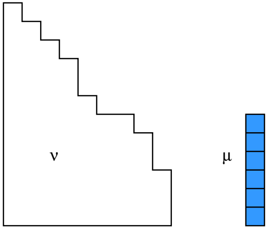
- After the rearrangement of the last hook, we obtain the final tableau T^F, which is semistandard and its reverse reading word is a lattice permutation.

Pieri's rule case: $\mu = (1^n)$



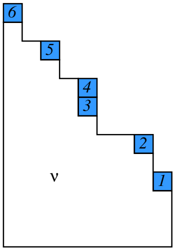
- Set $\nu = (9, 9, 9, 8, 8, 7, 5, 4, 4, 3, 2, 1)$.
- Let $\mu = (1^6)$ and ϑ be such that $c^{\vartheta}_{\nu,\mu} = 1$.
- The unique LR-filling of shape ϑ/ν and type μ is a vertical strip.

Pieri's rule case: $\mu = (1^n)$



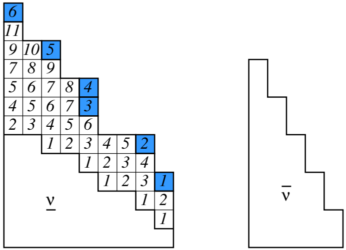
- Set $\nu = (9, 9, 9, 8, 8, 7, 5, 4, 4, 3, 2, 1)$.
- Let $\mu = (1^6)$ and ϑ be such that $c^{\vartheta}_{\nu,\mu} = 1$.
- The unique LR-filling of shape ϑ/ν and type μ is a vertical strip.

Pieri's rule case: $\mu = (1^n)$



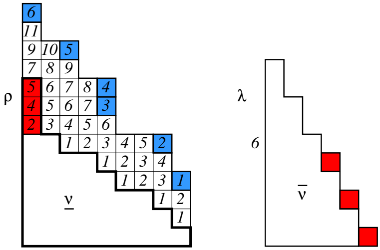
- Set $\nu = (9, 9, 9, 8, 8, 7, 5, 4, 4, 3, 2, 1)$.
- Let $\mu = (1^6)$ and ϑ be such that $c^{\vartheta}_{\nu,\mu} = 1$.
- The unique LR-filling of shape ϑ/ν and type μ is a vertical strip.

Pieri's rule case: $\mu = (1^n)$



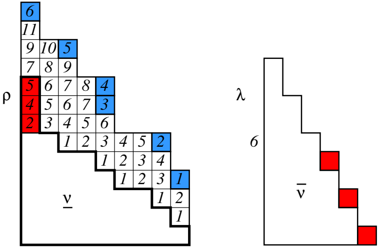
- Construct a tableau of shape $\vartheta/\underline{\nu}$, where the cells of ϑ/ν are filled with its unique LR-filling of type μ and those of $\nu/\underline{\nu}$ with the N-filling.
- The partition λ is obtained making $\overline{\nu}$ grow in each corner up to row $n = 6$, while ρ is $\underline{\nu}$ plus a final sequence of parts equal to 1 of length $n - |\lambda/\overline{\nu}|$.

Pieri's rule case: $\mu = (1^n)$



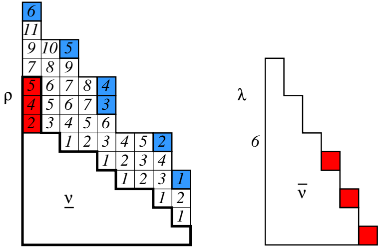
- Construct a tableau of shape $\vartheta/\underline{\nu}$, where the cells of ϑ/ν are filled with its unique LR-filling of type μ and those of $\nu/\underline{\nu}$ with the N-filling.
- The partition λ is obtained making $\overline{\nu}$ grow in each corner up to row $n = 6$, while ρ is $\underline{\nu}$ plus a final sequence of parts equal to 1 of length $n - |\lambda/\overline{\nu}|$.

Pieri's rule case: $\mu = (1^n)$



- The tableau $T^{(0)}$ is set to be equal to T^{I}.
- Rearrange the entries of $T^{(0)}$ to obtain a semistandard tableau. Also this algorithm consists in constructing a sequence of hooks, one per each cell of the vertical strip, but it is different from the previous one.

