Computation of the Character Table of the Symmetric Group

Xingyou Zhou

Advisor: Stavros Garoufalidis

2022-05-19

Contents

Introduction

Introduction

Character Table

Background

- Character Table

Definition. If V is a representation of a group G, its ***character*** χ_V is the complex-valued function on the group defined by

$$\chi_V(g) = Tr(g|_V)$$

where $Tr(g|_V)$ is the trace of g on V.

Definition. For representation $G \to GL(V)$ the character table is a square matrix:

1. the first row lists the conjugacy classes of G,

2. the first column lists the irreducible representations of V,

3. the rest entires are character values each corresponding the conjugacy class and irreducible representation.

Introduction

Character Table

Background

- Character Table

Theorem 1. The conjugacy classes of symmetric group S_n are determined by partitions.

Theorem 2. The irreducible representations of symmetric group S_n are determined by partitions.

Example. S_3 : { (123),(132),(12),(13),(23),() }

Partition	Conjugacy Classes	Irreducible Representations
(1,1,1)	()	alternating
(2,1)	(12), (13), (23)	standard
(3)	(123), (132)	trivial

Introduction

Character Table

Background

- Character Table

Example.

The character table of symmetric group S_3

	(1,1,1)	(2,1)	(3)
trivial	1	1	1
standard	2	0	-1
alternating	1	-1	1

Character Table of S_3

The first row lists the *conjugacy classes* of S_3.

The first column lists the *irreducible representations* of S_3.

The rest entires are *character values* each corresponding the conjugacy class and irreducible representation.

Introduction

Character Table

Background

- Background

Character Table is very useful!

In chemistry, crystallography, and spectroscopy, character tables of point groups are used to classify e.g. molecular vibrations according to their symmetry, and to predict whether a transition between two states is forbidden for symmetry reasons.

Many university level textbooks on physical chemistry, quantum chemistry, spectroscopy and inorganic chemistry devote a chapter to the use of symmetry group character tables.

---- Wikipedia

Problem

Sagemath, GAP, etc., are too slow to compute the table.
We need a specific algorithm to reduce computation time.

Introduction

Character Table

Background

- Background

Test Results for Sagemath

Symmetric Group S_n $(n = 20, 25, 30)$

```
In [8]: start = time.time()
        SymmetricGroup(20).character_table()
        print("spend time: ", time.time() - start, "s")

        spend time:  16.497802019119263 s

In [9]: start = time.time()
        SymmetricGroup(25).character_table()
        print("spend time: ", time.time() - start, "s")

        spend time:  162.95479273796082 s

In [10]: start = time.time()
         SymmetricGroup(30).character_table()
         print("spend time: ", time.time() - start, "s")

         spend time:  1615.2928960546875 s
```

S_n	Time spent (s)
S_{20}	16.49
S_{25}	162.95
S_{30}	1615.29

How To Compute

How To Compute

Nakayama's rule

Young Diagram

Young Tableau

Border Strip

Border Strip Tableau

- Murnaghan–Nakayama Rule

Since partition can determine the conjugacy class and irreducible representation of S_n , we let partition λ specify the conjugacy class, and ρ the irreducible representation, χ_ρ^λ the corresponding character value.

Then we have **Murnaghan–Nakayama Rule:**

Non-Recursive Version

$$\chi_\rho^\lambda = \sum_{T \in BST(\lambda,\rho)} (-1)^{ht(T)}$$

Recursive Version

$$\chi_\rho^\lambda = \sum_{\mu \in BS(\lambda,\rho_1)} (-1)^{ht(\mu)} \chi_{\rho\backslash\rho_1}^{\lambda\backslash\mu}$$

with stop condition: $\chi_{()}^{()} = 1$.

How To Compute

Nakayama's rule

Young Diagram

Young Tableau

Border Strip

Border Strip Tableau

Murnaghan–Nakayama Rule

Non-Recursive Version

$$\chi_\rho^\lambda = \sum_{T \in BST(\lambda,\rho)} (-1)^{ht(T)}$$

Recursive Version

$$\chi_\rho^\lambda = \sum_{\mu \in BS(\lambda,\rho_1)} (-1)^{ht(\mu)} \chi_{\rho\backslash\rho_1}^{\lambda\backslash\mu}$$

stop condition: $\chi_{()}^{()} = 1$

- Young Diagram

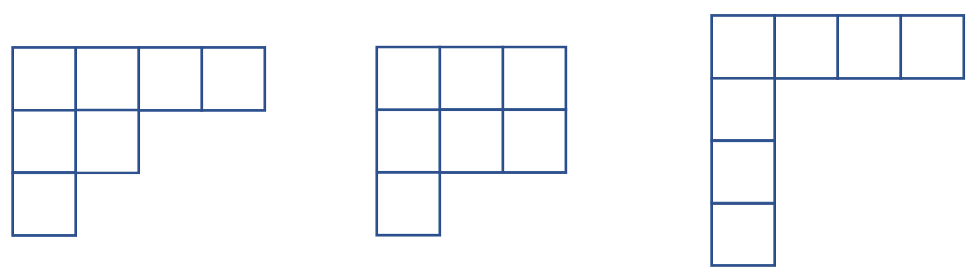
Definition. A **partition** is a sequence $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_l)$ of nonnegative integers in decreasing order: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_l > 0$, l is the length of λ, $|\lambda| = \sum_{i=1}^{l} \lambda_i$ is the size of λ.

Example. (size: 7)

$$\lambda = (4,2,1)\ ,\ \mu = (3,3,1)\ ,\ \rho = (4,1,1,1)$$

Definition. **Young diagram** is a finite collection of boxes, which contains λ_i boxes in row i for a given partition λ.

Example.



$(4,2,1)$ $(3,3,1)$ $(4,1,1,1)$

How To Compute

Nakayama's rule

Young Diagram

Young Tableau

Border Strip

Border Strip Tableau

Murnaghan–Nakayama Rule

Non-Recursive Version

$$\chi_{\rho}^{\lambda} = \sum_{T \in BST(\lambda,\rho)} (-1)^{ht(T)}$$

Recursive Version

$$\chi_{\rho}^{\lambda} = \sum_{\mu \in BS(\lambda,\rho_1)} (-1)^{ht(\mu)} \chi_{\rho \backslash \rho_1}^{\lambda \backslash \mu}$$

stop condition: $\chi_{()}^{()} = 1$

- Young Tableau

Definition. A **Young tableau** is obtained by filling in the boxes of the Young diagram with symbols taken from some ordered alphabet. We say it is **standard** (SYT) if the entries in each row and each column are increasing.

Example.

2	1	3	4
7	5		
6			

(4,2,1)

1	2	3
4	5	6
7		

(3,3,1)

1	5	6	7
2			
3			
4			

(4,1,1,1)

How To Compute

Nakayama's rule

Young Diagram

Young Tableau

Border Strip

Border Strip Tableau

Murnaghan–Nakayama Rule

Non-Recursive Version

$$\chi_\rho^\lambda = \sum_{T \in BST(\lambda,\rho)} (-1)^{ht(T)}$$

Recursive Version

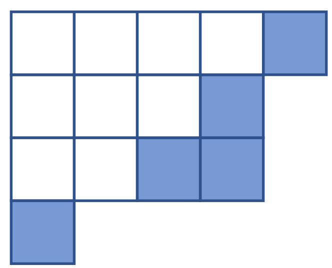
$$\chi_\rho^\lambda = \sum_{\mu \in BS(\lambda,\rho_1)} (-1)^{ht(\mu)} \chi_{\rho \backslash \rho_1}^{\lambda \backslash \mu}$$

stop condition: $\chi_{()}^{()} = 1$

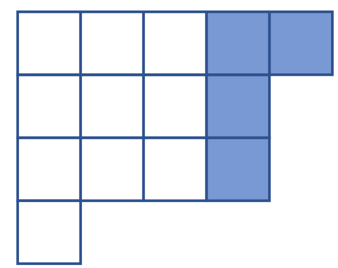
- Border Strip (BS)

Definition. Let λ, μ be partitions with $\lambda \supseteq \mu$, then the set-theoretic difference $\lambda \backslash \mu$ is called a **skew diagram**. A skew diagram is border strip if it is connected (two boxes have a common edge) and does not contain a 2×2-block of boxes. The height $ht(\lambda)$ of a border strip is the number of rows it touches minus one. We define $BS(\lambda, l)$ as the set of all border strips with length $l = |\lambda \backslash \mu|$ in partition λ.

Example.



$(5,4,4,1)\backslash(4,3,2)$



$(5,4,4,1)\backslash(3,3,3,1)$

The height of border strip $\rho = (5,4,4,1)\backslash(3,3,3,1)$

$$ht(\rho) = 3 - 1 = 2$$

How To Compute

Nakayama's rule

Young Diagram

Young Tableau

Border Strip

Border Strip Tableau

Murnaghan–Nakayama Rule

Non-Recursive Version

$$\chi_\rho^\lambda = \sum_{T \in BST(\lambda,\rho)} (-1)^{ht(T)}$$

Recursive Version

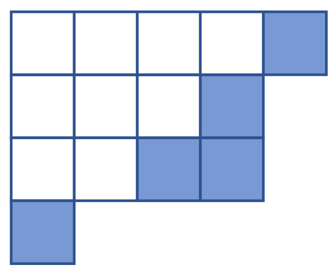
$$\chi_\rho^\lambda = \sum_{\mu \in \boxed{BS(\lambda,\rho_1)}} (-1)^{\boxed{ht(\mu)}} \chi_{\boxed{\rho\setminus\rho_1}}^{\boxed{\lambda\setminus\mu}}$$

stop condition: $\chi_{()}^{()} = 1$

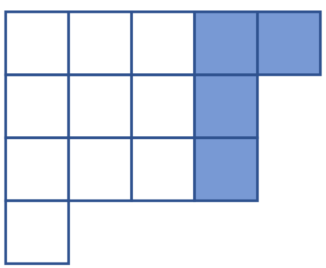
- Border Strip (BS)

Definition. Let λ, μ be partitions with $\lambda \supseteq \mu$, then the set-theoretic difference $\boxed{\lambda\setminus\mu}$ is called a **skew diagram**. A skew diagram is border strip if it is connected (two boxes have a common edge) and does not contain a 2×2-block of boxes. The height $\boxed{ht(\lambda)}$ of a border strip is the number of rows it touches minus one. We define $\boxed{BS(\lambda, l)}$ as the set of all border strips with length $l = |\lambda\setminus\mu|$ in partition λ.

Example.



$(5,4,4,1)\setminus(4,3,2)$



$(5,4,4,1)\setminus(3,3,3,1)$

The height of border strip $\rho = (5,4,4,1)\setminus(3,3,3,1)$

$$ht(\rho) = 3 - 1 = 2$$

How To Compute

Nakayama's rule

Young Diagram

Young Tableau

Border Strip

Border Strip Tableau

Murnaghan–Nakayama Rule

Non-Recursive Version

$$\chi_\rho^\lambda = \sum_{T \in BST(\lambda,\rho)} (-1)^{ht(T)}$$

Recursive Version

$$\chi_\rho^\lambda = \sum_{\mu \in BS(\lambda,\rho_1)} (-1)^{ht(\mu)} \chi_{\rho \backslash \rho_1}^{\lambda \backslash \mu}$$

stop condition: $\chi_{()}^{()} = 1$

- Border Strip Tableau (BST)

Definition. The border strip tableau $T(\lambda,\rho)$ of a partition λ and a weight partition ρ is a filling of the Young diagram λ such that exactly ρ_i boxes are labeled, and rows and columns are non-decreasing. Moreover, for each i the boxes labeled i must form a border strip. The height $ht(T)$ is the sum of the heights of the border strips in tableau $T(\lambda,\rho)$, i.e., $ht(T) = \sum_{\mu \in T(\lambda,\rho)} ht(\mu)$. We define $BST(\lambda,\rho)$ as the set of all possible border strip tableaux determined by partition λ and ρ . i.e., $BST(\lambda,\rho) = \{T_1(\lambda,\rho), T_2(\lambda,\rho), \cdots\}$.

Example. (size: 8) $\lambda = (5,2,1)$, $\rho = (3,3,1,1)$

1	1	1	3	4
2	2			
2				

T_1

1	1	2	2	2
1	3			
4				

T_2

1	1	2	2	2
1	4			
3				

T_3

1	2	2	3	4
1	2			
1				

T_4

1	2	2	2	4
1	3			
1				

T_5

1	2	2	2	3
1	4			
1				

T_6

The height of T_1: $ht(T_1) = \sum_{\mu \in T_1(\lambda,\rho)} ht(\mu) = 0 + 1 + 0 + 0 = 1$

How To Compute

Nakayama’s rule

Young Diagram

Young Tableau

Border Strip

Border Strip Tableau

Murnaghan–Nakayama Rule

Non-Recursive Version

$$\chi_\rho^\lambda = \sum_{T \in \boxed{BST(\lambda,\rho)}} (-1)^{\boxed{ht(T)}}$$

Recursive Version

$$\chi_\rho^\lambda = \sum_{\mu \in BS(\lambda,\rho_1)} (-1)^{ht(\mu)} \chi_{\rho \backslash \rho_1}^{\lambda \backslash \mu}$$

stop condition: $\chi_{()}^{()} = 1$

- Border Strip Tableau (BST)

Definition. The border strip tableau $T(\lambda, \rho)$ of a partition λ and a weight partition ρ is a filling of the Young diagram λ such that exactly ρ_i boxes are labeled, and rows and columns are non-decreasing. Moreover, for each i the boxes labeled i must form a border strip. The height $ht(T)$ is the sum of the heights of the border strips in tableau $T(\lambda, \rho)$, i.e., $\boxed{ht(T)} = \sum_{\mu \in T(\lambda,\rho)} ht(\mu)$. We define $\boxed{BST(\lambda, \rho)}$ as the set of all possible border strip tableaux determined by partition λ and ρ . i.e., $BST(\lambda, \rho) = \{T_1(\lambda, \rho), T_2(\lambda, \rho), \cdots\}$.

