k-Schur functions

Henry Twiss

University of Minnesota

July 2020

Outline

Back of Pocket

Back of Pocket

k-Schur functions $s_\lambda^{(k)}$ are sums of Schur functions over a special set of tableaux $\mathbb{A}_\lambda^{(k)}$ with coefficients in $\mathbb{Z}[t]$.

Outline

Conventions

Conventions

We assume familiarity with the definition of partitions and Young tableaux.

Conventions

We assume familiarity with the definition of partitions and Young tableaux. Let

$$\mathrm{dg}(\lambda) = \{(i, j) \mid 1 \leq i \leq \ell(\lambda) \text{ and } 1 \leq j \leq \lambda_i\}.$$

Conventions

We assume familiarity with the definition of partitions and Young tableaux. Let

$$\mathrm{dg}(\lambda) = \{(i,j) \mid 1 \leq i \leq \ell(\lambda) \text{ and } 1 \leq j \leq \lambda_i\}.$$

We give partitions the ordering $\lambda \subseteq \mu$ if $\mathrm{dg}(\lambda) \subset \mathrm{dg}(\mu)$.

Conventions

We assume familiarity with the definition of partitions and Young tableaux. Let

$$\mathrm{dg}(\lambda) = \{(i, j) \mid 1 \le i \le \ell(\lambda) \text{ and } 1 \le j \le \lambda_i\}.$$

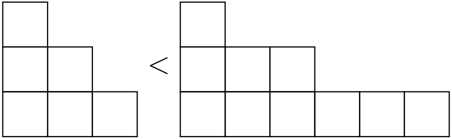
We give partitions the ordering $\lambda \subseteq \mu$ if $\mathrm{dg}(\lambda) \subset \mathrm{dg}(\mu)$. If $\lambda = (3, 2, 1)$ and $\mu = (6, 3, 1)$, then $\lambda < \mu$.

Conventions

We assume familiarity with the definition of partitions and Young tableaux. Let

$$\mathrm{dg}(\lambda) = \{(i,j) \mid 1 \leq i \leq \ell(\lambda) \text{ and } 1 \leq j \leq \lambda_i\}.$$

We give partitions the ordering $\lambda \subseteq \mu$ if $\mathrm{dg}(\lambda) \subset \mathrm{dg}(\mu)$. If $\lambda = (3,2,1)$ and $\mu = (6,3,1)$, then $\lambda < \mu$.



Conventions

Given a Young tableau T, the reading word w of T is read from top to bottom and left to right.

Conventions

Given a Young tableau T, the reading word w of T is read from top to bottom and left to right.

If T is

5		
4	4	
1	2	2

Conventions

Given a Young tableau T, the reading word w of T is read from top to bottom and left to right.
If T is

5		
4	4	
1	2	2

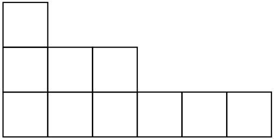
then $w = 544122$.

Conventions

We call λ k-bounded if $\lambda_1 \leq k$.

Conventions

We call λ k-bounded if $\lambda_1 \leq k$. The partition



is 6-bounded and not 5-bounded.

Strips

Strips

Definition 1.1

If $\lambda \subseteq \mu$, μ/λ is a called a skew partition and its cells are $\mathrm{dg}(\mu) - \mathrm{dg}(\lambda)$.

Strips

Definition 1.1

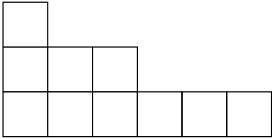
If $\lambda \subseteq \mu$, μ/λ is a called a skew partition and its cells are $\mathrm{dg}(\mu) - \mathrm{dg}(\lambda)$. μ/λ is a horizontal (vertical) strip if there is at most one cell in each column (row).

Strips

Definition 1.1

If $\lambda \subseteq \mu$, μ/λ is a called a skew partition and its cells are $\mathrm{dg}(\mu) - \mathrm{dg}(\lambda)$. μ/λ is a horizontal (vertical) strip if there is at most one cell in each column (row).

If $\mu = (6, 3, 1)$ and $\lambda = (3, 2, 1)$,

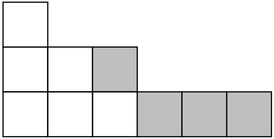


Strips

Definition 1.1

If $\lambda \subseteq \mu$, μ/λ is a called a skew partition and its cells are $\mathrm{dg}(\mu) - \mathrm{dg}(\lambda)$. μ/λ is a horizontal (vertical) strip if there is at most one cell in each column (row).

