

Building Mathematical Bridges Between Symmetric Functions

George H. Seelinger

Jefferson Scholars Foundation

ghs9ae@virginia.edu

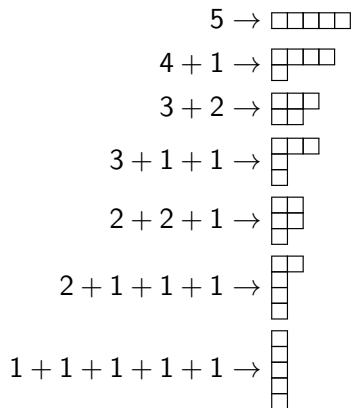
28 November 2018

Partitions of 5

How many ways can we write a positive integer as a sum of positive integers?

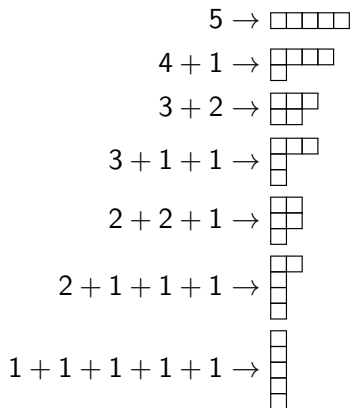
Partitions of 5

How many ways can we write a positive integer as a sum of positive integers?



Partitions of 5

How many ways can we write a positive integer as a sum of positive integers?



We will use these diagrams to describe a type of symmetric function called a “Schur function.”

Raising Operators

To do this, we will need functions that change partition diagrams called “raising operators.”

Raising Operators

To do this, we will need functions that change partition diagrams called “raising operators.”

We can change partition diagrams by moving boxes.

$$R_{1,3} \left(\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & & \\ \hline \square & & \\ \hline \end{array} \right) = \begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline & & & \\ \hline & & & \\ \hline \end{array}$$

$$R_{2,3} \left(\begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array} \right) = \begin{array}{|c|c|} \hline \square & \square \\ \hline & \square \\ \hline \end{array}$$

Raising Operators

To do this, we will need functions that change partition diagrams called “raising operators.”

We can change partition diagrams by moving boxes.

$$R_{1,3} \left(\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & & \\ \hline \square & & \\ \hline \end{array} \right) = \begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline & & & \\ \hline & & & \\ \hline \end{array}$$

$$R_{2,3} \left(\begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array} \right) = \begin{array}{|c|c|} \hline \square & \square \\ \hline & \square \\ \hline & \\ \hline \end{array}$$

If the result “does not make sense”, we get 0:

$$R_{1,4} \left(\begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array} \right) = 0$$

Schur functions

We define a new class of functions. Given a partition diagram λ with ℓ rows, we have definition

Schur functions

We define a new class of functions. Given a partition diagram λ with ℓ rows, we have definition

Definition

$$s_\lambda = (1 - R_{1,2}) \\ (1 - R_{1,3})(1 - R_{2,3}) \\ \dots \\ (1 - R_{1,\ell})(1 - R_{2,\ell}) \cdots (1 - R_{\ell-2,\ell})(1 - R_{\ell-1,\ell})\lambda$$

Schur functions

We define a new class of functions. Given a partition diagram λ with ℓ rows, we have definition

Definition

$$s_\lambda = (1 - R_{1,2}) \\ (1 - R_{1,3})(1 - R_{2,3}) \\ \dots \\ (1 - R_{1,\ell})(1 - R_{2,\ell}) \cdots (1 - R_{\ell-2,\ell})(1 - R_{\ell-1,\ell})\lambda$$

Example

$$s_{\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & & \\ \hline \square & & \\ \hline \end{array}} = (1 - R_{1,2})(1 - R_{1,3})(1 - R_{2,3}) \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & & \\ \hline \square & & \\ \hline \end{array}$$

Example

$$s_{\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & & \\ \hline \square & & \\ \hline \end{array}} = (1 - R_{1,2})(1 - R_{1,3})(1 - R_{2,3}) s_{\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \\ \hline \square & & \\ \hline \end{array}}$$

Example

$$s_{\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \\ \hline \square & & \\ \hline \end{array}} = (1 - R_{1,2})(1 - R_{1,3})(1 - R_{2,3}) s_{\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array}}$$