Example. (size: 8) $\lambda = (5,2,1)$, $\rho = (3,3,1,1)$

1	1	1	3	4
2	2			
2				

T_1

1	1	2	2	2
1	3			
4				

T_2

1	1	2	2	2
1	4			
3				

T_3

1	2	2	3	4
1	2			
1				

T_4

1	2	2	2	4
1	3			
1				

T_5

1	2	2	2	3
1	4			
1				

T_6

The height of T_1: $ht(T_1) = \sum_{\mu \in T_1(\lambda,\rho)} ht(\mu) = 0 + 1 + 0 + 0 = 1$

Examples

Example

Non-Recursive Version: (size: 8) $\lambda = (5,2,1)$, $\rho = (3,3,1,1)$

1	1	1	3	4
2	2			
2				

T_1

1	1	2	2	2
1	3			
4				

T_2

1	1	2	2	2
1	4			
3				

T_3

1	2	2	3	4
1	2			
1				

T_4

1	2	2	2	4
1	3			
1				

T_5

1	2	2	2	3
1	4			
1				

T_6

$ht(T_1) = \sum_{\mu \in T_1(\lambda,\rho)} ht(\mu) = 0 + 1 + 0 + 0 = 1$

$ht(T_2) = \sum_{\mu \in T_2(\lambda,\rho)} ht(\mu) = 1 + 0 + 0 + 0 = 1$

$ht(T_3) = \sum_{\mu \in T_3(\lambda,\rho)} ht(\mu) = 1 + 0 + 0 + 0 = 1$

$ht(T_4) = \sum_{\mu \in T_4(\lambda,\rho)} ht(\mu) = 2 + 1 + 0 + 0 = 3$

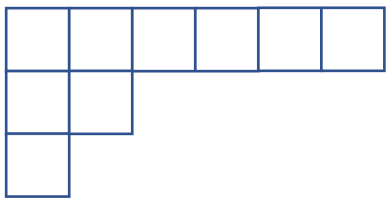
$ht(T_5) = \sum_{\mu \in T_5(\lambda,\rho)} ht(\mu) = 2 + 0 + 0 + 0 = 2$

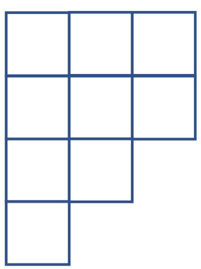
$ht(T_6) = \sum_{\mu \in T_6(\lambda,\rho)} ht(\mu) = 2 + 0 + 0 + 0 = 2$

$$\begin{aligned}\chi^{(5,2,1)}_{(3,3,1,1)} &= \sum_{T \in BST(\lambda,\rho)} (-1)^{ht(T)} \\ &= (-1)^1 + (-1)^1 + (-1)^1 + (-1)^3 + (-1)^2 + (-1)^2 \\ &= -2\end{aligned}$$

Example

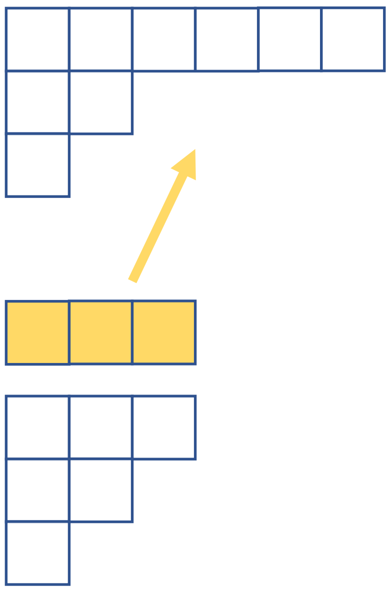
Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3,3,2,1)$





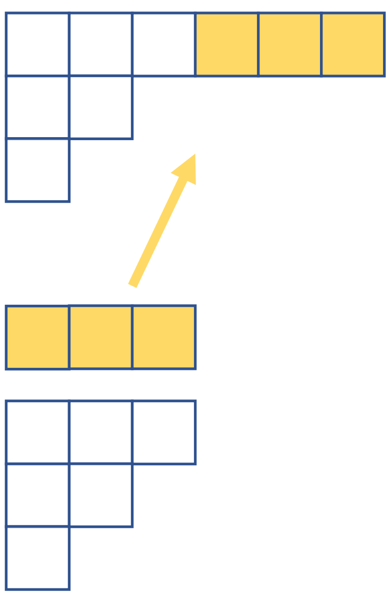
Example

Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (\boxed{3}\,3,2,1)$



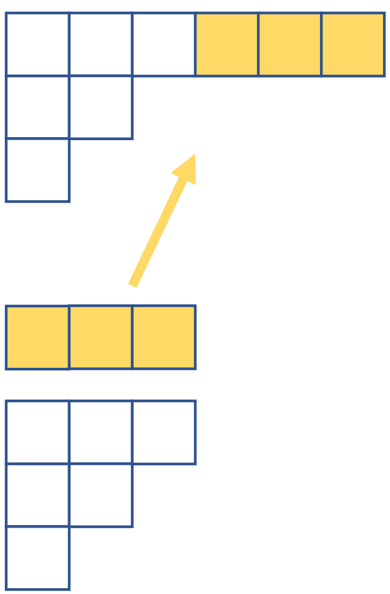
Example

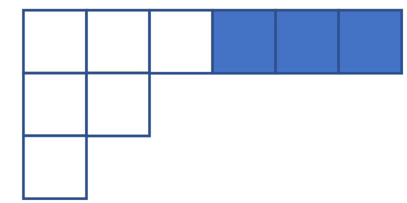
Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (\boxed{3}\,3,2,1)$



Example

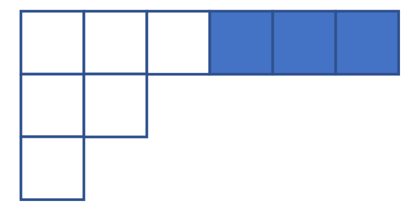
Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (\boxed{3}\,3,2,1)$

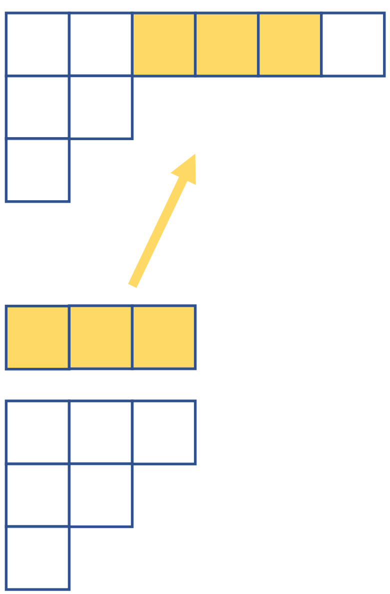




Example

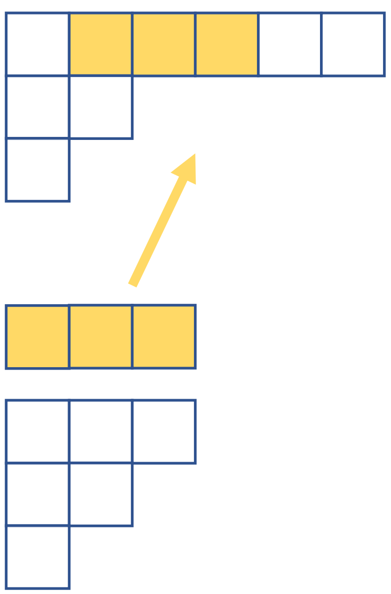
Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (\boxed{3}\,3,2,1)$

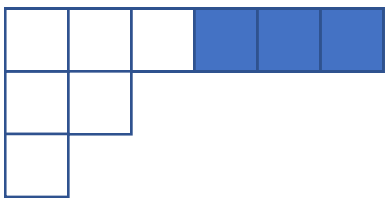




Example

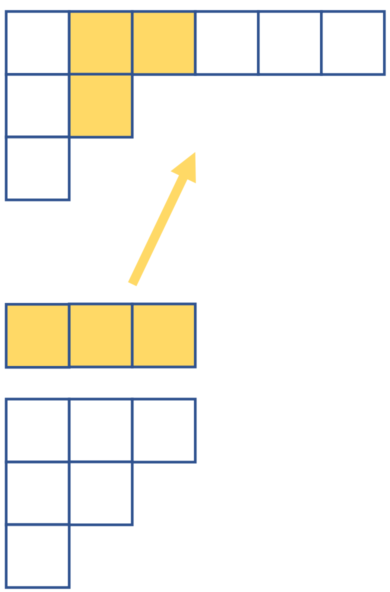
Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (\boxed{3}\,3,2,1)$

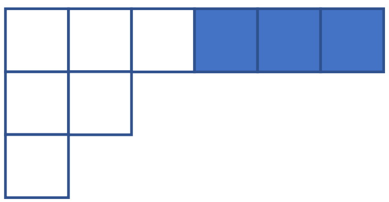




Example

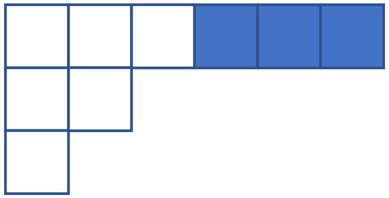
Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (\boxed{3}\,3,2,1)$

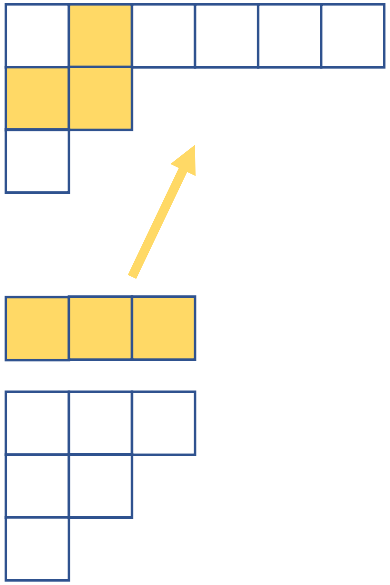




Example

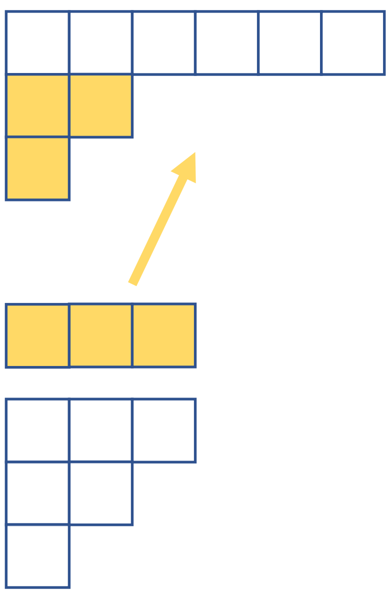
Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (\boxed{3}\,3,2,1)$

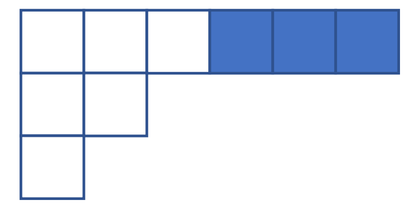




Example

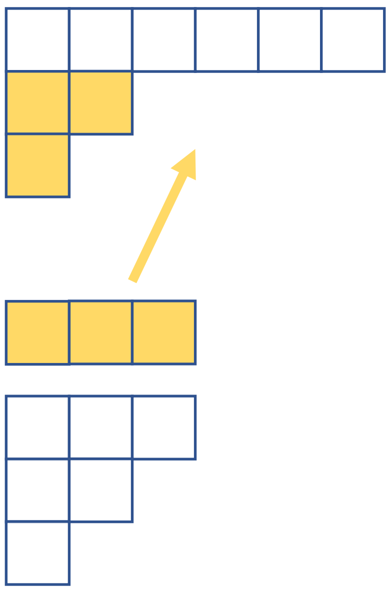
Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (\boxed{3}\,3,2,1)$