If $\mu = (6, 3, 1)$ and $\lambda = (3, 2, 1)$, the horizontal strip μ/λ is

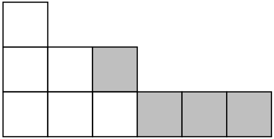


Strips

Definition 1.1

If $\lambda \subseteq \mu$, μ/λ is a called a skew partition and its cells are $\mathrm{dg}(\mu) - \mathrm{dg}(\lambda)$. μ/λ is a horizontal (vertical) strip if there is at most one cell in each column (row).

If $\mu = (6, 3, 1)$ and $\lambda = (3, 2, 1)$, the horizontal strip μ/λ is



Horizontal (vertical) strips **do not** need to lie in a single row (column).

Hooks

Hooks

Definition 1.2

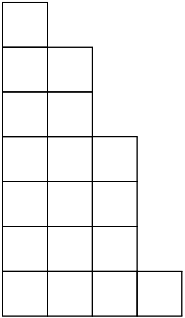
If $(i,j) \in \mathrm{dg}(\lambda)$, then the hook length $\mathrm{hook}_\lambda(i,j)$ is the number of cells above and to the right of (i,j) including (i,j).

Hooks

Definition 1.2

If $(i, j) \in \mathrm{dg}(\lambda)$, then the hook length $\mathrm{hook}_\lambda(i, j)$ is the number of cells above and to the right of (i, j) including (i, j).

If $\lambda = (4, 3, 3, 3, 2, 2, 1)$,

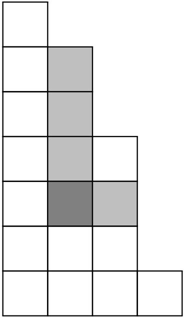


Hooks

Definition 1.2

If $(i, j) \in \mathrm{dg}(\lambda)$, then the hook length $\mathrm{hook}_\lambda(i, j)$ is the number of cells above and to the right of (i, j) including (i, j).

If $\lambda = (4, 3, 3, 3, 2, 2, 1)$, then $\mathrm{hook}_\lambda(3, 2) = 5$



k-Splits

k-Splits

Definition 1.3

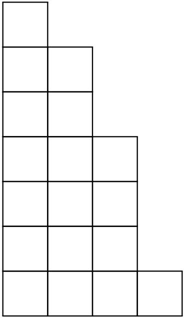
The k-split of λ, denoted $\lambda^{\to k}$, is a list of partitions formed by successively splitting off parts of λ with hook length k, starting with the first part, until that is no longer possible.

k-Splits

Definition 1.3

The k-split of λ, denoted $\lambda^{\to k}$, is a list of partitions formed by successively splitting off parts of λ with hook length k, starting with the first part, until that is no longer possible.

For the previous partition,

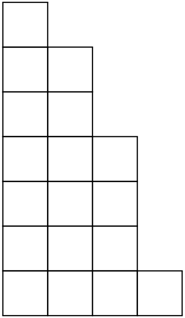


k-Splits

Definition 1.3

The k-split of λ, denoted $\lambda^{\to k}$, is a list of partitions formed by successively splitting off parts of λ with hook length k, starting with the first part, until that is no longer possible.

For the previous partition, $\lambda^{\to 4} = ((4), (3, 3), (3, 2), (2, 1))$.

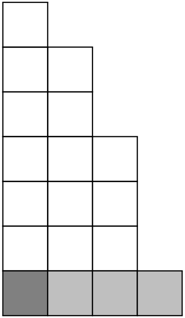


k-Splits

Definition 1.3

The k-split of λ, denoted $\lambda^{\to k}$, is a list of partitions formed by successively splitting off parts of λ with hook length k, starting with the first part, until that is no longer possible.

For the previous partition, $\lambda^{\to 4} = ((4), (3,3), (3,2), (2,1))$.

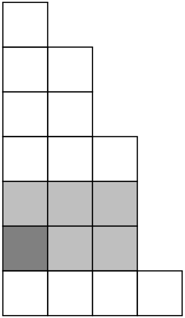


k-Splits

Definition 1.3

The k-split of λ, denoted $\lambda^{\rightarrow k}$, is a list of partitions formed by successively splitting off parts of λ with hook length k, starting with the first part, until that is no longer possible.

For the previous partition, $\lambda^{\rightarrow 4} = ((4), (3,3), (3,2), (2,1))$.