Recall the foil method from high school:

$$\begin{aligned} & (1 - R_{1,2})(1 - R_{1,3})(1 - R_{2,3}) \\ &= (1 - R_{1,2} - R_{1,3} + R_{1,2}R_{1,3})(1 - R_{2,3}) \\ &= 1 - R_{1,2} - R_{1,3} - R_{2,3} + R_{1,2}R_{1,3} + R_{1,2}R_{2,3} + R_{1,3}R_{2,3} - R_{1,2}R_{1,3}R_{2,3} \end{aligned}$$

Example

$$s_{\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & & \\ \hline \square & & \\ \hline \end{array}} = (1 - R_{1,2})(1 - R_{1,3})(1 - R_{2,3}) s_{\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \square & & \\ \hline \end{array}}$$

Recall the foil method from high school:

$$\begin{aligned} & (1 - R_{1,2})(1 - R_{1,3})(1 - R_{2,3}) \\ &= (1 - R_{1,2} - R_{1,3} + R_{1,2}R_{1,3})(1 - R_{2,3}) \\ &= 1 - R_{1,2} - R_{1,3} - R_{2,3} + R_{1,2}R_{1,3} + R_{1,2}R_{2,3} + R_{1,3}R_{2,3} - R_{1,2}R_{1,3}R_{2,3} \end{aligned}$$

So, we must compute $s_{\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & & \\ \hline \square & & \\ \hline \end{array}} =$

$$(1 - R_{1,2} - R_{1,3} - R_{2,3} + R_{1,2}R_{1,3} + R_{1,2}R_{2,3} + R_{1,3}R_{2,3} - R_{1,2}R_{1,3}R_{2,3}) s_{\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \square & & \\ \hline \end{array}}$$

Example

$$s_{\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & & \\ \hline \square & & \\ \hline \square & & \\ \hline \end{array}} = (1 - R_{1,2})(1 - R_{1,3})(1 - R_{2,3})\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & & \\ \hline \square & & \\ \hline \square & & \\ \hline \end{array}$$

Example

$$s_{\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}} = (1 - R_{1,2})(1 - R_{1,3})(1 - R_{2,3})\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}$$

$$\begin{array}{ccc} -R_{1,2}(\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}) & -R_{1,3}(\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}) & -R_{2,3}(\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}) \\ +R_{1,2}R_{1,3}(\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}) & +R_{1,2}R_{2,3}(\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}) & +R_{1,3}R_{2,3}(\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}) \\ -R_{1,2}R_{1,3}R_{2,3}(\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}) & & \end{array}$$

Example

$$s_{\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}} = (1 - R_{1,2})(1 - R_{1,3})(1 - R_{2,3})\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}$$

$$\begin{array}{r}
 -R_{1,2}(\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}) \\
 +R_{1,2}R_{1,3}(\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}) \\
 \end{array}
 \begin{array}{r}
 \begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix} \\
 -R_{1,3}(\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}) \\
 +R_{1,2}R_{2,3}(\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}) \\
 -R_{1,2}R_{1,3}R_{2,3}(\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}) \\
 \end{array}
 \begin{array}{r}
 -R_{2,3}(\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}) \\
 +R_{1,3}R_{2,3}(\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}) \\
 \end{array}
 =
 \begin{array}{r}
 -\begin{smallmatrix} \square & \square & \square \\ \square & \square & \square \end{smallmatrix} \\
 +\begin{smallmatrix} \square & \square & \square & \square \\ \square & \square & \square & \square \end{smallmatrix} \\
 \end{array}
 \begin{array}{r}
 \begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix} \\
 -\begin{smallmatrix} \square & \square & \square \\ \square & \square & \square \end{smallmatrix} \\
 +\begin{smallmatrix} \square & \square & \square \\ \square & \square & \square \end{smallmatrix} \\
 -0 \\
 \end{array}
 \begin{array}{r}
 -\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix} \\
 +0 \\
 \end{array}$$

Example continued

Example

$$s_{\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array}} = (1 - R_{1,2})(1 - R_{1,3})(1 - R_{2,3})\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array}$$

$$\begin{array}{l} -R_{1,2}(\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array}) \\ +R_{1,2}R_{1,3}(\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array}) \end{array} \quad \begin{array}{l} -R_{1,3}(\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array}) \\ +R_{1,2}R_{2,3}(\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array}) \\ -R_{1,2}R_{1,3}R_{2,3}(\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array}) \end{array} \quad \begin{array}{l} -R_{2,3}(\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array}) \\ +R_{1,3}R_{2,3}(\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array}) \end{array} = \begin{array}{l} -\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array} \\ +\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array} \end{array} \quad \begin{array}{l} -\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array} \\ +\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array} \\ -0 \end{array} \quad \begin{array}{l} -\begin{array}{|c|c|} \hline \square & \square \\ \hline \end{array} \\ +0 \end{array}$$

Adding it all together, we get

Solution

$$s_{\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array}} = \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array} - \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array} - \begin{array}{|c|c|} \hline \square & \square \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array}$$

Why Schur functions?