Pieri's rule case: $\mu = (1^n)$

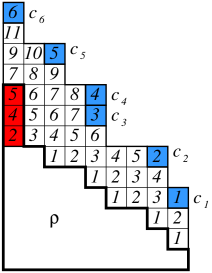


- The tableau $T^{(0)}$ is set to be equal to T^I.
- Rearrange the entries of $T^{(0)}$ to obtain a semistandard tableau. Also this algorithm consists in constructing a sequence of hooks, one per each cell of the vertical strip, but it is different from the previous one.

Pieri's rule case: $\mu = (1^n)$

Algorithm

- Let $c_1, \ldots, c_n$ be the cells of the vertical strip numbered bottom-to-top.
- Construct a sequence of hooks and reorder every hook according to the entry of the corresponding cell c_k.

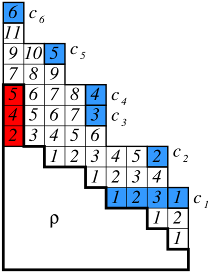


- After the rearrangement of the last hook, we obtain the final tableau T^F.

Pieri's rule case: $\mu = (1^n)$

Algorithm

- Let $c_1, \ldots, c_n$ be the cells of the vertical strip numbered bottom-to-top.
- Construct a sequence of hooks and reorder every hook according to the entry of the corresponding cell c_k.

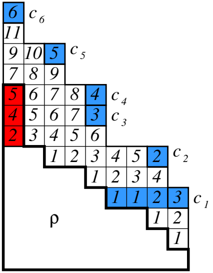


- After the rearrangement of the last hook, we obtain the final tableau T^F.

Pieri's rule case: $\mu = (1^n)$

Algorithm

- Let $c_1, \ldots, c_n$ be the cells of the vertical strip numbered bottom-to-top.
- Construct a sequence of hooks and reorder every hook according to the entry of the corresponding cell c_k.

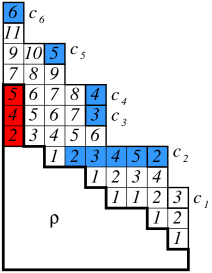


- After the rearrangement of the last hook, we obtain the final tableau T^F.

Pieri's rule case: $\mu = (1^n)$

Algorithm

- Let $c_1, \ldots, c_n$ be the cells of the vertical strip numbered bottom-to-top.
- Construct a sequence of hooks and reorder every hook according to the entry of the corresponding cell c_k.

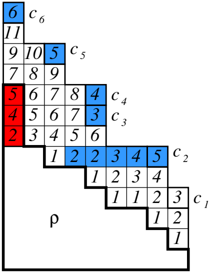


- After the rearrangement of the last hook, we obtain the final tableau T^F.

Pieri's rule case: $\mu = (1^n)$

Algorithm

- Let $c_1, \ldots, c_n$ be the cells of the vertical strip numbered bottom-to-top.
- Construct a sequence of hooks and reorder every hook according to the entry of the corresponding cell c_k.

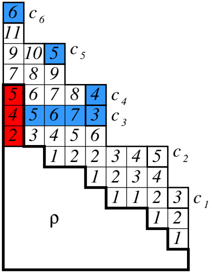


- After the rearrangement of the last hook, we obtain the final tableau T^F.

Pieri's rule case: $\mu = (1^n)$

Algorithm

- Let $c_1, \ldots, c_n$ be the cells of the vertical strip numbered bottom-to-top.
- Construct a sequence of hooks and reorder every hook according to the entry of the corresponding cell c_k.

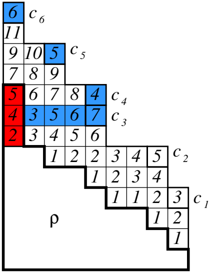


- After the rearrangement of the last hook, we obtain the final tableau T^F.

Pieri's rule case: $\mu = (1^n)$

Algorithm

- Let $c_1, \ldots, c_n$ be the cells of the vertical strip numbered bottom-to-top.
- Construct a sequence of hooks and reorder every hook according to the entry of the corresponding cell c_k.