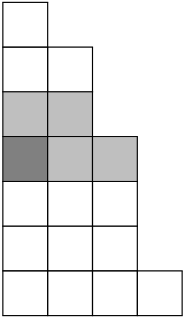


k-Splits

Definition 1.3

The k-split of λ, denoted $\lambda^{\to k}$, is a list of partitions formed by successively splitting off parts of λ with hook length k, starting with the first part, until that is no longer possible.

For the previous partition, $\lambda^{\to 4} = ((4), (3,3), (3,2), (2,1))$.

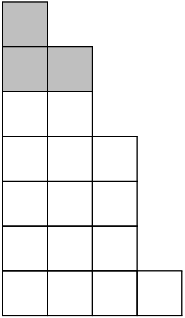


k-Splits

Definition 1.3

The k-split of λ, denoted $\lambda^{\to k}$, is a list of partitions formed by successively splitting off parts of λ with hook length k, starting with the first part, until that is no longer possible.

For the previous partition, $\lambda^{\to 4} = ((4), (3,3), (3,2), (2,1))$.



Outline

Charges

Charges

Definition 2.1

Let T be a semi-standard Young tableau. The charge $\mathrm{charge}(T)$ is computed as follows:

Charges

Definition 2.1

Let T be a semi-standard Young tableau. The charge $\mathrm{charge}(T)$ is computed as follows:

- Starting from the rightmost 1, choose a sequence of i cells containing $1, 2, \ldots, i$ such that the $j+1$th cell is the south-eastern most one above j modulo $\mathrm{shape}(T)$.

Charges

Definition 2.1

Let T be a semi-standard Young tableau. The charge $\mathrm{charge}(T)$ is computed as follows:

- Starting from the rightmost 1, choose a sequence of i cells containing $1, 2, \ldots, i$ such that the $j+1$th cell is the south-eastern most one above j modulo $\mathrm{shape}(T)$.
- Construct an index vector I according to the rule:

$$I_1 = 0 \quad \text{and} \quad I_r = \begin{cases} I_{r-1} + 1 & \text{if } r \text{ is east of } r-1, \\ I_{r-1} & \text{otherwise.} \end{cases}$$

Charges

Definition 2.1

Let T be a semi-standard Young tableau. The charge $\mathrm{charge}(T)$ is computed as follows:

- Starting from the rightmost 1, choose a sequence of i cells containing $1, 2, \ldots, i$ such that the $j+1$th cell is the south-eastern most one above j modulo $\mathrm{shape}(T)$.
- Construct an index vector I according to the rule:

$$I_1 = 0 \quad \text{and} \quad I_r = \begin{cases} I_{r-1} + 1 & \text{if } r \text{ is east of } r-1, \\ I_{r-1} & \text{otherwise.} \end{cases}$$

- Remove the labels in the cells corresponding to this sequence and repeat.

Charges

Definition 2.1

Let T be a semi-standard Young tableau. The charge $\mathrm{charge}(T)$ is computed as follows:

- Starting from the rightmost 1, choose a sequence of i cells containing $1, 2, \ldots, i$ such that the $j+1$th cell is the south-eastern most one above j modulo $\mathrm{shape}(T)$.
- Construct an index vector I according to the rule:

$$I_1 = 0 \quad \text{and} \quad I_r = \begin{cases} I_{r-1} + 1 & \text{if } r \text{ is east of } r-1, \\ I_{r-1} & \text{otherwise.} \end{cases}$$

- Remove the labels in the cells corresponding to this sequence and repeat.
- $\mathrm{charge}(T)$ is the sum of the entries of all I.