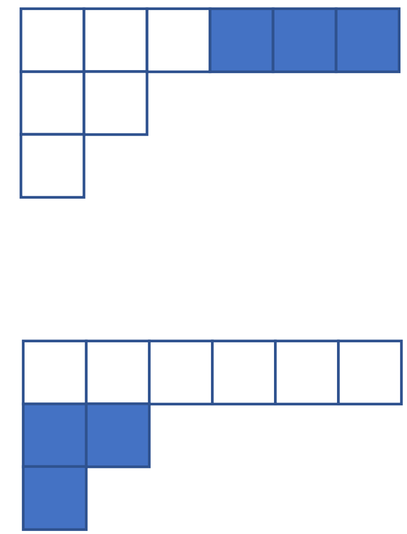




Example

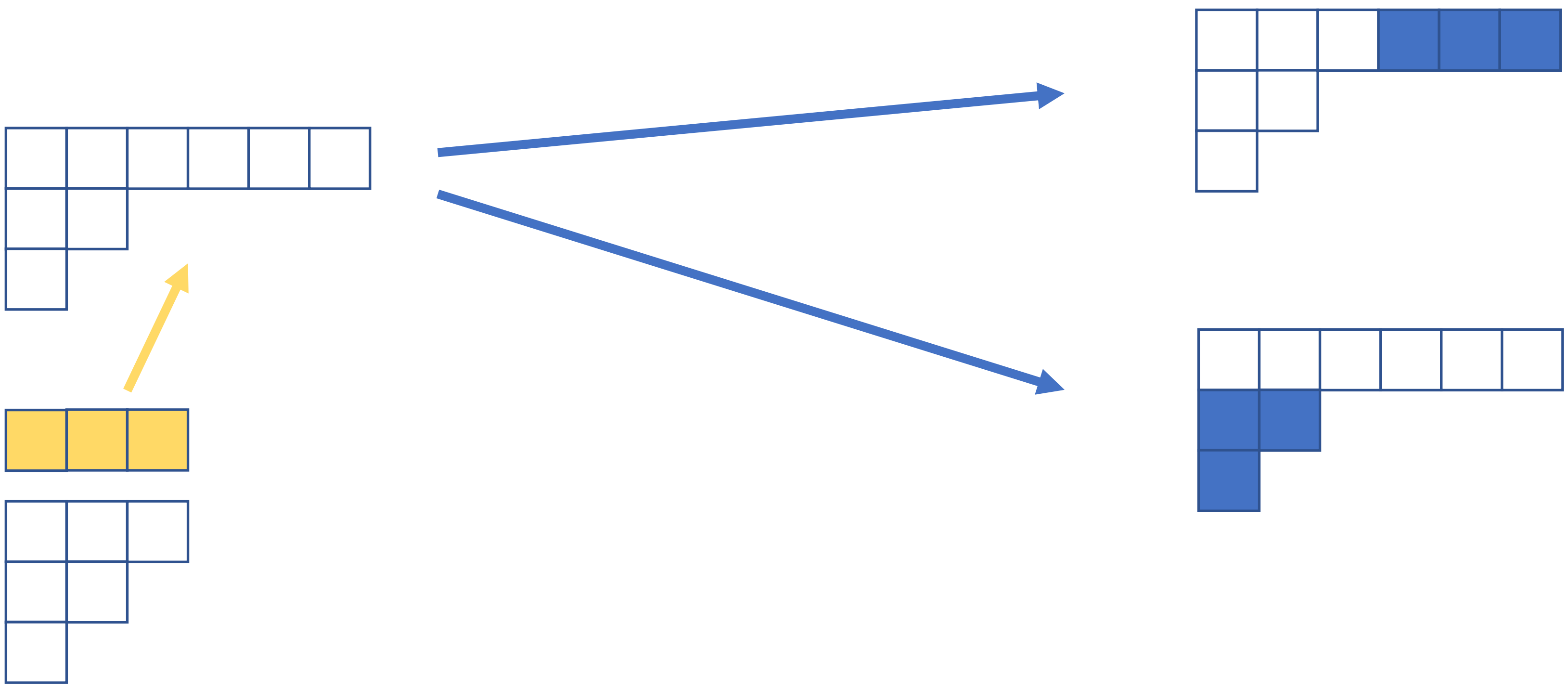
Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (\boxed{3}\,3,2,1)$





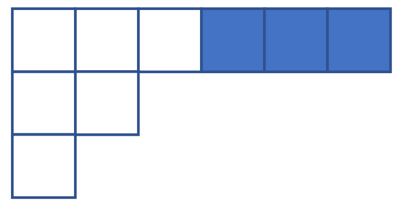
Example

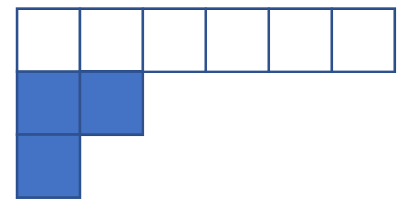
Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (\boxed{3},3,2,1)$



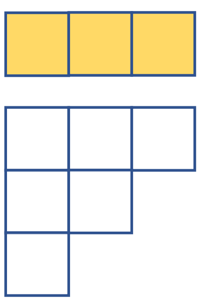
Example

Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (\boxed{3},3,2,1)$



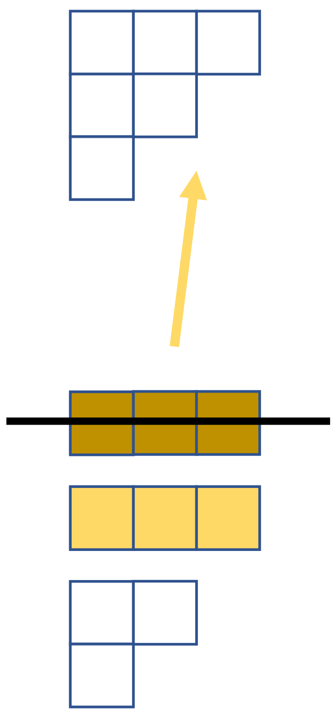


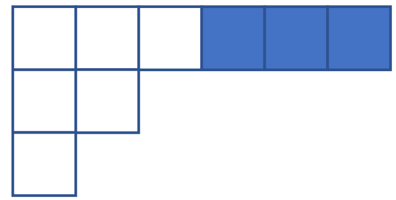
$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 \chi^{(3,2,1)}_{(3,2,1)} + (-1)^1 \chi^{(5)}_{(3,2,1)}$$

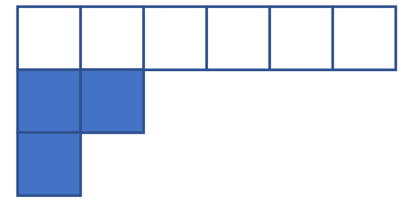


Example

Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3\,\boxed{3}\,2,1)$



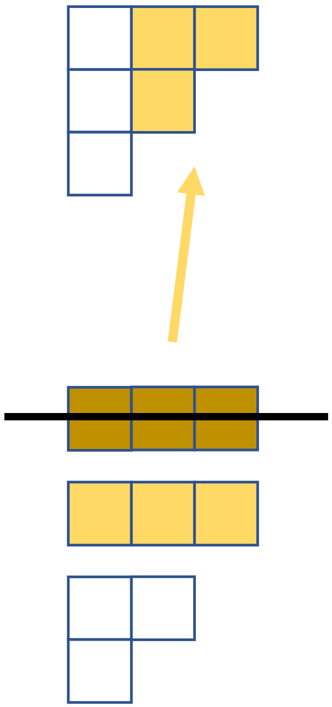


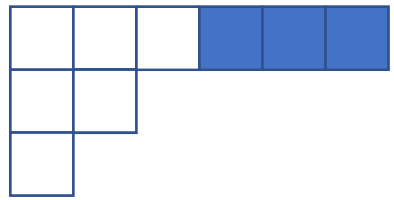


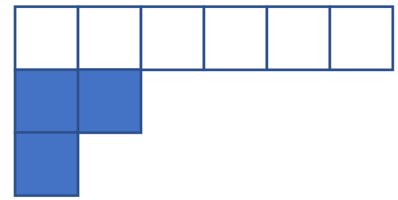
$$\chi^{(6,2,1)}_{(3,3,2,1)} \quad = \quad (-1)^0 \chi^{(3,2,1)}_{(3,2,1)} \quad + \quad (-1)^1 \chi^{(5)}_{(3,2,1)}$$

Example

Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3\,\boxed{3}\,2,1)$



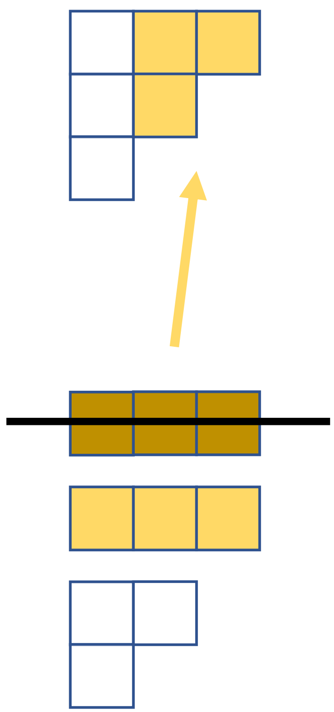


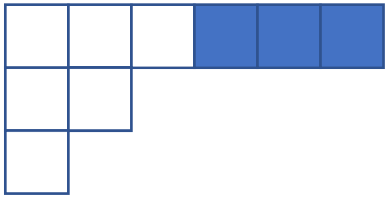


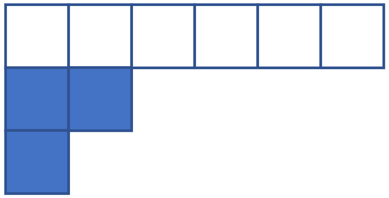
$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0\chi^{(3,2,1)}_{(3,2,1)} + (-1)^1\chi^{(5)}_{(3,2,1)}$$

Example

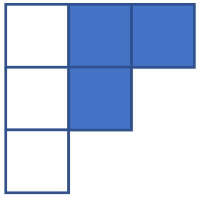
Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3\,\boxed{3}\,2,1)$





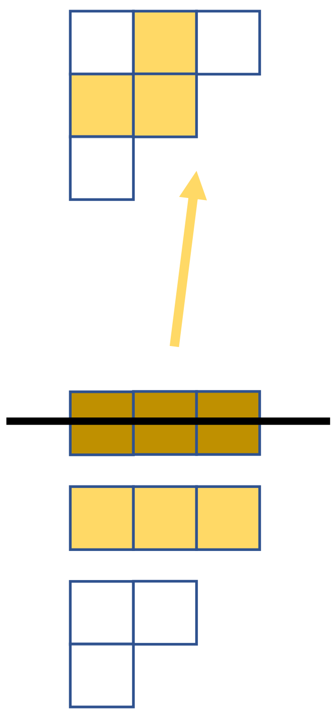


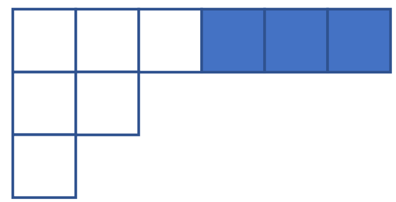
$$\chi^{(6,2,1)}_{(3,3,2,1)} \quad = \quad (-1)^0 \chi^{(3,2,1)}_{(3,2,1)} \quad + \quad (-1)^1 \chi^{(5)}_{(3,2,1)}$$

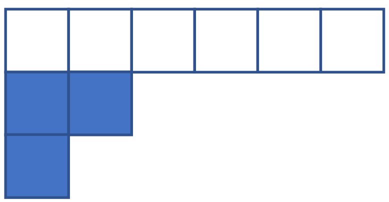


Example

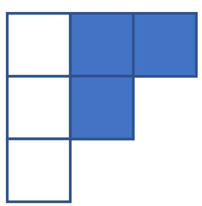
Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3\,\boxed{3}\,2,1)$





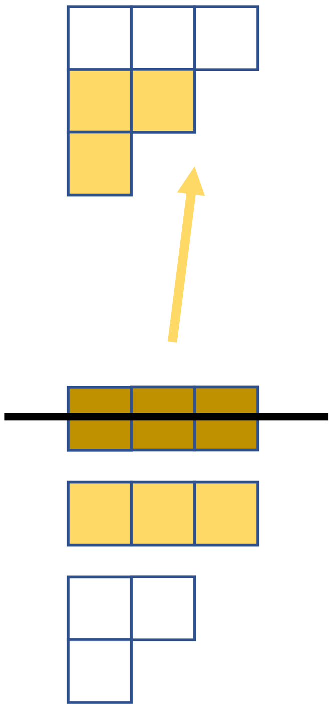


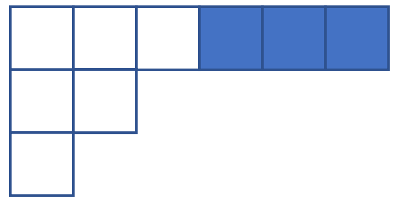
$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 \chi^{(3,2,1)}_{(3,2,1)} + (-1)^1 \chi^{(5)}_{(3,2,1)}$$

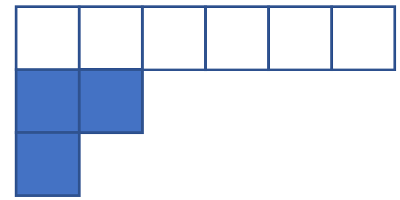


Example

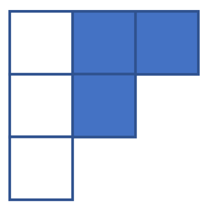
Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3\,\boxed{3}\,2,1)$