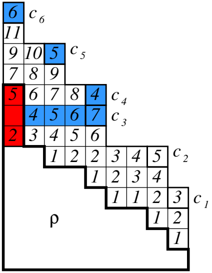


- After the rearrangement of the last hook, we obtain the final tableau T^F.

Pieri's rule case: $\mu = (1^n)$

Algorithm

- Let $c_1, \ldots, c_n$ be the cells of the vertical strip numbered bottom-to-top.
- Construct a sequence of hooks and reorder every hook according to the entry of the corresponding cell c_k.

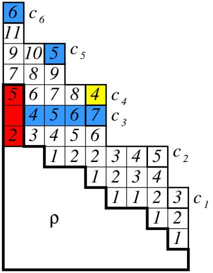


- After the rearrangement of the last hook, we obtain the final tableau T^F.

Pieri's rule case: $\mu = (1^n)$

Algorithm

- Let $c_1, \ldots, c_n$ be the cells of the vertical strip numbered bottom-to-top.
- Construct a sequence of hooks and reorder every hook according to the entry of the corresponding cell c_k.

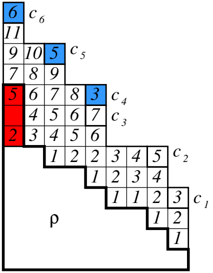


- After the rearrangement of the last hook, we obtain the final tableau T^F.

Pieri's rule case: $\mu = (1^n)$

Algorithm

- Let $c_1, \ldots, c_n$ be the cells of the vertical strip numbered bottom-to-top.
- Construct a sequence of hooks and reorder every hook according to the entry of the corresponding cell c_k.

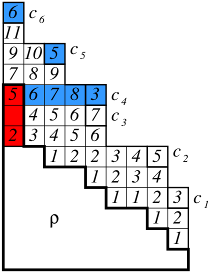


- After the rearrangement of the last hook, we obtain the final tableau T^F.

Pieri's rule case: $\mu = (1^n)$

Algorithm

- Let $c_1, \ldots, c_n$ be the cells of the vertical strip numbered bottom-to-top.
- Construct a sequence of hooks and reorder every hook according to the entry of the corresponding cell c_k.

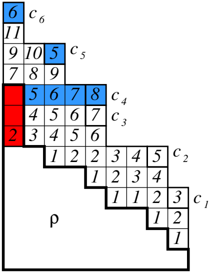


- After the rearrangement of the last hook, we obtain the final tableau T^F.

Pieri's rule case: $\mu = (1^n)$

Algorithm

- Let $c_1, \dots, c_n$ be the cells of the vertical strip numbered bottom-to-top.
- Construct a sequence of hooks and reorder every hook according to the entry of the corresponding cell c_k.

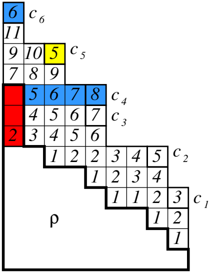


- After the rearrangement of the last hook, we obtain the final tableau T^F.

Pieri's rule case: $\mu = (1^n)$

Algorithm

- Let $c_1, \ldots, c_n$ be the cells of the vertical strip numbered bottom-to-top.
- Construct a sequence of hooks and reorder every hook according to the entry of the corresponding cell c_k.

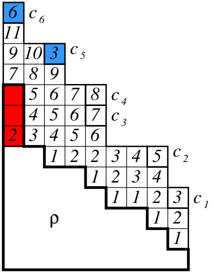


- After the rearrangement of the last hook, we obtain the final tableau T^F.

Pieri's rule case: $\mu = (1^n)$

Algorithm

- Let $c_1, \dots, c_n$ be the cells of the vertical strip numbered bottom-to-top.
- Construct a sequence of hooks and reorder every hook according to the entry of the corresponding cell c_k.

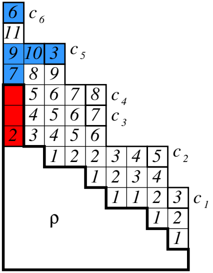


- After the rearrangement of the last hook, we obtain the final tableau T^F.