- Schur functions encode the possible ways certain abstract algebraic objects appear in n -dimensional space. (If $n = 3$, we have 3D space.)

Why Schur functions?

- Schur functions encode the possible ways certain abstract algebraic objects appear in n -dimensional space. (If $n = 3$, we have 3D space.)
- Schur functions make computer computations easier.

Why Schur functions?

- Schur functions encode the possible ways certain abstract algebraic objects appear in n -dimensional space. (If $n = 3$, we have 3D space.)
- Schur functions make computer computations easier.

Problem

However, the formula for Schur functions is complicated. If we have another formula for Schur functions, how can we prove they give the same result?

Multiplication for Symmetric Functions

Let us introduce a rule for multiplication of partition diagrams by “stacking.”

Rule for Multiplication (Example)

$$\begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array} \cdot \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array} = \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} = \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} = \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array} \cdot \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array}$$

Multiplication for Symmetric Functions

Let us introduce a rule for multiplication of partition diagrams by “stacking.”

Rule for Multiplication (Example)

$$\square\square \cdot \square = \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} = \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} = \square \cdot \square\square$$

Schur functions are a sum of partition diagrams, so we can compute

Example

$$\begin{aligned} \color{red}{\square\square\square} \cdot s_{\begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \end{array}} &= \color{red}{\square\square\square} \cdot \left(\begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} - \begin{array}{|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} - \begin{array}{|c|c|} \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \end{array} + \begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \end{array} \right) \\ &= \begin{array}{|c|c|} \hline \color{red}{\square\square\square} & \square \\ \hline \color{red}{\square\square\square} & \square \\ \hline \end{array} - \begin{array}{|c|c|} \hline \color{red}{\square\square\square} & \square & \square \\ \hline \color{red}{\square\square\square} & \square & \square \\ \hline \end{array} - \begin{array}{|c|c|} \hline \color{red}{\square\square\square} & \square & \square & \square \\ \hline \color{red}{\square\square\square} & \square & \square & \square \\ \hline \end{array} + \begin{array}{|c|c|c|c|} \hline \color{red}{\square\square\square} & \square & \square & \square \\ \hline \color{red}{\square\square\square} & \square & \square & \square \\ \hline \end{array} \end{aligned}$$

Multiplication for Symmetric Functions

Let us introduce a rule for multiplication of partition diagrams by “stacking.”

Rule for Multiplication (Example)

$$\begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array} \cdot \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array} = \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} = \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} = \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array} \cdot \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array}$$

Schur functions are a sum of partition diagrams, so we can compute

Example

$$\begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array} \cdot s_{\begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array}} = \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array} \cdot \left(\begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} - \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} - \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \right) \\ = \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} - \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} - \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array}$$

Problem

Result is in terms of partition diagrams, but we would like a result in terms of Schur functions.

The Pieri Rule

Example

$$\begin{array}{c} \color{red}{\square} \color{red}{\square} \color{red}{\square} \color{red}{\square} \\ \color{red}{\square} \end{array} \cdot s_{\begin{array}{c} \square \\ \square \end{array}} = s_{\begin{array}{c} \color{red}{\square} \color{red}{\square} \color{red}{\square} \color{red}{\square} \\ \color{red}{\square} \end{array}} + s_{\begin{array}{c} \color{red}{\square} \color{red}{\square} \color{red}{\square} \color{red}{\square} \\ \color{red}{\square} \end{array}} + s_{\begin{array}{c} \color{red}{\square} \color{red}{\square} \color{red}{\square} \color{red}{\square} \\ \color{red}{\square} \end{array}} + s_{\begin{array}{c} \color{red}{\square} \color{red}{\square} \color{red}{\square} \color{red}{\square} \\ \color{red}{\square} \end{array}}$$