Charges

Let T

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

Charges

Let T

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

They have index vectors

Charges

Let T

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

They have index vectors

Charges

Let T

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

They have index vectors

Charges

Let T

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

6					
4	5				
3	4				
2	**2**	3	5		
1	1	**1**	2	3	7

They have index vectors

Charges

Let T

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

They have index vectors

Charges

Let T

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

6					
4	5				
3	4				
2	**2**	3	**5**		
1	1	**1**	2	3	7

They have index vectors

Charges

Let T

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

6					
4	5				
3	4				
2	**2**	3	**5**		
1	1	**1**	2	3	**7**

They have index vectors

Charges

Let T

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

They have index vectors

$$I = [0, 0, 0, 0, 1, 1, 2]$$

Charges

Let T

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

	5				
	4				
2		3			
1	1		2	3	

They have index vectors

$$I = [0, 0, 0, 0, 1, 1, 2]$$

Charges

Let T

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

	5				
	4				
2		3			
1	1		2	3	

They have index vectors

$$I = [0, 0, 0, 0, 1, 1, 2]$$

Charges

Let T

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

	5				
	4				
2		3			
1	1		2	3	

They have index vectors

$$I = [0, 0, 0, 0, 1, 1, 2]$$

Charges

Let T

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

	5				
	4				
2		3			
1	1		2	3	

They have index vectors

$$I = [0, 0, 0, 0, 1, 1, 2]$$

Charges

Let T

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

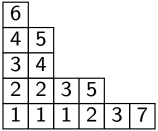
	5				
	4				
2		3			
1	1		2	3	

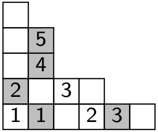
They have index vectors

$$I = [0, 0, 0, 0, 1, 1, 2]$$

Charges

Let T



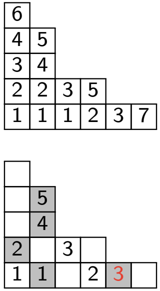


They have index vectors

$$I = [0, 0, 0, 0, 1, 1, 2]$$

Charges

Let T



They have index vectors

$$I = [0, 0, 0, 0, 1, 1, 2] \quad \text{and} \quad I = [0, 0, 1, 1, 1]$$

Charges

Let T

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

		3			
1			2		

They have index vectors

$$I = [0, 0, 0, 0, 1, 1, 2] \quad \text{and} \quad I = [0, 0, 1, 1, 1]$$

Charges

Let T

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

		3			
1			2		

They have index vectors

$$I = [0, 0, 0, 0, 1, 1, 2] \quad \text{and} \quad I = [0, 0, 1, 1, 1]$$

Charges

Let T

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

		3			
1			2		

They have index vectors

$$I = [0, 0, 0, 0, 1, 1, 2] \quad \text{and} \quad I = [0, 0, 1, 1, 1]$$

Charges

Let T

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

		3			
1			2		

They have index vectors

$$I = [0, 0, 0, 0, 1, 1, 2] \quad \text{and} \quad I = [0, 0, 1, 1, 1]$$

Charges

Let T

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7

		3			
1			2		

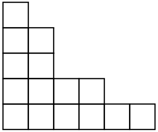
They have index vectors

$$I = [0, 0, 0, 0, 1, 1, 2], \quad I = [0, 0, 1, 1, 1], \quad \text{and} \quad I = [0, 1, 1].$$

Charges

Let T

6					
4	5				
3	4				
2	2	3	5		
1	1	1	2	3	7



They have index vectors

$$I = [0, 0, 0, 0, 1, 1, 2], \quad I = [0, 0, 1, 1, 1], \quad \text{and} \quad I = [0, 1, 1].$$

So $\mathrm{charge}(T) = 9$.

The σ_i-Operator

The σ_i-Operator

Definition 2.2

Let σ_i be the operator which acts on a SSTY T with a labels i and b labels $i+1$ as follows:

The σ_i-Operator

Definition 2.2

Let σ_i be the operator which acts on a SSTY T with a labels i and b labels $i+1$ as follows:

- Take the subword of w consisting of only is and $(i+1)$s.

The σ_i-Operator

Definition 2.2

Let σ_i be the operator which acts on a SSTY T with a labels i and b labels $i+1$ as follows:

- Take the subword of w consisting of only is and $(i+1)$s.
- Place a closed parenthesis ")" under each letter i.

The σ_i-Operator

Definition 2.2

Let σ_i be the operator which acts on a SSTY T with a labels i and b labels $i+1$ as follows:

- Take the subword of w consisting of only is and $(i+1)$s.
- Place a closed parenthesis ")" under each letter i.
- Place an open parenthesis "(" under each letter $i+1$.

The σ_i-Operator

Definition 2.2

Let σ_i be the operator which acts on a SSTY T with a labels i and b labels $i+1$ as follows:

- Take the subword of w consisting of only is and $(i+1)$s.
- Place a closed parenthesis ")" under each letter i.
- Place an open parenthesis "(" under each letter $i+1$.
- From right to left change the free parentheses such that the reading word has b labels i and a labels $i+1$.

The σ_i-Operator

Definition 2.2

Let σ_i be the operator which acts on a SSTY T with a labels i and b labels $i+1$ as follows:

- Take the subword of w consisting of only is and $(i+1)$s.
- Place a closed parenthesis ")" under each letter i.
- Place an open parenthesis "(" under each letter $i+1$.
- From right to left change the free parentheses such that the reading word has b labels i and a labels $i+1$.
- Place the new word back into $\mathrm{shape}(T)$.