Pieri's rule case: $\mu = (1^n)$

Algorithm

- Let $c_1, \ldots, c_n$ be the cells of the vertical strip numbered bottom-to-top.
- Construct a sequence of hooks and reorder every hook according to the entry of the corresponding cell c_k.

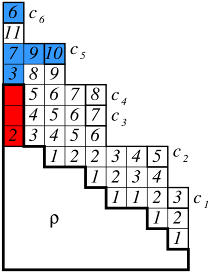


- After the rearrangement of the last hook, we obtain the final tableau T^F.

Pieri's rule case: $\mu = (1^n)$

Algorithm

- Let $c_1, \dots, c_n$ be the cells of the vertical strip numbered bottom-to-top.
- Construct a sequence of hooks and reorder every hook according to the entry of the corresponding cell c_k.

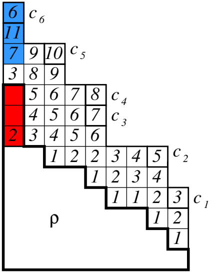


- After the rearrangement of the last hook, we obtain the final tableau T^F.

Pieri's rule case: $\mu = (1^n)$

Algorithm

- Let $c_1, \ldots, c_n$ be the cells of the vertical strip numbered bottom-to-top.
- Construct a sequence of hooks and reorder every hook according to the entry of the corresponding cell c_k.

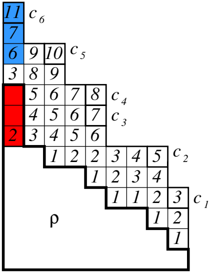


- After the rearrangement of the last hook, we obtain the final tableau T^F.

Pieri's rule case: $\mu = (1^n)$

Algorithm

- Let $c_1, \ldots, c_n$ be the cells of the vertical strip numbered bottom-to-top.
- Construct a sequence of hooks and reorder every hook according to the entry of the corresponding cell c_k.

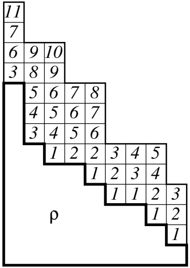


- After the rearrangement of the last hook, we obtain the final tableau T^F.

Pieri's rule case: $\mu = (1^n)$

Algorithm

- Let $c_1, \ldots, c_n$ be the cells of the vertical strip numbered bottom-to-top.
- Construct a sequence of hooks and reorder every hook according to the entry of the corresponding cell c_k.



- After the rearrangement of the last hook, we obtain the final tableau T^F.

Stembridge's characterization

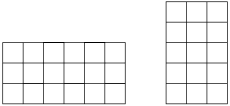
Theorem [Stembridge]

The product $s_\mu s_\nu$ is multiplicity-free if and only if

- *μ or ν is a one-line rectangle (Pieri's rule),or*
- *μ and ν are rectangles, or*
- *μ is a two-line rectangle and ν a fat hook or vice-versa, or*
- *μ is a rectangle and ν is a near-rectangle or vice-versa.*

Rectangles case

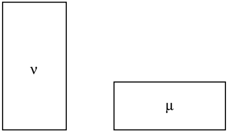
- $\mu = (a^c)$ and $\nu = (b^d)$ are rectangles



(6^3) (3^5)

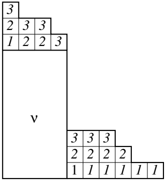
- We prove the validity of the Conjecture for the pair (μ, ν) only in case $c \leq d$ or $c > d$ and $a < b$.
- The validity in case $c > d$ and $a \geq b$ follows from the property that the Conjecture holds for the pair (μ, ν) iff it holds for (ν', μ').

Rectangles case: $c \leq d$



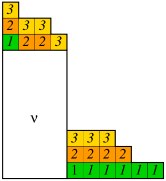
- Set $\nu = (b^d) = (4^8)$.
- Let $\mu = (a^c) = (7^3)$ be a rectangle and ϑ be such that $c^{\vartheta}_{\nu,\mu} = 1$.
- The unique LR-filling of shape ϑ/ν and type μ can be divided into three horizontal strips.