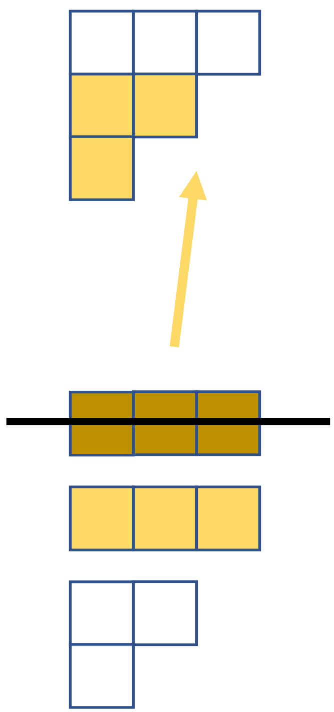


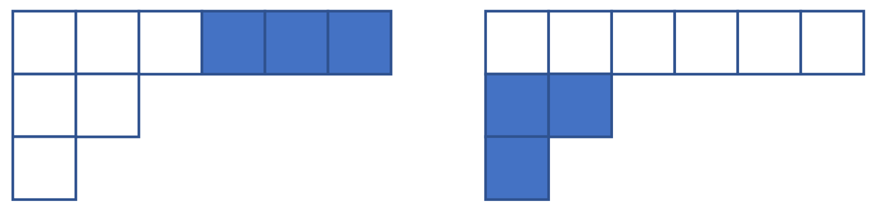
$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 \chi^{(3,2,1)}_{(3,2,1)} + (-1)^1 \chi^{(5)}_{(3,2,1)}$$



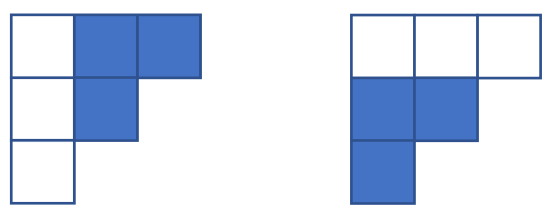
Example

Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3\,\boxed{3}\,2,1)$



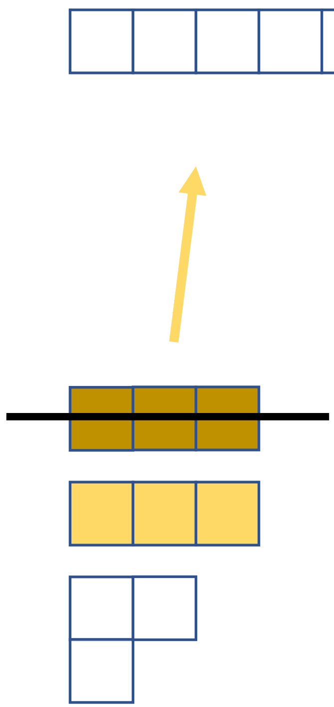


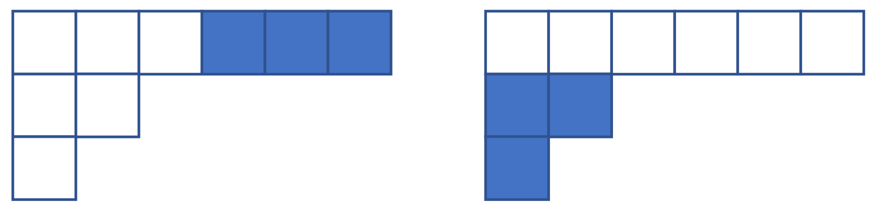
$$\chi^{(6,2,1)}_{(3,3,2,1)} \quad = \quad (-1)^0 \chi^{(3,2,1)}_{(3,2,1)} \quad + \quad (-1)^1 \chi^{(5)}_{(3,2,1)}$$



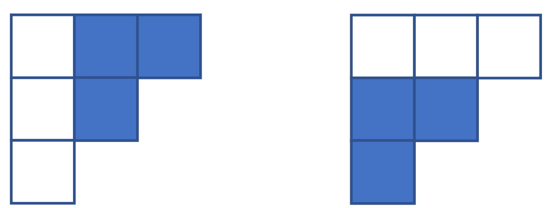
Example

Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3\,\boxed{3}\,2,1)$



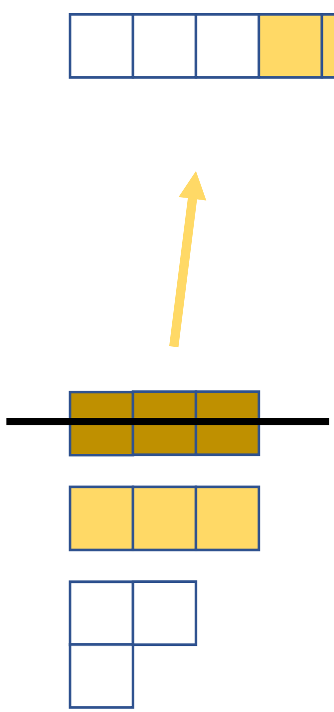


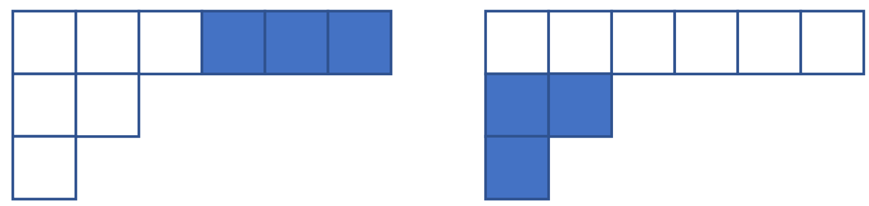
$$\chi^{(6,2,1)}_{(3,3,2,1)} \quad = \quad (-1)^0 \chi^{(3,2,1)}_{(3,2,1)} \quad + \quad (-1)^1 \chi^{(5)}_{(3,2,1)}$$



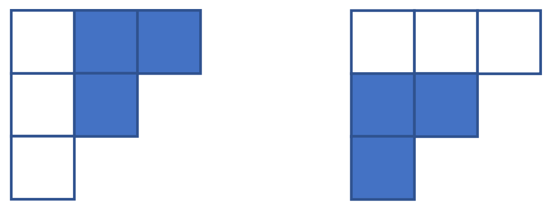
Example

Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3\,\boxed{3}\,2,1)$



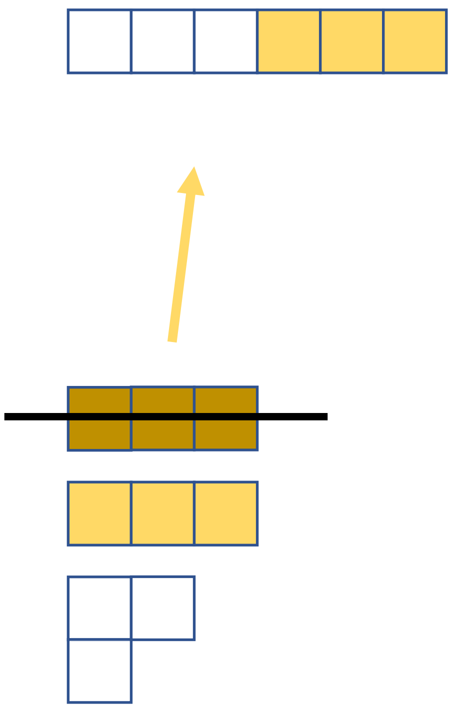


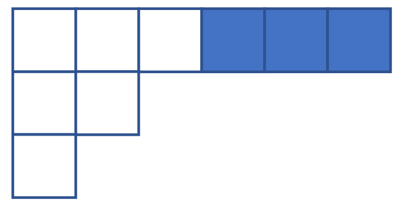
$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 \chi^{(3,2,1)}_{(3,2,1)} + (-1)^1 \chi^{(5)}_{(3,2,1)}$$

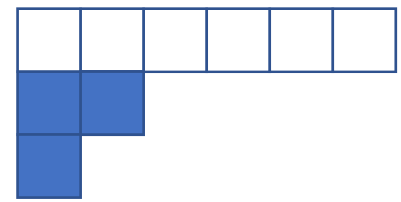


Example

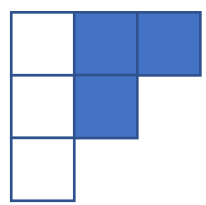
Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3\ \boxed{3}\ 2,1)$

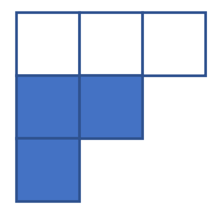






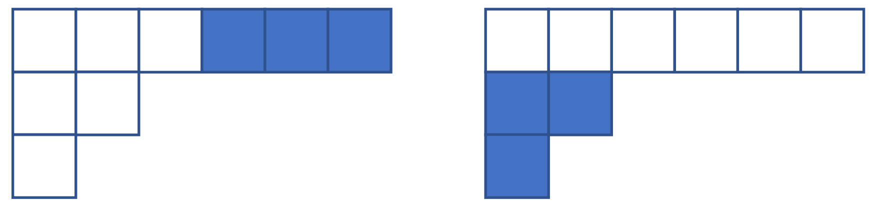
$$\chi^{(6,2,1)}_{(3,3,2,1)} \quad = \quad (-1)^0\chi^{(3,2,1)}_{(3,2,1)} \quad + \quad (-1)^1\chi^{(5)}_{(3,2,1)}$$



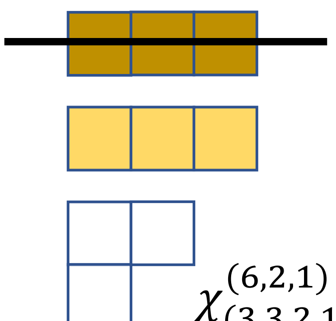


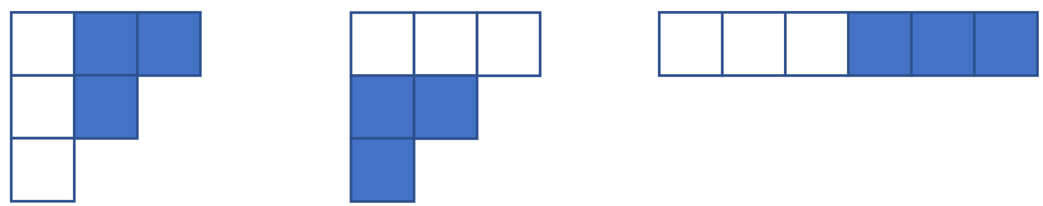
Example

Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3\,\boxed{3}\,2,1)$



$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 \chi^{(3,2,1)}_{(3,2,1)} + (-1)^1 \chi^{(5)}_{(3,2,1)}$$

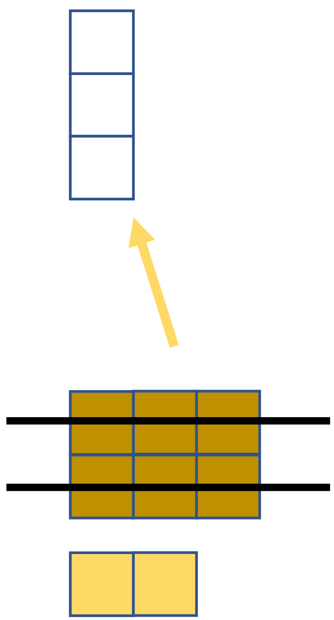


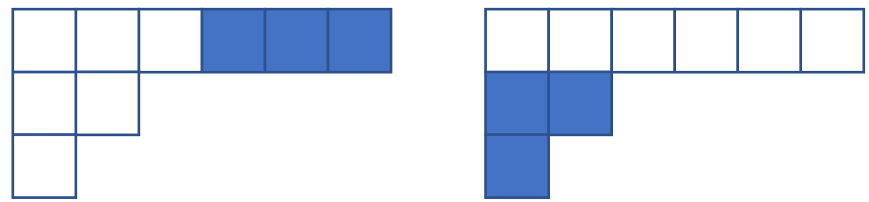


$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 [\, (-1)^1 \chi^{(1,1,1)}_{(2,1)} + (-1)^1 \chi^{(3)}_{(2,1)} \,] + (-1)^1 [(-1)^0 \chi^{(3)}_{(2,1)}]$$

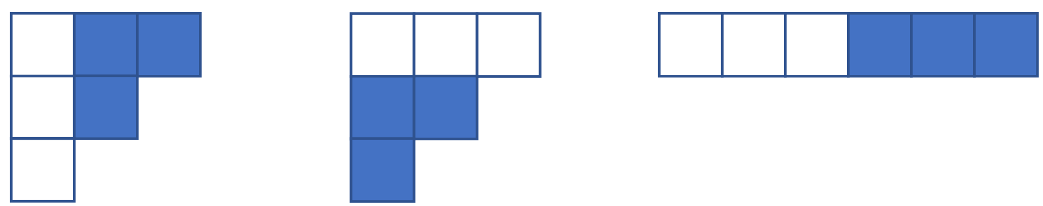
Example

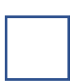
Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3,3\,\boxed{2}\,1)$





$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0\chi^{(3,2,1)}_{(3,2,1)} + (-1)^1\chi^{(5)}_{(3,2,1)}$$

