

# 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?

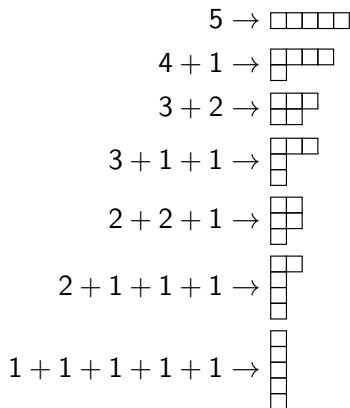
# Partitions of 5

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

$$\begin{array}{lcl} 5 & \rightarrow & \begin{array}{|c|c|c|c|c|} \hline & & & & \\ \hline \end{array} \\ 4 + 1 & \rightarrow & \begin{array}{|c|c|c|c|} \hline & & & \\ \hline \end{array} \\ 3 + 2 & \rightarrow & \begin{array}{|c|c|} \hline & \\ \hline \end{array} \\ 3 + 1 + 1 & \rightarrow & \begin{array}{|c|c|} \hline & \\ \hline \end{array} \\ 2 + 2 + 1 & \rightarrow & \begin{array}{|c|c|} \hline & \\ \hline \end{array} \\ 2 + 1 + 1 + 1 & \rightarrow & \begin{array}{|c|} \hline \\ \hline \end{array} \\ 1 + 1 + 1 + 1 + 1 & \rightarrow & \begin{array}{|c|} \hline \\ \hline \end{array} \end{array}$$

# 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.

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

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

# Raising Operators

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

We can change partition diagrams by moving boxes.

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

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

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  $\lambda$  with  $\ell$  rows, we have definition



# Schur functions

We define a new class of functions. Given a partition diagram  $\lambda$  with  $\ell$  rows, we have definition

## Definition

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

# Schur functions

We define a new class of functions. Given a partition diagram  $\lambda$  with  $\ell$  rows, we have definition

## Definition

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

## Example

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

## Example

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

# Example continued

## Example

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

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 continued

## Example

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

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

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

## Example continued

### Example

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

# Example continued

## Example

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

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

# Example continued

## Example

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

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



# Example continued

## Example

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

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

Adding it all together, we get

## Solution

$$s_{\begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}} = \begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix} - \begin{smallmatrix} \square & \square & \square \\ \square & \square & \square \end{smallmatrix} - \begin{smallmatrix} \square & \square & \square \\ \square & \square & \square \end{smallmatrix} + \begin{smallmatrix} \square & \square & \square \\ \square & \square & \square \end{smallmatrix}$$

# 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)

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

# Multiplication for Symmetric Functions

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

## Rule for Multiplication (Example)

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

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

## Example

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

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

# Multiplication for Symmetric Functions

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

## Rule for Multiplication (Example)

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

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

## Example

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

## Problem

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

# The Pieri Rule

## Example

$$\begin{array}{|c|c|c|c|} \hline \color{red}{\blacksquare} & \color{red}{\blacksquare} & \color{red}{\blacksquare} & \color{red}{\blacksquare} \\ \hline \end{array} \cdot s_{\begin{array}{|c|c|} \hline \square & \square \\ \hline \end{array}} = s_{\begin{array}{|c|c|c|c|} \hline \color{red}{\blacksquare} & \color{red}{\blacksquare} & \color{red}{\blacksquare} & \color{red}{\blacksquare} \\ \hline \color{red}{\blacksquare} & & & \\ \hline \end{array}} + s_{\begin{array}{|c|c|c|c|} \hline \color{red}{\blacksquare} & \color{red}{\blacksquare} & \color{red}{\blacksquare} & \color{red}{\blacksquare} \\ \hline & \color{red}{\blacksquare} & & \\ \hline \end{array}} + s_{\begin{array}{|c|c|c|c|} \hline \color{red}{\blacksquare} & \color{red}{\blacksquare} & \color{red}{\blacksquare} & \color{red}{\blacksquare} \\ \hline & & \color{red}{\blacksquare} & \\ \hline \end{array}} + s_{\begin{array}{|c|c|c|c|} \hline \color{red}{\blacksquare} & \color{red}{\blacksquare} & \color{red}{\blacksquare} & \color{red}{\blacksquare} \\ \hline & & & \color{red}{\blacksquare} \\ \hline \end{array}}$$



# The Pieri Rule

## Example

$$\begin{array}{|c|c|c|c|} \hline \text{red} & \text{red} & \text{red} & \text{red} \\ \hline \end{array} \cdot s_{\begin{array}{|c|c|} \hline \square & \square \\ \hline \end{array}} = s_{\begin{array}{|c|c|c|} \hline \text{red} & \text{red} & \text{red} \\ \hline \square & & \\ \hline \end{array}} + s_{\begin{array}{|c|c|c|} \hline \text{red} & \text{red} & \text{red} \\ \hline \square & \text{red} & \\ \hline \end{array}} + s_{\begin{array}{|c|c|c|} \hline \text{red} & \text{red} & \text{red} \\ \hline \square & & \text{red} \\ \hline \end{array}} + s_{\begin{array}{|c|c|c|} \hline \text{red} & \text{red} & \text{red} \\ \hline \square & \text{red} & \text{red} \\ \hline \end{array}}$$

- 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.