Rectangles case: $c \leq d$



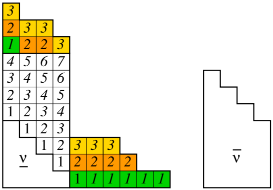
- Set $\nu = (b^d) = (4^8)$.
- Let $\mu = (a^c) = (7^3)$ be a rectangle and ϑ be such that $c^{\vartheta}_{\nu,\mu} = 1$.
- The unique LR-filling of shape ϑ/ν and type μ can be divided into three horizontal strips.

Rectangles case: $c \leq d$



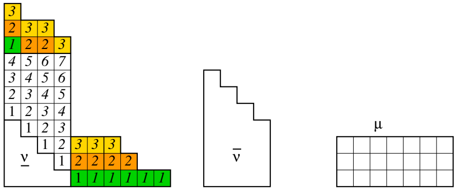
- Set $\nu = (b^d) = (4^8)$.
- Let $\mu = (a^c) = (7^3)$ be a rectangle and ϑ be such that $c^{\vartheta}_{\nu,\mu} = 1$.
- The unique LR-filling of shape ϑ/ν and type μ can be divided into three horizontal strips.

Rectangles case: $c \leq d$



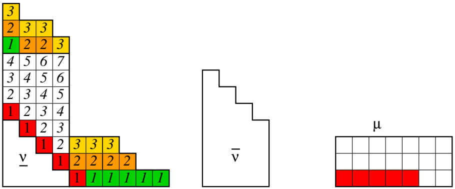
- Construct a tableau of shape $\vartheta/\underline{\nu}$, where the cells of ϑ/ν are filled with its unique LR-filling of type μ and those of $\nu/\underline{\nu}$ with the N-filling.
- The pair $(\lambda, \rho) = (\mu, \nu)^*$ is obtained from $(\overline{\nu}, \underline{\nu})$ by means of the recursive definition adding the cells of μ row-by-row either to $\underline{\nu}$ or $\overline{\nu}$.

Rectangles case: $c \leq d$



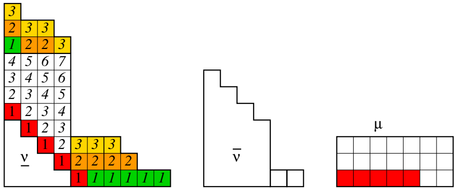
- Construct a tableau of shape $\vartheta/\underline{\nu}$, where the cells of ϑ/ν are filled with its unique LR-filling of type μ and those of $\nu/\underline{\nu}$ with the N-filling.
- The pair $(\lambda, \rho) = (\mu, \nu)^*$ is obtained from $(\overline{\nu}, \underline{\nu})$ by means of the recursive definition adding the cells of μ row-by-row either to $\underline{\nu}$ or $\overline{\nu}$.

Rectangles case: $c \leq d$



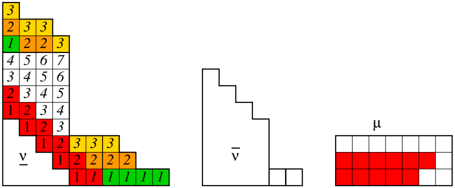
- Construct a tableau of shape $\vartheta/\underline{\nu}$, where the cells of ϑ/ν are filled with its unique LR-filling of type μ and those of $\nu/\underline{\nu}$ with the N-filling.
- The pair $(\lambda, \rho) = (\mu, \nu)^*$ is obtained from $(\overline{\nu}, \underline{\nu})$ by means of the recursive definition adding the cells of μ row-by-row either to $\underline{\nu}$ or $\overline{\nu}$.

Rectangles case: $c \leq d$



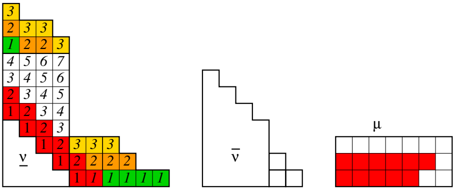
- Construct a tableau of shape $\vartheta/\underline{\nu}$, where the cells of ϑ/ν are filled with its unique LR-filling of type μ and those of $\nu/\underline{\nu}$ with the N-filling.
- The pair $(\lambda, \rho) = (\mu, \nu)^*$ is obtained from $(\overline{\nu}, \underline{\nu})$ by means of the recursive definition adding the cells of μ row-by-row either to $\underline{\nu}$ or $\overline{\nu}$.