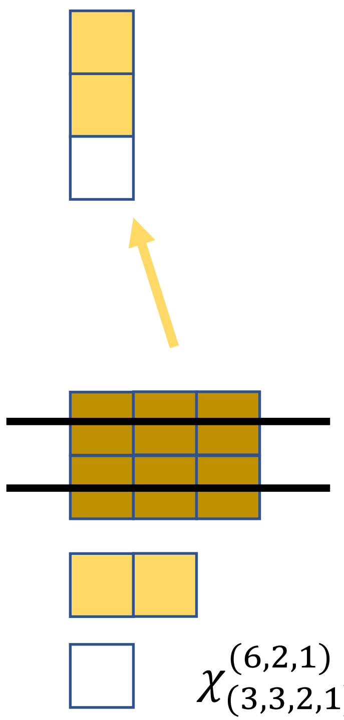


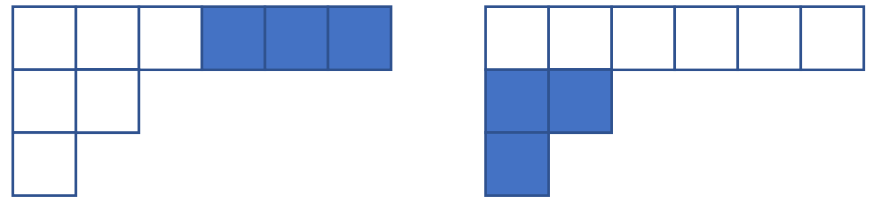


$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0[\,(-1)^1\chi^{(1,1,1)}_{(2,1)} + (-1)^1\chi^{(3)}_{(2,1)}\,] + (-1)^1[(-1)^0\chi^{(3)}_{(2,1)}]$$

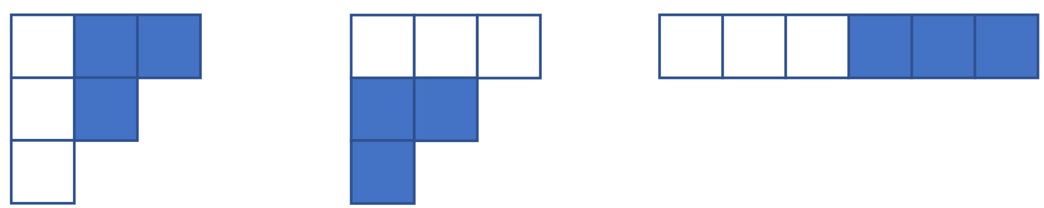
Example

Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3,3\boxed{2}1)$





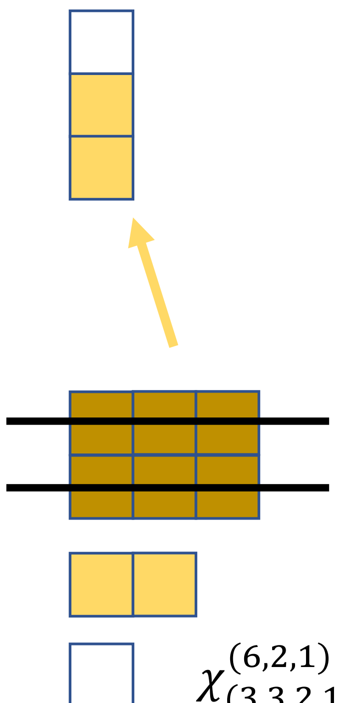
$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 \chi^{(3,2,1)}_{(3,2,1)} + (-1)^1 \chi^{(5)}_{(3,2,1)}$$

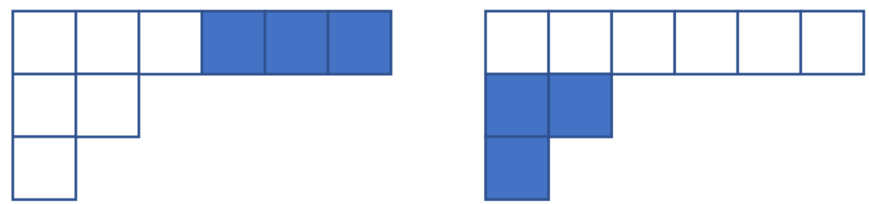


$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 [\, (-1)^1 \chi^{(1,1,1)}_{(2,1)} + (-1)^1 \chi^{(3)}_{(2,1)} \,] + (-1)^1 [(-1)^0 \chi^{(3)}_{(2,1)}]$$

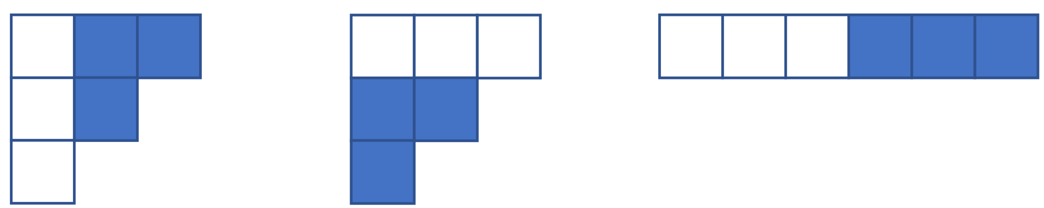
Example

Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3,3\,\boxed{2}\,1)$





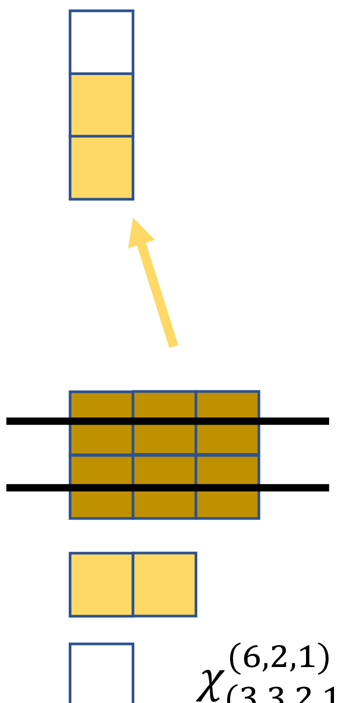
$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0\chi^{(3,2,1)}_{(3,2,1)} + (-1)^1\chi^{(5)}_{(3,2,1)}$$

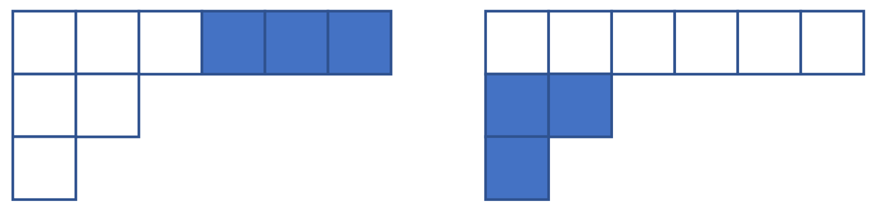


$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0[\,(-1)^1\chi^{(1,1,1)}_{(2,1)} + (-1)^1\chi^{(3)}_{(2,1)}\,] + (-1)^1[(-1)^0\chi^{(3)}_{(2,1)}]$$

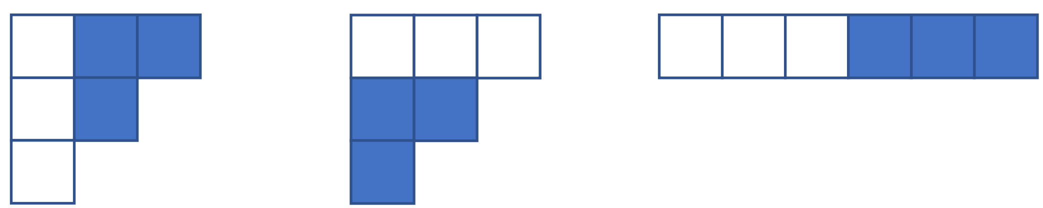
Example

Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3,3\,\boxed{2}\,1)$

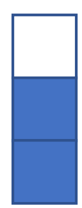




$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0\chi^{(3,2,1)}_{(3,2,1)} + (-1)^1\chi^{(5)}_{(3,2,1)}$$

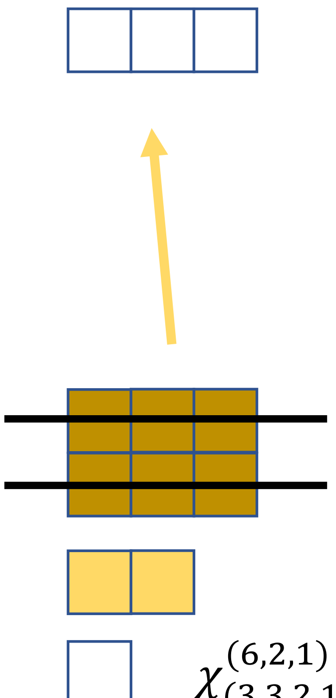


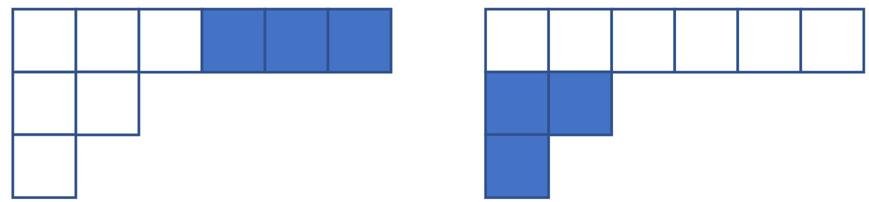
$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0[\,(-1)^1\chi^{(1,1,1)}_{(2,1)} + (-1)^1\chi^{(3)}_{(2,1)}\,] + (-1)^1[(-1)^0\chi^{(3)}_{(2,1)}]$$



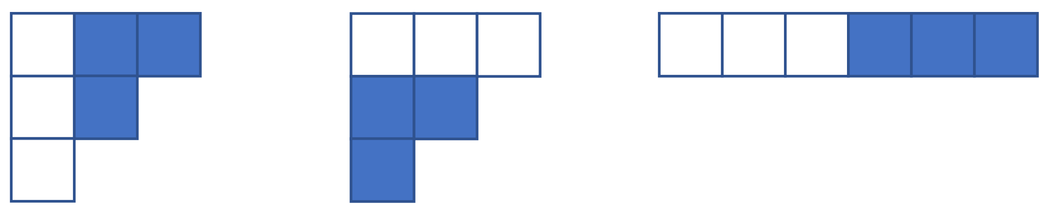
Example

Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3,3\,\boxed{2}\,1)$

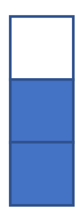




$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 \chi^{(3,2,1)}_{(3,2,1)} + (-1)^1 \chi^{(5)}_{(3,2,1)}$$

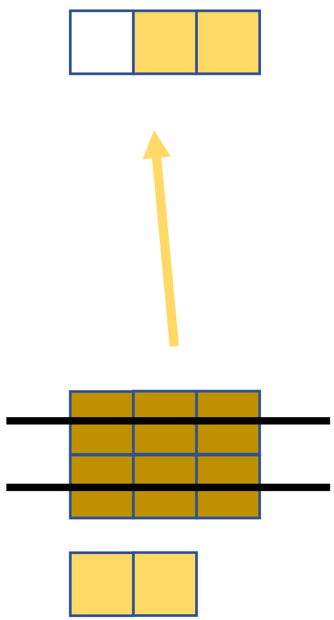


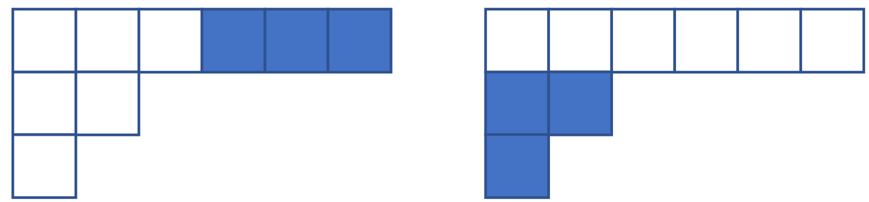
$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 [\, (-1)^1 \chi^{(1,1,1)}_{(2,1)} + (-1)^1 \chi^{(3)}_{(2,1)} \,] + (-1)^1 [(-1)^0 \chi^{(3)}_{(2,1)}]$$



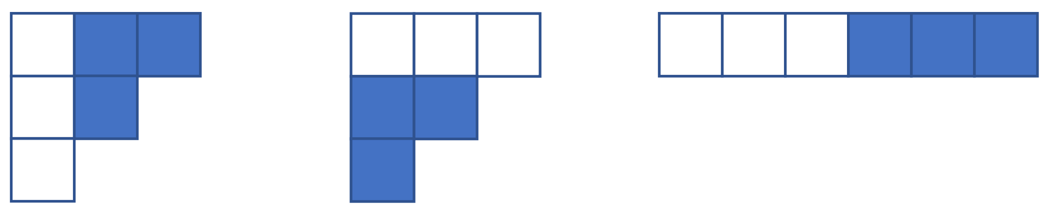
Example

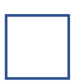
Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3,3\,\boxed{2}\,1)$



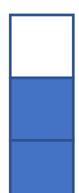


$$\chi^{(6,2,1)}_{(3,3,2,1)} \quad = \quad (-1)^0 \chi^{(3,2,1)}_{(3,2,1)} \quad + \quad (-1)^1 \chi^{(5)}_{(3,2,1)}$$