The σ_i-Operator

Consider

4	5	5						
3	3	4	6	6				
2	2	3	3	3	5	6		
1	1	1	1	2	2	3	3	4

The σ_i-Operator

Consider

4	5	5						
3	3	4	6	6				
2	2	3	3	3	5	6		
1	1	1	1	2	2	3	3	4

There are $a = 7$ threes and $b = 4$ fours.

The σ_i-Operator

Consider

4	5	5						
3	3	4	6	6				
2	2	3	3	3	5	6		
1	1	1	1	2	2	3	3	4

There are $a = 7$ threes and $b = 4$ fours. The action of σ_3 is

$$\begin{matrix} 4334333334 \\ ())()))))(\end{matrix} \rightarrow \begin{matrix} 4334344444 \\ ())()(((((\end{matrix}$$

The σ_i-Operator

Consider

4	5	5						
3	3	4	6	6				
2	2	3	3	3	5	6		
1	1	1	1	2	2	3	3	4

There are $a = 7$ threes and $b = 4$ fours. The action of σ_3 is

$$4334333334 \to 4334344444$$

( )) ( ))))) (( )) ( ) (((((

Producing

4	5	5						
3	3	4	6	6				
2	2	3	4	4	5	6		
1	1	1	1	2	2	4	4	4

The $\mathbb{B}_r$-Operator

The $\mathbb{B}_r$-Operator

Definition 2.3

Let $\mathbb{B}_r$ be the operator which acts on SSTY T as follows:

The $\mathbb{B}_r$-Operator

Definition 2.3

Let $\mathbb{B}_r$ be the operator which acts on SSTY T as follows:

- Adds a horizontal strip of size r to T in all possible ways.

The $\mathbb{B}_r$-Operator

Definition 2.3

Let $\mathbb{B}_r$ be the operator which acts on SSTY T as follows:

- Adds a horizontal strip of size r to T in all possible ways.
- Acts by operators σ_i to change the weight so that there are r labels 1.

The $\mathbb{B}_r$-Operator

Definition 2.3

Let $\mathbb{B}_r$ be the operator which acts on SSTY T as follows:

- Adds a horizontal strip of size r to T in all possible ways.
- Acts by operators σ_i to change the weight so that there are r labels 1.

$\mathbb{B}_3$ acts on

3	
2	
1	1

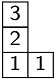
by returning the following set tableaux:

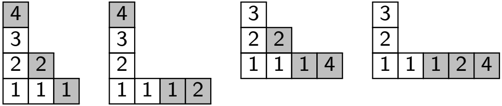
The $\mathbb{B}_r$-Operator

Definition 2.3

Let $\mathbb{B}_r$ be the operator which acts on SSTY T as follows:

- Adds a horizontal strip of size r to T in all possible ways.
- Acts by operators σ_i to change the weight so that there are r labels 1.

$\mathbb{B}_3$ acts on  by returning the following set tableaux:



The Schensted Algorithm

The Schensted Algorithm

Definition 2.4

Given a SSYT T, let T_1 be the first row of T and $T_{2\sim}$ be the SSYT with T_1 removed.

The Schensted Algorithm

Definition 2.4

Given a SSYT T, let T_1 be the first row of T and $T_{2\sim}$ be the SSYT with T_1 removed. Given a word w we will construct a SSYT by insertion $k \to T$ recursively as follows:

The Schensted Algorithm

Definition 2.4

Given a SSYT T, let T_1 be the first row of T and $T_{2\sim}$ be the SSYT with T_1 removed. Given a word w we will construct a SSYT by insertion $k \rightarrow T$ recursively as follows:

- Start with the leftmost letter in w, and call it k.

The Schensted Algorithm

Definition 2.4

Given a SSYT T, let T_1 be the first row of T and $T_{2\sim}$ be the SSYT with T_1 removed. Given a word w we will construct a SSYT by insertion $k \to T$ recursively as follows:

- Start with the leftmost letter in w, and call it k.
- If $T = \varnothing$, $k \to T$ is $\boxed{k}$.

The Schensted Algorithm

Definition 2.4

Given a SSYT T, let T_1 be the first row of T and $T_{2\sim}$ be the SSYT with T_1 removed. Given a word w we will construct a SSYT by insertion $k \to T$ recursively as follows:

- Start with the leftmost letter in w, and call it k.
- If $T = \varnothing$, $k \to T$ is $\boxed{k}$.
- If k is larger than the largest entry in T_1, then $k \to T$ is given by appending k at the end of T_1.