Rectangles case: $c \leq d$



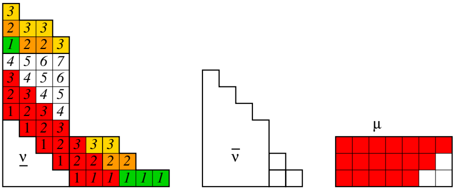
- Construct a tableau of shape $\vartheta/\underline{\nu}$, where the cells of ϑ/ν are filled with its unique LR-filling of type μ and those of $\nu/\underline{\nu}$ with the N-filling.
- The pair $(\lambda, \rho) = (\mu, \nu)^*$ is obtained from $(\overline{\nu}, \underline{\nu})$ by means of the recursive definition adding the cells of μ row-by-row either to $\underline{\nu}$ or $\overline{\nu}$.

Rectangles case: $c \leq d$



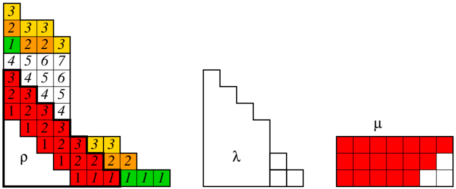
- Construct a tableau of shape $\vartheta/\underline{\nu}$, where the cells of ϑ/ν are filled with its unique LR-filling of type μ and those of $\nu/\underline{\nu}$ with the N-filling.
- The pair $(\lambda, \rho) = (\mu, \nu)^*$ is obtained from $(\overline{\nu}, \underline{\nu})$ by means of the recursive definition adding the cells of μ row-by-row either to $\underline{\nu}$ or $\overline{\nu}$.

Rectangles case: $c \leq d$



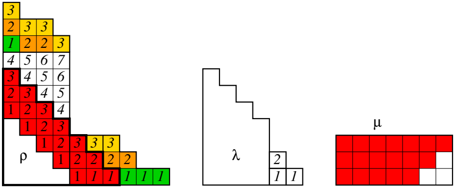
- Construct a tableau of shape $\vartheta/\underline{\nu}$, where the cells of ϑ/ν are filled with its unique LR-filling of type μ and those of $\nu/\underline{\nu}$ with the N-filling.
- The pair $(\lambda, \rho) = (\mu, \nu)^*$ is obtained from $(\overline{\nu}, \underline{\nu})$ by means of the recursive definition adding the cells of μ row-by-row either to $\underline{\nu}$ or $\overline{\nu}$.

Rectangles case: $c \leq d$



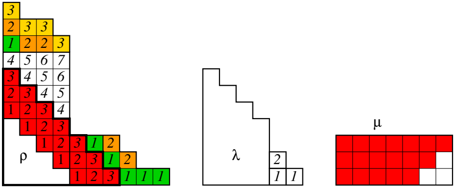
- Construct a tableau of shape $\vartheta/\underline{\nu}$, where the cells of ϑ/ν are filled with its unique LR-filling of type μ and those of $\nu/\underline{\nu}$ with the N-filling.
- The pair $(\lambda, \rho) = (\mu, \nu)^*$ is obtained from $(\overline{\nu}, \underline{\nu})$ by means of the recursive definition adding the cells of μ row-by-row either to $\underline{\nu}$ or $\overline{\nu}$.

Rectangles case: $c \leq d$



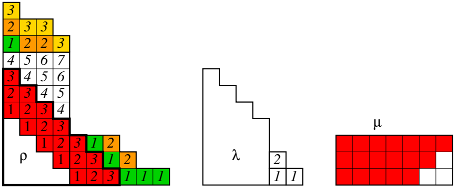
- The tableau T^I has not type λ.
- The tableau $T^{(0)}$ is obtained modifying the entries of T^I.
- Rearrange the entries of $T^{(0)}$ to obtain a semistandard tableau. Basically, in this case, the algorithm consists in repeating three times - one per each horizontal strip - the algorithm given for a one-line rectangle.

Rectangles case: $c \leq d$



- The tableau T^I has not type λ.
- The tableau $T^{(0)}$ is obtained modifying the entries of T^I.
- Rearrange the entries of $T^{(0)}$ to obtain a semistandard tableau. Basically, in this case, the algorithm consists in repeating three times - one per each horizontal strip - the algorithm given for a one-line rectangle.