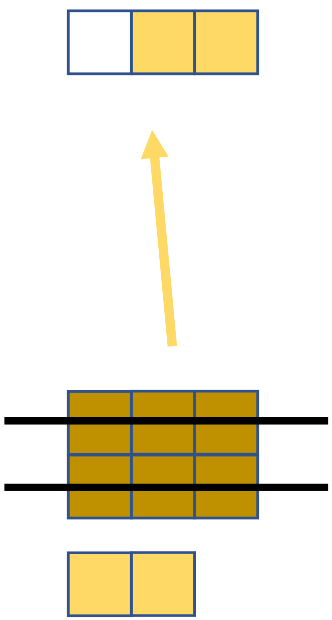


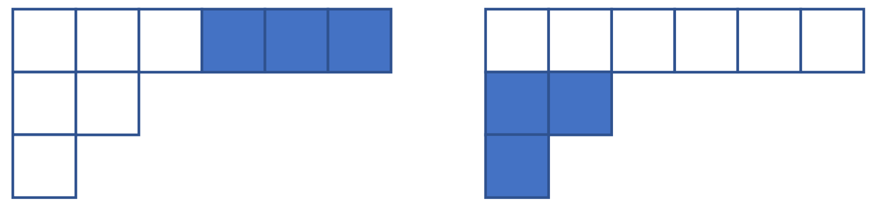
$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 [\, (-1)^1 \chi^{(1,1,1)}_{(2,1)} + (-1)^1 \chi^{(3)}_{(2,1)} \,] \; + \; (-1)^1 [(-1)^0 \chi^{(3)}_{(2,1)}]$$



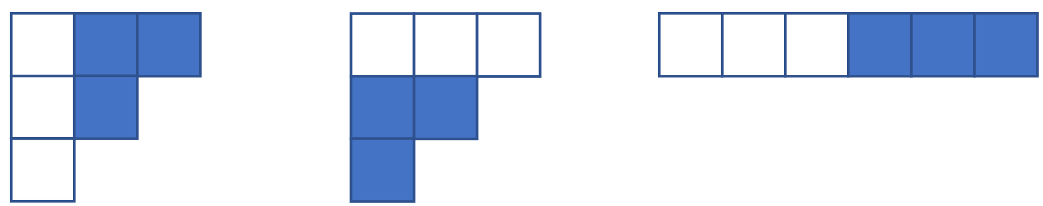
Example

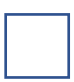
Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3,3\,\boxed{2}\,1)$



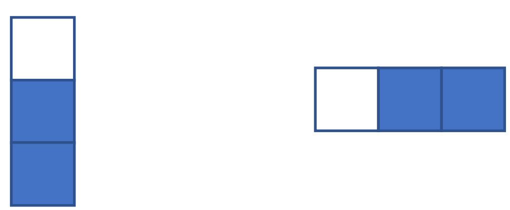


$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 \chi^{(3,2,1)}_{(3,2,1)} + (-1)^1 \chi^{(5)}_{(3,2,1)}$$



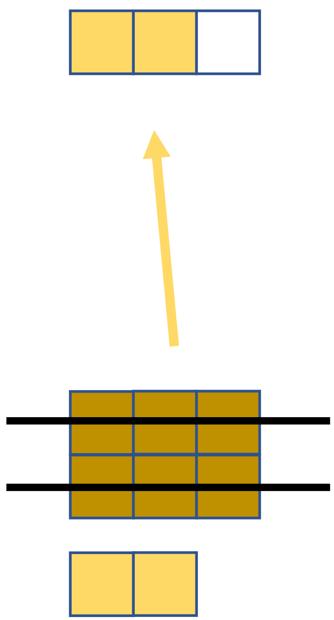


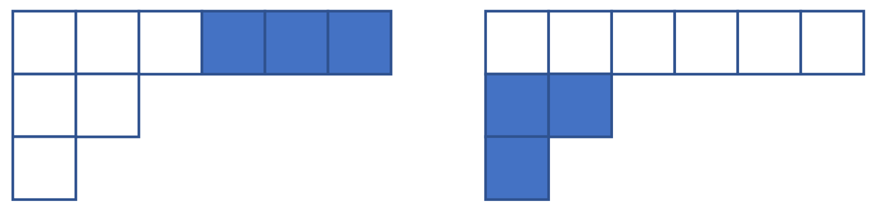
$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 [\, (-1)^1 \chi^{(1,1,1)}_{(2,1)} + (-1)^1 \chi^{(3)}_{(2,1)} \,] + (-1)^1 [(-1)^0 \chi^{(3)}_{(2,1)}]$$



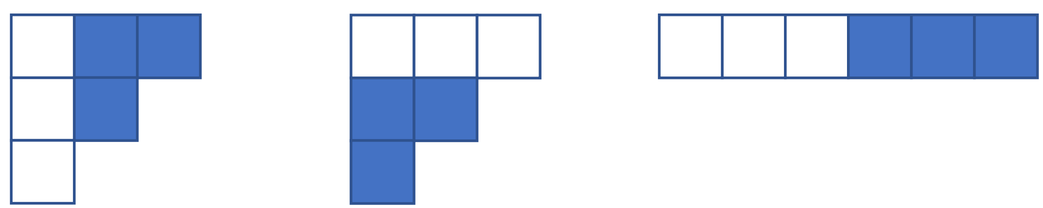
Example

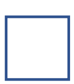
Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3,3\boxed{2}1)$





$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0\chi^{(3,2,1)}_{(3,2,1)} + (-1)^1\chi^{(5)}_{(3,2,1)}$$

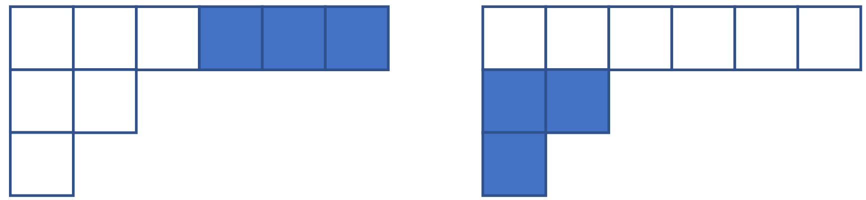




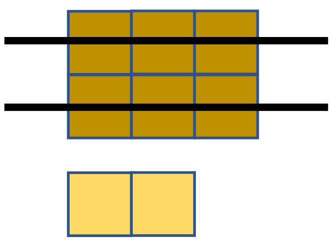
$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0[\,(-1)^1\chi^{(1,1,1)}_{(2,1)} + (-1)^1\chi^{(3)}_{(2,1)}\,] + (-1)^1[(-1)^0\chi^{(3)}_{(2,1)}]$$

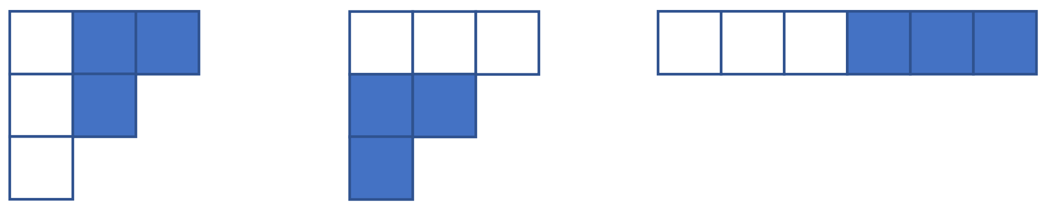
Example

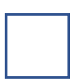
Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3,3\,\boxed{2}\,1)$



$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 \chi^{(3,2,1)}_{(3,2,1)} + (-1)^1 \chi^{(5)}_{(3,2,1)}$$



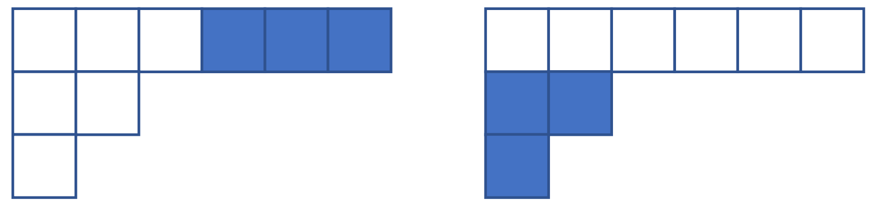




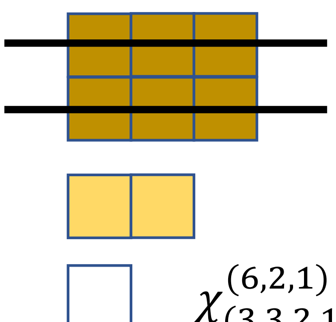
$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 [\, (-1)^1 \chi^{(1,1,1)}_{(2,1)} + (-1)^1 \chi^{(3)}_{(2,1)} \,] + (-1)^1 [(-1)^0 \chi^{(3)}_{(2,1)}]$$

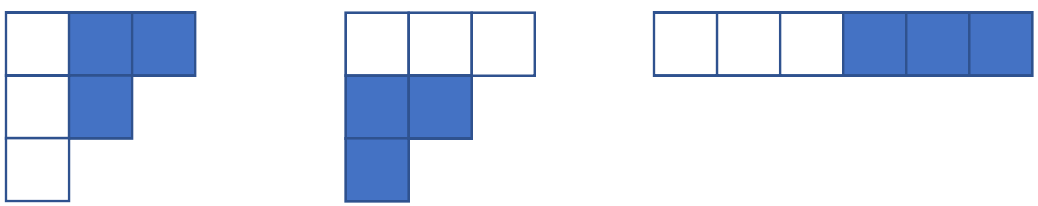
Example

Recursive Version: (size: 9) $\lambda = (6,2,1)$, $\rho = (3,3\,\boxed{2}\,1)$

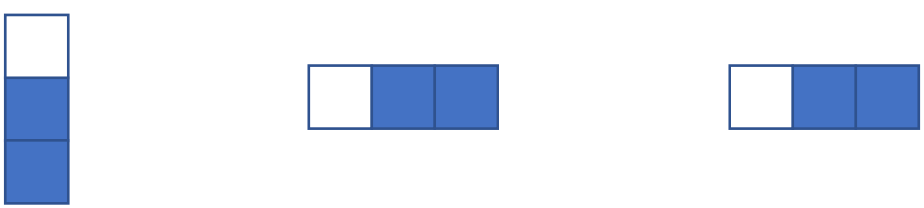


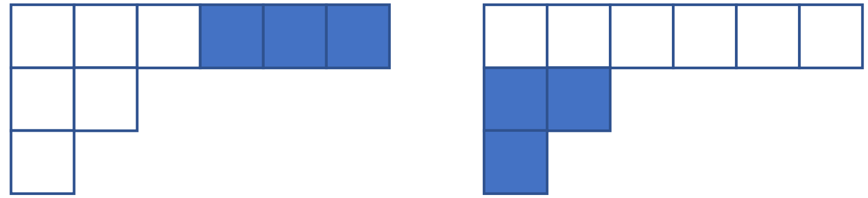
$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 \chi^{(3,2,1)}_{(3,2,1)} + (-1)^1 \chi^{(5)}_{(3,2,1)}$$



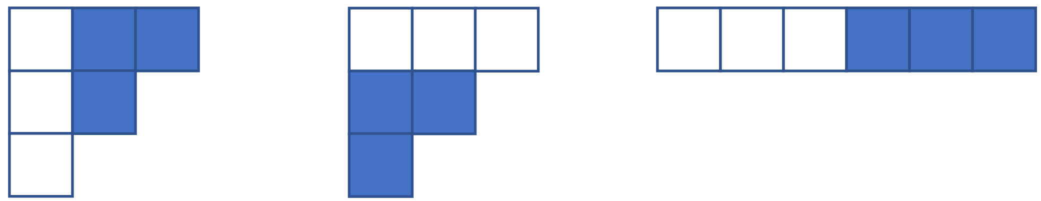


$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 [\, (-1)^1 \chi^{(1,1,1)}_{(2,1)} + (-1)^1 \chi^{(3)}_{(2,1)} \,] + (-1)^1 [(-1)^0 \chi^{(3)}_{(2,1)}]$$

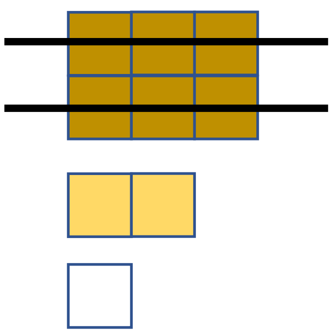


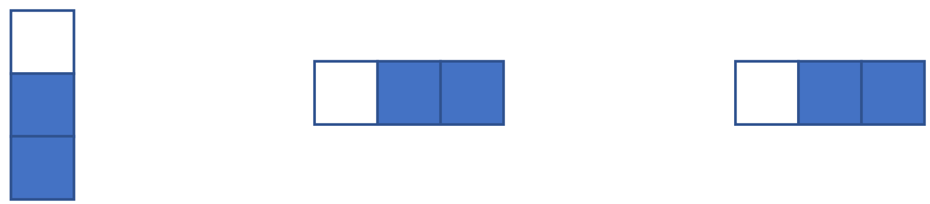


$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0\chi^{(3,2,1)}_{(3,2,1)} + (-1)^1\chi^{(5)}_{(3,2,1)}$$