Example

$$\begin{array}{c} \color{red}{\square} \color{red}{\square} \color{red}{\square} \color{red}{\square} \\ \color{red}{\square} \end{array} \cdot S_{\begin{array}{c} \square \\ \square \end{array}} = S_{\begin{array}{c} \color{red}{\square} \color{red}{\square} \color{red}{\square} \\ \color{red}{\square} \color{red}{\square} \end{array}} + S_{\begin{array}{c} \color{red}{\square} \color{red}{\square} \color{red}{\square} \\ \color{red}{\square} \color{red}{\square} \end{array}} + S_{\begin{array}{c} \color{red}{\square} \color{red}{\square} \color{red}{\square} \\ \color{red}{\square} \color{red}{\square} \end{array}} + S_{\begin{array}{c} \color{red}{\square} \color{red}{\square} \color{red}{\square} \\ \color{red}{\square} \color{red}{\square} \end{array}}$$

- In general, we get the result in terms of Schur functions by finding all ways to add the red boxes such that we only add at most one box to each column.

Example

$$s_{(4)} \cdot s_{(2)} = s_{(5,1)} + s_{(4,2)} + s_{(3,3)} + s_{(3,2,1)}$$

- In general, we get the result in terms of Schur functions by finding all ways to add the red boxes such that we only add at most one box to each column.
- We call this method *the Pieri rule* and it is a fundamental property of Schur functions.

One approach to show two formulas for Schur functions are the same:

Proof technique

Base cases are equal

Pieri rules are the same

Linear algebra

Functions are the same!

What do I think about?

- Most problems about Schur functions are solved.

What do I think about?

- Most problems about Schur functions are solved.
- Instead, I think about a class of functions called “type C dual affine Stanley symmetric functions” which have similar properties to Schur functions.

What do I think about?

- Most problems about Schur functions are solved.
- Instead, I think about a class of functions called “type C dual affine Stanley symmetric functions” which have similar properties to Schur functions.
- However, the current formula for these functions is not as concrete as the formula I gave you for Schur functions.

Type C dual affine Stanley symmetric functions

Start with “word” with letters given by colors, $\{\color{red}\blacksquare, \color{blue}\blacksquare, \color{green}\blacksquare\}$. For example, let's use $w = \color{green}\blacksquare\color{blue}\blacksquare\color{green}\blacksquare$.

Type C dual affine Stanley symmetric functions

Start with “word” with letters given by colors, $\{\color{red}\square, \color{blue}\square, \color{green}\square\}$. For example, let's use $w = \color{green}\square\color{blue}\square\color{green}$.

We must find all “subword decompositions” of w that are also subwords of $\rho = \color{blue}\square\color{green}\color{blue}\color{red}\square$ or any of its “rotations” $\color{red}\square\color{blue}\color{green}\square, \color{blue}\square\color{red}\color{blue}\square, \color{green}\square\color{blue}\color{red}\square$.

Type C dual affine Stanley symmetric functions

Start with “word” with letters given by colors, $\{\color{red}\square, \color{blue}\square, \color{green}\square\}$. For example, let's use $w = \color{green}\square\color{blue}\square$.

We must find all “subword decompositions” of w that are also subwords of $\rho = \color{blue}\square\color{green}\color{red}\square$ or any of its “rotations” $\color{red}\square\color{blue}\color{green}\square, \color{blue}\square\color{red}\color{green}\square, \color{green}\square\color{red}\color{blue}\square$.

Example

$\color{green}\square|\color{blue}\square$ is a subword decomposition of w where each part appears as a subword of $\rho = \color{blue}\square\color{green}\color{red}\square$, but $\color{green}\square\color{blue}\square$ is not a subword of ρ or any of its rotations.

Example continued

Then, you take all such subword decompositions to get a formula

The diagram illustrates the decomposition of a 3x3 grid into subwords. On the left, three 3x3 grids are shown with colored cells (green and blue) and vertical lines indicating subword boundaries. The first grid has a blue cell at (1,2) and a green cell at (1,3). The second grid has a blue cell at (1,2) and a green cell at (1,1). The third grid has a blue cell at (1,2) and a green cell at (1,3). An arrow points to the right, where two subword shapes are shown: a 2x2 square and a 1x3 horizontal row. A second arrow points to the right, where the formula $Q^{(2)} = 4 * \text{[2x2 square]} + 8 * \text{[1x3 row]}$ is shown. The subword shapes in the formula are represented by white boxes with black outlines.