Rectangles case: $c \leq d$

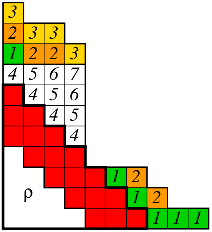


- The tableau T^I has not type λ.
- The tableau $T^{(0)}$ is obtained modifying the entries of T^I.
- Rearrange the entries of $T^{(0)}$ to obtain a semistandard tableau. Basically, in this case, the algorithm consists in repeating three times - one per each horizontal strip - the algorithm given for a one-line rectangle.

Rectangles case: $c \leq d$

Algorithm

- Firstly, consider the horizontal strip with 1 entries, then the one with 2 entries and, finally, the last one with 3 entries.
- Each time we consider a horizontal strip with i entries, we construct and reorder vertical hooks that do not involve cells filled with $i - 1$.

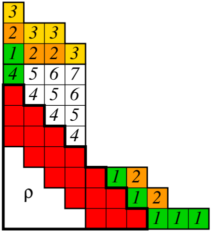


- After the rearrangement of the last hooks, we obtain the final tableau T^F.

Rectangles case: $c \leq d$

Algorithm

- Firstly, consider the horizontal strip with 1 entries, then the one with 2 entries and, finally, the last one with 3 entries.
- Each time we consider a horizontal strip with i entries, we construct and reorder vertical hooks that do not involve cells filled with $i-1$.

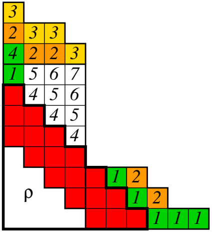


- After the rearrangement of the last hooks, we obtain the final tableau T^F.

Rectangles case: $c \leq d$

Algorithm

- Firstly, consider the horizontal strip with 1 entries, then the one with 2 entries and, finally, the last one with 3 entries.
- Each time we consider a horizontal strip with i entries, we construct and reorder vertical hooks that do not involve cells filled with $i - 1$.

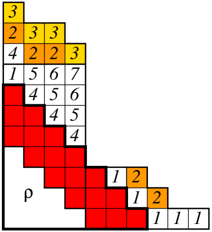


- After the rearrangement of the last hooks, we obtain the final tableau T^F.

Rectangles case: $c \leq d$

Algorithm

- Firstly, consider the horizontal strip with 1 entries, then the one with 2 entries and, finally, the last one with 3 entries.
- Each time we consider a horizontal strip with i entries, we construct and reorder vertical hooks that do not involve cells filled with $i - 1$.

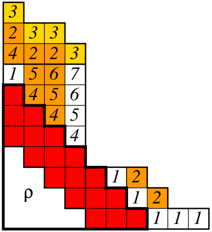


- After the rearrangement of the last hooks, we obtain the final tableau T^F.

Rectangles case: $c \leq d$

Algorithm

- Firstly, consider the horizontal strip with 1 entries, then the one with 2 entries and, finally, the last one with 3 entries.
- Each time we consider a horizontal strip with i entries, we construct and reorder vertical hooks that do not involve cells filled with $i - 1$.

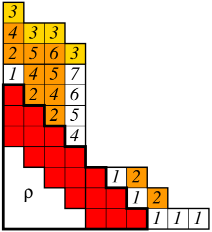


- After the rearrangement of the last hooks, we obtain the final tableau T^F.

Rectangles case: $c \leq d$

Algorithm

- Firstly, consider the horizontal strip with 1 entries, then the one with 2 entries and, finally, the last one with 3 entries.
- Each time we consider a horizontal strip with i entries, we construct and reorder vertical hooks that do not involve cells filled with $i - 1$.

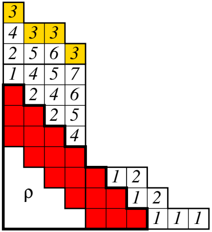


- After the rearrangement of the last hooks, we obtain the final tableau T^F.