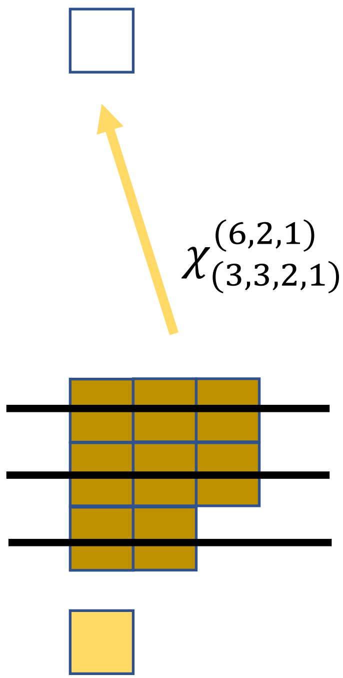


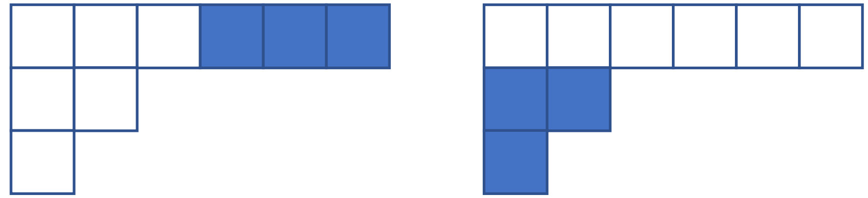
$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0[\,(-1)^1\chi^{(1,1,1)}_{(2,1)} + (-1)^1\chi^{(3)}_{(2,1)}\,] + (-1)^1[(-1)^0\chi^{(3)}_{(2,1)}]$$



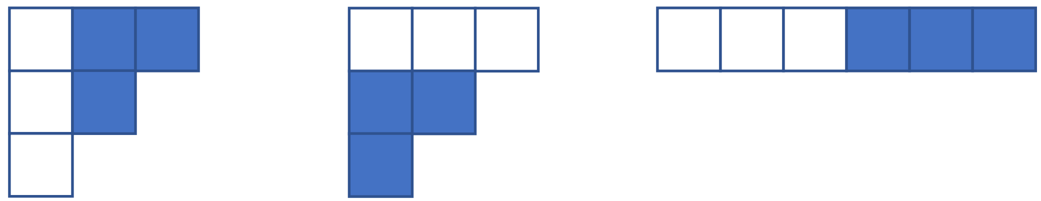


$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0\{\,(-1)^1[\,(-1)^1\chi^{(1)}_{(1)}] + (-1)^1[\,(-1)^0\chi^{(1)}_{(1)}]\,\}$$
$$+\ (-1)^1\{(-1)^0[\,(-1)^0\chi^{(1)}_{(1)}]\,\}$$

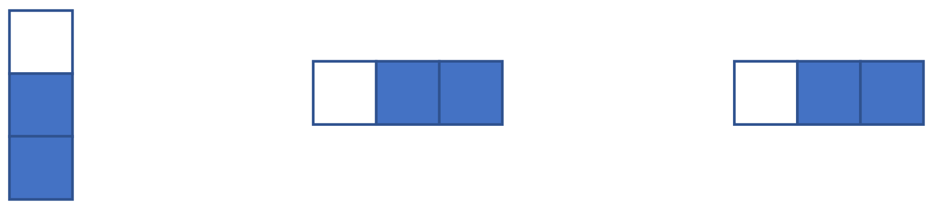




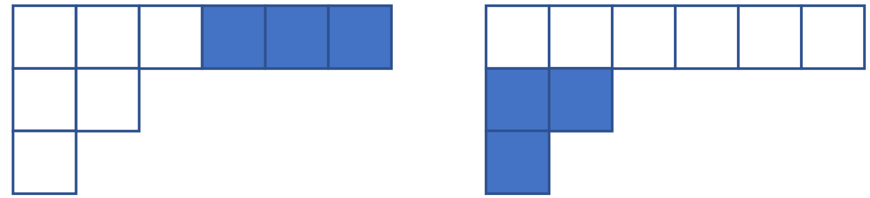
$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 \chi^{(3,2,1)}_{(3,2,1)} + (-1)^1 \chi^{(5)}_{(3,2,1)}$$



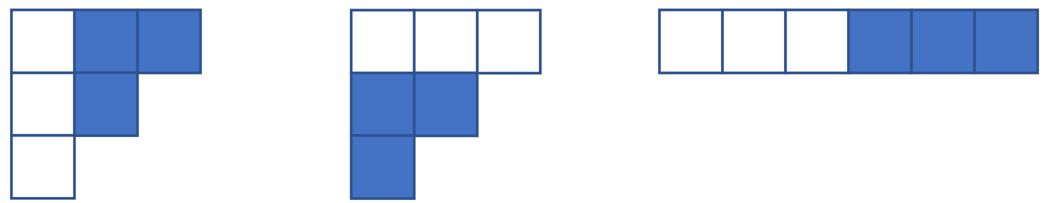
$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 [\, (-1)^1 \chi^{(1,1,1)}_{(2,1)} + (-1)^1 \chi^{(3)}_{(2,1)} \,] + (-1)^1 [(-1)^0 \chi^{(3)}_{(2,1)}]$$

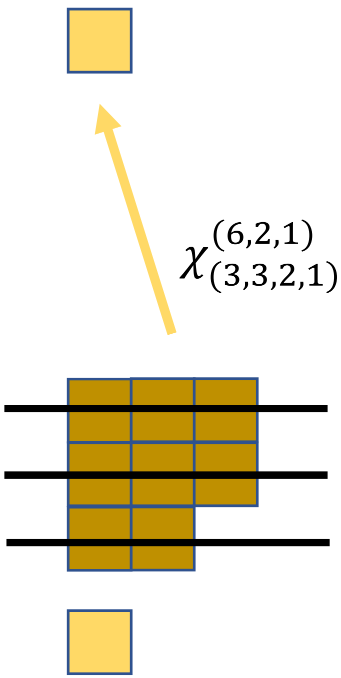


$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 \{\, (-1)^1 [\, (-1)^1 \chi^{(1)}_{(1)}] + (-1)^1 [\, (-1)^0 \chi^{(1)}_{(1)}] \,\} + (-1)^1 \{(-1)^0 [\, (-1)^0 \chi^{(1)}_{(1)}] \,\}$$

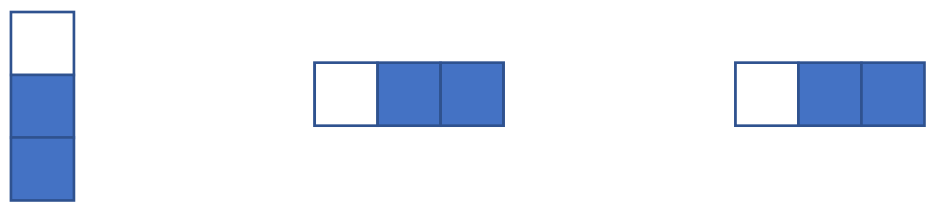


$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 \chi^{(3,2,1)}_{(3,2,1)} + (-1)^1 \chi^{(5)}_{(3,2,1)}$$

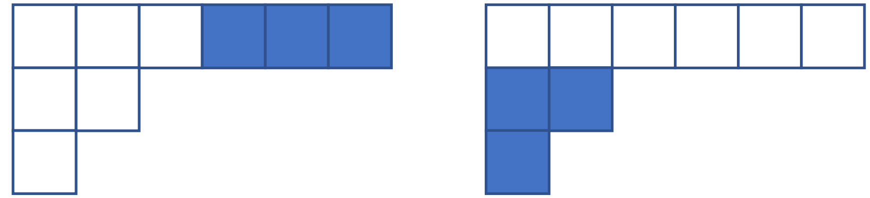




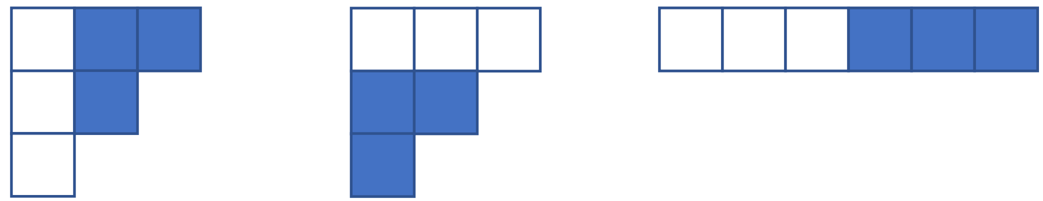
$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 [\, (-1)^1 \chi^{(1,1,1)}_{(2,1)} + (-1)^1 \chi^{(3)}_{(2,1)} \,] + (-1)^1 [(-1)^0 \chi^{(3)}_{(2,1)}]$$

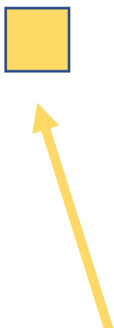


$$\begin{aligned}\chi^{(6,2,1)}_{(3,3,2,1)} = {} & (-1)^0 \{\, (-1)^1 [\, (-1)^1 \chi^{(1)}_{(1)}] + (-1)^1 [\, (-1)^0 \chi^{(1)}_{(1)}] \,\} \\ & + (-1)^1 \{(-1)^0 [\, (-1)^0 \chi^{(1)}_{(1)}] \,\}\end{aligned}$$

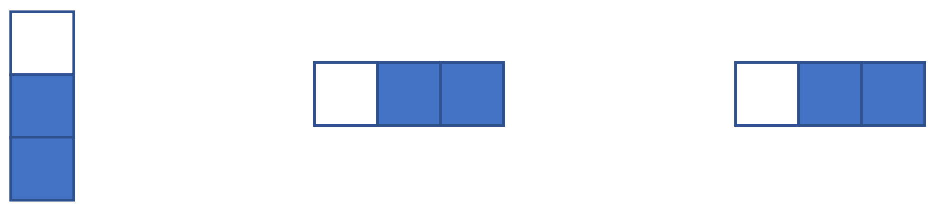


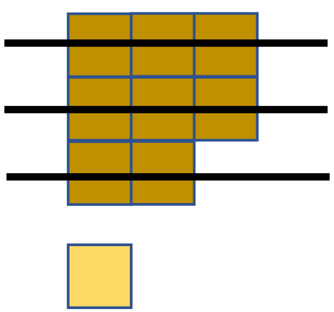
$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0\chi^{(3,2,1)}_{(3,2,1)} + (-1)^1\chi^{(5)}_{(3,2,1)}$$



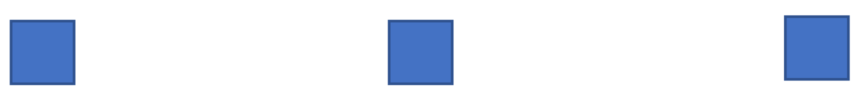


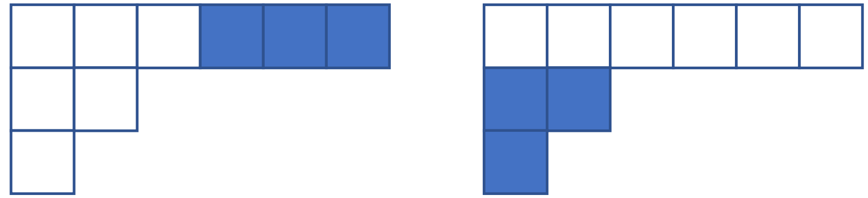
$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0[\,(-1)^1\chi^{(1,1,1)}_{(2,1)} + (-1)^1\chi^{(3)}_{(2,1)}\,] + (-1)^1[(-1)^0\chi^{(3)}_{(2,1)}]$$



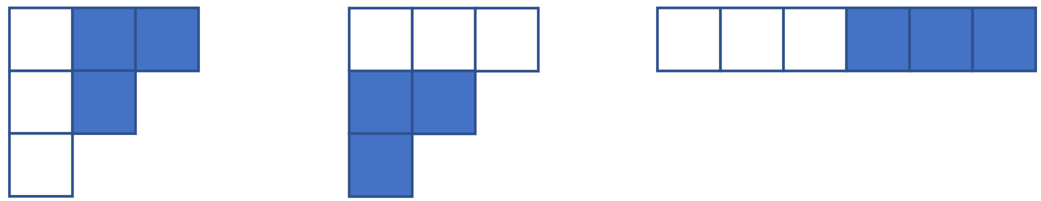


$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0\{\,(-1)^1[\,(-1)^1\chi^{(1)}_{(1)}] + (-1)^1[\,(-1)^0\chi^{(1)}_{(1)}]\,\}$$
$$+\ (-1)^1\{(-1)^0[\,(-1)^0\chi^{(1)}_{(1)}]\,\}$$

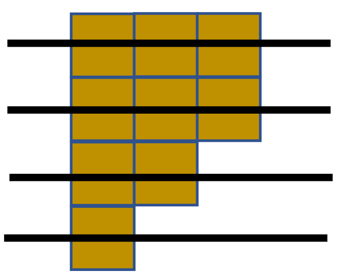


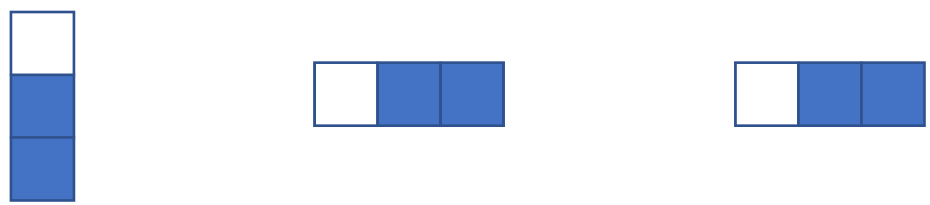


$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 \chi^{(3,2,1)}_{(3,2,1)} + (-1)^1 \chi^{(5)}_{(3,2,1)}$$