Example continued

Then, you take all such subword decompositions to get a formula

The diagram illustrates the decomposition of a 3x3 grid into two Young diagrams. On the left, a 3x3 grid is shown with colored cells: the top row has a green cell, a blue cell, and a green cell; the middle row has a green cell, a blue cell, and a green cell; the bottom row has a green cell, a blue cell, and a green cell. Vertical lines separate the columns. An arrow points to two Young diagrams: the first is a 2x2 square with a cell below it, and the second is a vertical column of three cells. A second arrow points to the formula $Q^{(2)} = 4 * \text{[2x2 grid]} + 8 * \text{[vertical column of 3 cells]}$. The Young diagrams in the formula are shown with colored cells: the 2x2 grid has a green cell at (1,1), a blue cell at (1,2), a blue cell at (2,1), and a green cell at (2,2); the vertical column has a green cell at (1,1), a blue cell at (2,1), and a green cell at (3,1).

But, unfortunately, you are not done!

Example continued

Then, you take all such subword decompositions to get a formula

The diagram illustrates the decomposition of a 3x3 grid into two Young diagrams. On the left, a 3x3 grid is shown with colored cells: the top row has a green cell, a blue cell, and a green cell; the middle row has a green cell, a blue cell, and a green cell; the bottom row has a green cell, a blue cell, and a green cell. A vertical line is drawn between the first and second columns. An arrow points to the right, where two Young diagrams are shown: the first is a 2x2 grid, and the second is a 3x1 grid. A second arrow points to the right, where the formula $Q^{(2)} = 4 * \text{2x2 grid} + 8 * \text{3x1 grid}$ is displayed. The 2x2 grid in the formula has a green cell in the top-left and a blue cell in the bottom-right. The 3x1 grid in the formula has a green cell in the top, a blue cell in the middle, and a green cell in the bottom.

But, unfortunately, you are not done!

Problem

You then have to take the “dual” of this function to get the Type C dual affine Stanley symmetric function, $P_{\text{grid}}^{(2)}$. This process is not direct and not computationally straightforward.

What have I done?

- I have a conjectured formula that describes type C dual affine Stanley symmetric functions ($P_w^{(n)}$) directly using raising operators.

What have I done?

- I have a conjectured formula that describes type C dual affine Stanley symmetric functions ($P_w^{(n)}$) directly using raising operators.
- Computational evidence suggests my conjecture is correct.

What have I done?

- I have a conjectured formula that describes type C dual affine Stanley symmetric functions ($P_w^{(n)}$) directly using raising operators.
- Computational evidence suggests my conjecture is correct.
- However, proving the formulas are the same directly would be quite hard, so instead I am seeking to use the Pieri rule approach

Base cases are equal

Pieri rules are the same

Linear algebra

Functions are the same!

Thank you for your support and for listening!



Jefferson Scholars Foundation

Symmetric Functions?

I pulled the wool over your eyes. Our partition diagrams represent polynomial functions with an infinite number of variables and an infinite number of terms.

Dictionary

$$\square \rightarrow h_{\square}(x_1, x_2, x_3, \dots) = x_1 + x_2 + x_3 + \dots$$

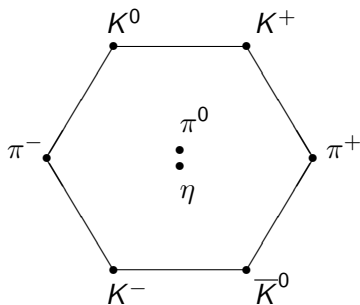
$$\square\square \rightarrow h_{\square\square}(x_1, x_2, x_3, \dots) = x_1^2 + x_1x_2 + x_1x_3 + \dots \\ + x_2^2 + x_2x_3 + \dots \\ + x_3^2 + x_3x_4 + \dots$$

$$\square\square\square \rightarrow h_{\square\square\square}(x_1, x_2, x_3, \dots) = x_1^3 + x_1^2x_2 + x_1x_2^2 + \dots$$

⋮

$$s_{\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & & \\ \hline \square & & \\ \hline \end{array}} = h_{\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & & \\ \hline \square & & \\ \hline \end{array}} - h_{\begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline \square & & & \\ \hline \square & & & \\ \hline \end{array}} - h_{\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \square & & \\ \hline \end{array}} + h_{\begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline \square & & & \\ \hline \square & & & \\ \hline \end{array}}$$

Applications?



The “eightfold way” from particle physics is encoded in Schur functions by

$$s_{\begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \\ \hline \end{array}}(e_{\epsilon_1}, e_{\epsilon_2}, e_{\epsilon_3})$$