# The Pieri Rule

## Example

$$\begin{array}{|c|c|c|c|} \hline \text{red} & \text{red} & \text{red} & \text{red} \\ \hline \end{array} \cdot s_{\begin{array}{|c|c|} \hline \square & \square \\ \hline \end{array}} = s_{\begin{array}{|c|c|c|} \hline \text{red} & \text{red} & \text{red} \\ \hline \square & \square & \square \\ \hline \end{array}} + s_{\begin{array}{|c|c|c|c|} \hline \text{red} & \text{red} & \text{red} & \text{red} \\ \hline \square & \square & \square & \square \\ \hline \end{array}} + s_{\begin{array}{|c|c|c|c|c|} \hline \text{red} & \text{red} & \text{red} & \text{red} & \text{red} \\ \hline \square & \square & \square & \square & \square \\ \hline \end{array}} + s_{\begin{array}{|c|c|c|c|c|c|} \hline \text{red} & \text{red} & \text{red} & \text{red} & \text{red} & \text{red} \\ \hline \square & \square & \square & \square & \square & \square \\ \hline \end{array}}$$

- 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.
- 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

```
graph TD; A[Base cases are equal] --> C[Linear algebra]; B[Pieri rules are the same] --> C; C --> D[Functions 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,  $\{\text{red}, \text{blue}, \text{green}\}$ . For example, let's use  $w = \text{green blue green}$ .

# Type C dual affine Stanley symmetric functions

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

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



# Type C dual affine Stanley symmetric functions

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

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

## Example

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

# Example continued

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

$$\begin{array}{c} \begin{array}{|c|c|} \hline \text{green} & \text{blue} \\ \hline \end{array} \begin{array}{|c|} \hline \text{green} \\ \hline \end{array} \\ \begin{array}{|c|c|c|} \hline \text{green} & \text{blue} & \text{green} \\ \hline \end{array} \\ \begin{array}{|c|c|c|} \hline \text{green} & \text{blue} & \text{green} \\ \hline \end{array} \end{array} \rightarrow \begin{array}{c} \begin{array}{|c|c|} \hline & \\ \hline \end{array} \\ \begin{array}{|c|} \hline \\ \hline \end{array} \\ \begin{array}{|c|} \hline \\ \hline \end{array} \end{array} \rightarrow Q^{(2)}_{\begin{array}{|c|c|} \hline \text{green} & \text{blue} \\ \hline \end{array}} = 4 * \begin{array}{|c|c|} \hline & \\ \hline \end{array} + 8 * \begin{array}{|c|} \hline \\ \hline \end{array}$$

# Example continued

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

$$\begin{array}{c} \begin{array}{|c|} \hline \begin{array}{cc} \text{green} & \text{blue} \end{array} \\ \hline \end{array} \begin{array}{|c|} \hline \text{green} \\ \hline \end{array} \\ \begin{array}{|c|} \hline \text{green} \\ \hline \end{array} \begin{array}{|c|} \hline \begin{array}{cc} \text{blue} & \text{green} \end{array} \\ \hline \end{array} \\ \begin{array}{|c|} \hline \text{green} \\ \hline \end{array} \begin{array}{|c|} \hline \begin{array}{cc} \text{blue} & \text{green} \end{array} \\ \hline \end{array} \end{array} \rightarrow \begin{array}{c} \begin{array}{|c|c|} \hline & \\ \hline \end{array} \\ \begin{array}{|c|} \hline \\ \hline \end{array} \\ \begin{array}{|c|} \hline \\ \hline \end{array} \end{array} \rightarrow Q^{(2)}_{\begin{array}{|c|c|} \hline \text{green} & \text{blue} & \text{green} \\ \hline \end{array}} = 4 * \begin{array}{|c|c|} \hline & \\ \hline \end{array} + 8 * \begin{array}{|c|} \hline \\ \hline \end{array}$$

But, unfortunately, you are not done!

## Example continued

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

$$\begin{array}{c} \begin{array}{|c|} \hline \begin{array}{cc} \text{green} & \text{blue} \end{array} \\ \hline \end{array} \begin{array}{|c|} \hline \text{green} \\ \hline \end{array} \\ \begin{array}{|c|} \hline \text{green} \\ \hline \end{array} \begin{array}{|c|} \hline \begin{array}{cc} \text{blue} & \text{green} \end{array} \\ \hline \end{array} \\ \begin{array}{|c|} \hline \text{green} \\ \hline \end{array} \begin{array}{|c|} \hline \begin{array}{cc} \text{blue} & \text{green} \end{array} \\ \hline \end{array} \end{array} \rightarrow \begin{array}{c} \begin{array}{|c|c|} \hline & \\ \hline \end{array} \\ \begin{array}{|c|} \hline \\ \hline \end{array} \\ \begin{array}{|c|} \hline \\ \hline \end{array} \end{array} \rightarrow Q^{(2)}_{\begin{array}{|c|c|} \hline \text{green} & \text{blue} & \text{green} \\ \hline \end{array}} = 4 * \begin{array}{|c|c|} \hline & \\ \hline \end{array} + 8 * \begin{array}{|c|} \hline \\ \hline \end{array}$$

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,  $P^{(2)}_{\begin{array}{|c|c|} \hline \text{green} & \text{blue} & \text{green} \\ \hline \end{array}}$ . 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

```
graph TD; A[Base cases are equal] -- solid arrow --> C[Linear algebra]; B[Pieri rules are the same] -. dashed arrow .-> C; C -- solid arrow --> D[Functions 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

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

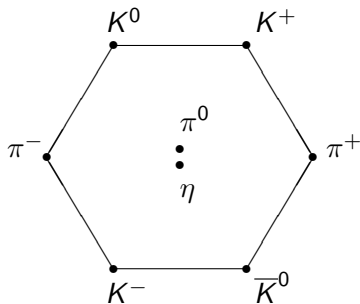
$$\square\square \rightarrow h_{\square\square}(x_1, x_2, x_3, \dots) = x_1^2 + x_1x_2 + x_1x_3 + \dots \\ + x_2^2 + x_2x_3 + \dots \\ + x_3^2 + x_3x_4 + \dots$$

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

$\vdots$

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

# Applications?



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

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