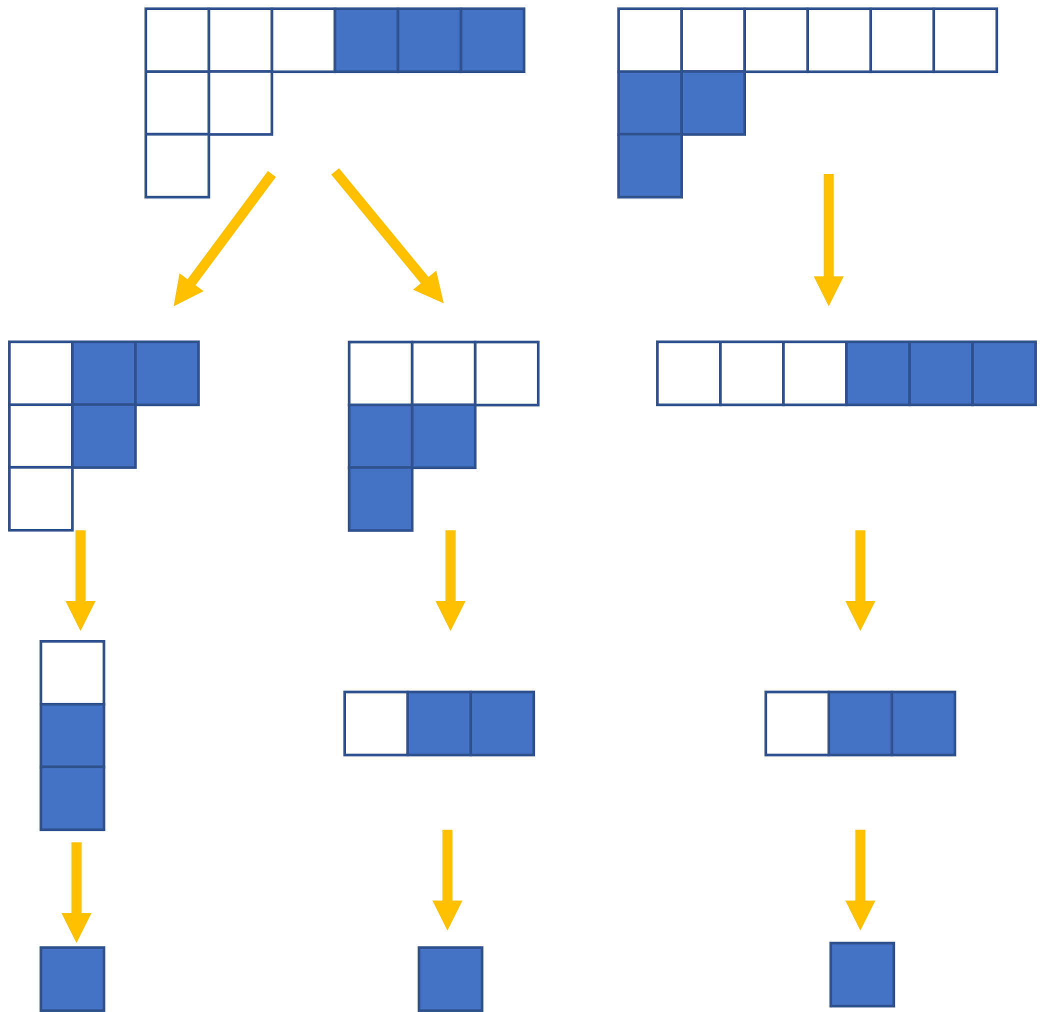


$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 [\, (-1)^1 \chi^{(1,1,1)}_{(2,1)} + (-1)^1 \chi^{(3)}_{(2,1)} \,] + (-1)^1 [(-1)^0 \chi^{(3)}_{(2,1)}]$$





$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 \{\, (-1)^1 [\, (-1)^1 \chi^{(1)}_{(1)}] + (-1)^1 [\, (-1)^0 \chi^{(1)}_{(1)}] \,\} + (-1)^1 \{(-1)^0 [\, (-1)^0 \chi^{(1)}_{(1)}] \,\}$$



$$\chi^{(6,2,1)}_{(3,3,2,1)} = (-1)^0 \{ (-1)^1 [(-1)^1 \; 1] + (-1)^1 [(-1)^0 \; 1] \} + (-1)^1 \{ (-1)^0 [(-1)^0 1] \}$$

$$= 1 \times [(-1)(-1) + (-1)1] + (-1)(1 \times 1)$$

$$= -1$$

Algorithm

Algorithm 1 Non-Recursive Version

```
function CALCULATE(λ, ρ)
    Sum ← 0
    for Tableau ∈ BST(λ, ρ) do
        if ht(Tableau) is even then        ▷ equivalent to (−1)^{ht(·)} = 1
            Sum ← Sum + 1
        else
            Sum ← Sum − 1
        end if
    end for
    return Sum
end function
```

Algorithm

Algorithm 2 Recursive Version

```
function CALCULATE(λ, ρ)
    if λ = () & ρ = () then
        return 1
    end if
    CacheValue ← CACHE(λ, ρ)                ▷ use cache to avoid heavy computation
    if CacheValue ≠ null then
        return CacheValue
    end if
    Length ← ρ_1
    SubWeightPartition ← ρ_{i>1}
    Sum ← 0
    for BorderStrip ∈ BS(λ, Length) do
        RestTableau ← λ\BorderStrip
        if ht(BorderStrip) is even then
            Sum ← Sum+CALCULATE(RestTableau, SubWeightPartition)
        else
            Sum ← Sum−CALCULATE(RestTableau, SubWeightPartition)
        end if
    end for
    CACHE(λ, ρ) ← Sum
    return Sum
end function
```

Optimization

Optimization

Problem

Partition Map

- Problem

We developed several versions…

Version	Language	Problem
1	Python	Slow (when n>25)
2	Java	java.lang.OutOfMemoryError (when n>30)
3	Golang	Hashmap does not work well
4 # final	Golang	-

Optimization

Problem

Partition Map

- Problem

We developed several versions…

Version	Language	Problem
1	Python	Slow (when n>25)
2	Java	java.lang.OutOfMemoryError (when n>30)
3	Golang	Hashmap does not work well
4 # final	Golang	-

Since we can not use partition directly as the **key** of the **map** in Golang, and using **string** as the key is very inefficient even though we can easily turn a partition into a string.

We then need to construct an injection or bijection from partition to integer.

Optimization

Problem

Partition Map

- Partition Map

Definition. (Ordering of Partition) For the same size partitions λ, ρ, if the number of rows of λ is less than or equal to ρ, we say partition $\lambda > \rho$ if the first different row (denote the index by j) satisfy $\lambda_j > \rho_j$.

Then for partitions $\{\mu^{(i)}\}$ of n, we have an increasing sequence $\{\mu^{(i)} \mid \mu^{(i+1)} > \mu^{(i)}\}$ by the compare method above, we say i is the index of such sequence, where i starts from 0.

Example. (n=4)

$$(1,1,1,1) < (2,1,1) < (2,2) < (3,1) < (4)$$

The index of $(2,1,1)$ is 1.

Optimization

Problem

Partition Map

- Partition Map

A natural idea: enumeration

Definition. Let $\{\lambda\}$ be the partitions of n, then we define $P(n,k)$ as the number of partitions of n whose first part is k $(\text{i.e.}, \lambda_1 = k)$, and $R(n,k)$ the number of partitions of n with all rows $\leq k$ $(\lambda_i \leq k)$.

Then we have the **Index function** $I_n : Partition \rightarrow \mathbb{N}$ of n:

$$I_n(\lambda) = \sum_{\substack{i=0,1,2,\ldots,n \\ m=n}} R(m-\lambda_i, \lambda_{i+1}-1)$$

where k is the number of rows of a given partition λ, and define $\lambda_0 = 0$.

The state stored in bytes of computer memory is

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4"></td><td colspan="22">$I_n(\lambda)$</td><td colspan="6">Size n</td></tr>
</table>

Results

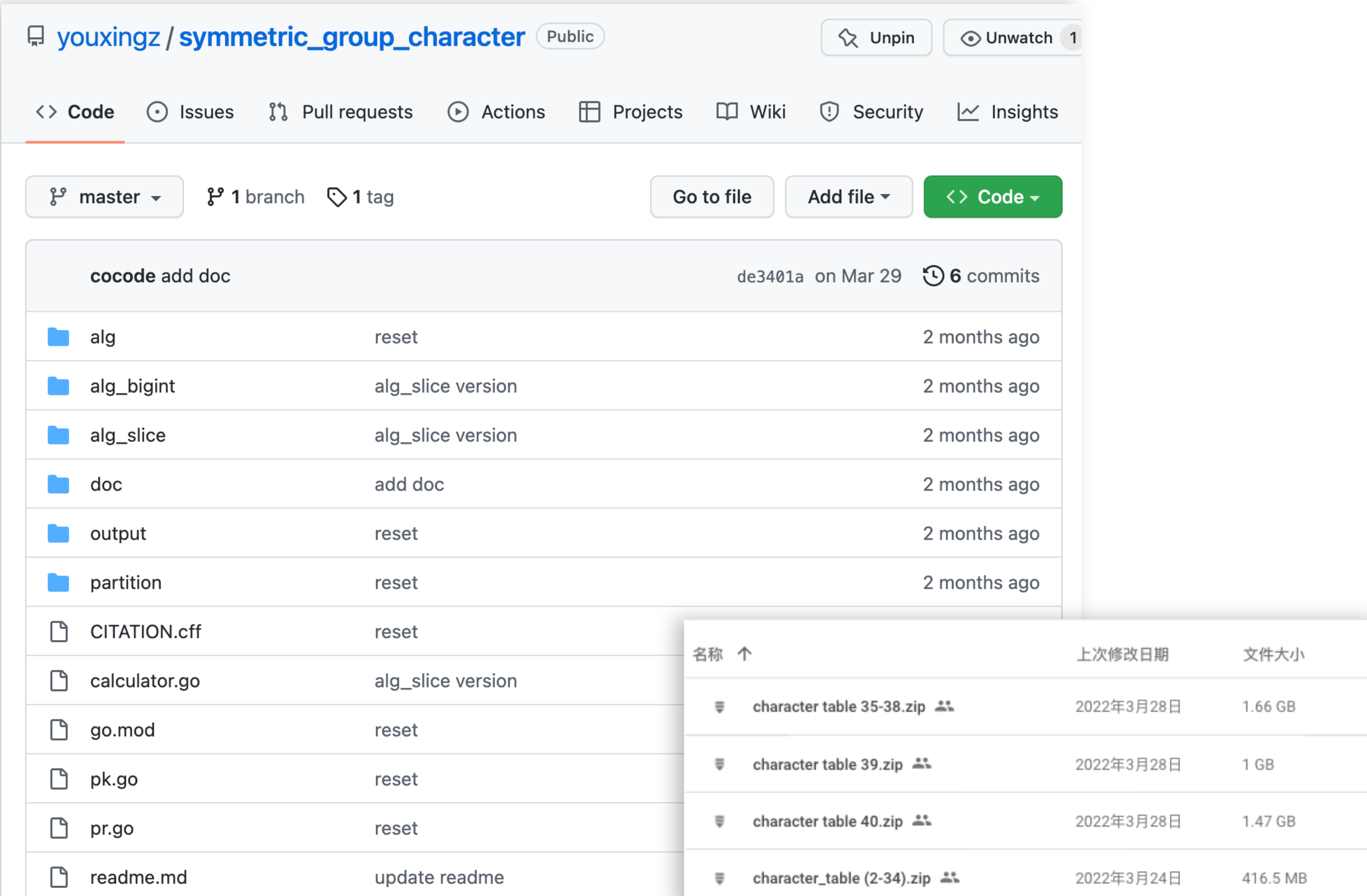
Test Results

n	number of partitions	spend time (non-recursive version, ms)	spend time (non-recursive version, multiple threads, ms)	spend time (recursive version, ms)	spend time (Sagemath)
5	7	0	0	0	13
10	42	1	0	1	101
15	176	17	4	21	1342
20	627	151	46	270	16497
25	1958	1524	434	2886	162954
30	5604	12254	3294	24189	1615291
35	14883	92343	25044	216933	-

(Test Device: MacBook Pro 13, 2.4 GHz 4-Core Intel Core i5, 8 GB 2133 MHz LPDDR3)

Up to 400 times faster than Sagemath!

Codes & Tables



Next

- Use GPU to speed up (50 times faster than current)
- Merge code into Sagemath

Refrence

[1] Joel Gibson. *Enumerating Partitions*. https://www.jgibson.id.au/articles/characters/#enumerating-partitions. 2021.

[2] William Fulton, Joe Harris. *Representation Theory. A First Course*. Springer Science, 2004.

[3] Youxing Z. *Calculate Character Table of S_n*. `https://github.com/youxingz/symmetric_group_character`. Source Code. Version 1.0. 2022.

Thank you!

Q&A