Rectangles case: $c \leq d$

Algorithm

- Firstly, consider the horizontal strip with 1 entries, then the one with 2 entries and, finally, the last one with 3 entries.
- Each time we consider a horizontal strip with i entries, we construct and reorder vertical hooks that do not involve cells filled with $i - 1$.

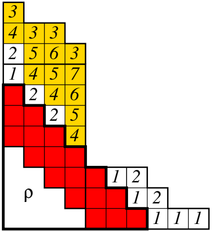


- After the rearrangement of the last hooks, we obtain the final tableau T^F.

Rectangles case: $c \leq d$

Algorithm

- Firstly, consider the horizontal strip with 1 entries, then the one with 2 entries and, finally, the last one with 3 entries.
- Each time we consider a horizontal strip with i entries, we construct and reorder vertical hooks that do not involve cells filled with $i - 1$.

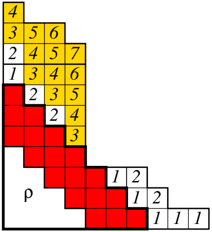


- After the rearrangement of the last hooks, we obtain the final tableau T^F.

Rectangles case: $c \leq d$

Algorithm

- Firstly, consider the horizontal strip with 1 entries, then the one with 2 entries and, finally, the last one with 3 entries.
- Each time we consider a horizontal strip with i entries, we construct and reorder vertical hooks that do not involve cells filled with $i - 1$.

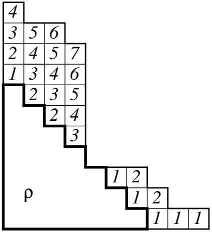


- After the rearrangement of the last hooks, we obtain the final tableau T^F.

Rectangles case: $c \leq d$

Algorithm

- Firstly, consider the horizontal strip with 1 entries, then the one with 2 entries and, finally, the last one with 3 entries.
- Each time we consider a horizontal strip with i entries, we construct and reorder vertical hooks that do not involve cells filled with $i - 1$.



- After the rearrangement of the last hooks, we obtain the final tableau T^F.

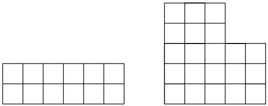
Stembridge's characterization

Theorem [Stembridge]

The product $s_\mu s_\nu$ is multiplicity-free if and only if

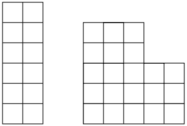
- μ *or* ν *is a one-line rectangle (Pieri's rule),or*
- μ *and* ν *are rectangles, or*
- μ *is a two-line rectangle and* ν *a fat hook or vice-versa, or*
- μ *is a rectangle and* ν *is a near-rectangle or vice-versa.*

Stembridge's case (iii)



- For the case μ two-line rectangle (horizontal or vertical) and ν fat hook the underlying idea is basically similar to the one of the Pieri's rule case. Nevertheless, this case is more complex and it splits in several sub-cases, which make the algorithms for solving this case substantially different from the previous ones.

Stembridge's case (iii)



- For the case μ two-line rectangle (horizontal or vertical) and ν fat hook the underlying idea is basically similar to the one of the Pieri's rule case. Nevertheless, this case is more complex and it splits in several sub-cases, which make the algorithms for solving this case substantially different from the previous ones.

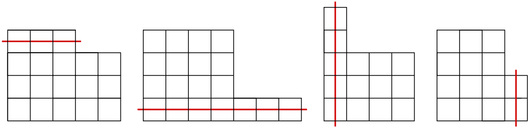
Stembridge's characterization

Theorem [Stembridge]

The product $s_\mu s_\nu$ is multiplicity-free if and only if

- μ *or* ν *is a one-line rectangle (Pieri's rule),or*
- μ *and* ν *are rectangles, or*
- μ *is a two-line rectangle and* ν *a fat hook or vice-versa, or*
- μ *is a rectangle and* ν *is a near-rectangle or vice-versa.*

Stembridge's case (iv) open



$(5^3,3)$ $(7,4^3)$ $(4^3,1^2)$ $(4^2,3^2)$

- The last case, μ rectangle and ν near rectangle (or vice-versa), is still open. It splits in more and more sub-cases, as well. We know how to treat some of these sub-cases simply applying algorithms similar to the one provided for rectangles.