The Schensted Algorithm

Definition 2.4

Given a SSYT T, let T_1 be the first row of T and $T_{2\sim}$ be the SSYT with T_1 removed. Given a word w we will construct a SSYT by insertion $k \to T$ recursively as follows:

- Start with the leftmost letter in w, and call it k.
- If $T = \varnothing$, $k \to T$ is $\boxed{k}$.
- If k is larger than the largest entry in T_1, then $k \to T$ is given by appending k at the end of T_1.
- Otherwise, find the leftmost entry in T_1 larger than k and let it be k'. Then $k \to T$ is given as the SSYT with first row T_1 except that k' has been replaced by k and the remaining rows are given by the insertion $T_{2\sim} \to k'$.

The Schensted Algorithm

Let $w = 34112312$.

The Schensted Algorithm

Let $w = 34112312$. The the Schensted algorithm yields:

$$\varnothing$$

The Schensted Algorithm

Let w = 34112312. The the Schensted algorithm yields:

3

The Schensted Algorithm

Let w = 34112312. The the Schensted algorithm yields:

3	4

The Schensted Algorithm

Let w = 34112312. The the Schensted algorithm yields:

3	
1	4

The Schensted Algorithm

Let w = 34112312. The the Schensted algorithm yields:

3	4
1	1

The Schensted Algorithm

Let w = 34112312. The the Schensted algorithm yields:

3	4	
1	1	2

The Schensted Algorithm

Let $w = 34112312$. The the Schensted algorithm yields:

3	4		
1	1	2	3

The Schensted Algorithm

Let w = 34112312. The the Schensted algorithm yields:

3			
2	4		
1	1	1	3

The Schensted Algorithm

Let w = 34112312. The the Schensted algorithm yields:

3	4		
2	3		
1	1	1	2

The Schensted Algorithm

Let w = 34112312. The the Schensted algorithm yields:

3	4		
2	3		
1	1	1	2

The Schensted algorithm is a piece of a well-known algorithm: Robinson-Schensted-Kunuth algorithm (RKS).

The Schensted Algorithm

Let w = 34112312. The the Schensted algorithm yields:

3	4		
2	3		
1	1	1	2

The Schensted algorithm is a piece of a well-known algorithm: Robinson-Schensted-Kunuth algorithm (RKS). Denote the Schensted algorithm by RKS.

Katabolizable Tableau

Katabolizable Tableau

Let $\lambda^{(*)} = (\lambda^{(1)}, \ldots, \lambda^{(r)})$ be a sequence of partitions with $r = \ell(\lambda^{(1)})$.

Katabolizable Tableau

Let $\lambda^{(*)} = (\lambda^{(1)}, \ldots, \lambda^{(r)})$ be a sequence of partitions with $r = \ell(\lambda^{(1)})$. Decompose $w = uv$ where u is the largest subword not containing r.

Katabolizable Tableau

Let $\lambda^{(*)} = (\lambda^{(1)}, \ldots, \lambda^{(r)})$ be a sequence of partitions with $r = \ell(\lambda^{(1)})$. Decompose $w = uv$ where u is the largest subword not containing r. Let v' be v with letters $1, \ldots, r$ deleted.

Katabolizable Tableau

Let $\lambda^{(*)} = (\lambda^{(1)}, \ldots, \lambda^{(r)})$ be a sequence of partitions with $r = \ell(\lambda^{(1)})$. Decompose $w = uv$ where u is the largest subword not containing r. Let v' be v with letters $1, \ldots, r$ deleted.

Definition 2.5

T is katabolizable with respect to $\lambda^{(*)}$ if

Katabolizable Tableau

Let $\lambda^{(*)} = (\lambda^{(1)}, \ldots, \lambda^{(r)})$ be a sequence of partitions with $r = \ell(\lambda^{(1)})$. Decompose $w = uv$ where u is the largest subword not containing r. Let v' be v with letters $1, \ldots, r$ deleted.

Definition 2.5

T is katabolizable with respect to $\lambda^{(*)}$ if

- T contains a subtableau of shape and weight $\lambda^{(1)}$.

Katabolizable Tableau

Let $\lambda^{(*)} = (\lambda^{(1)}, \ldots, \lambda^{(r)})$ be a sequence of partitions with $r = \ell(\lambda^{(1)})$. Decompose $w = uv$ where u is the largest subword not containing r. Let v' be v with letters $1, \ldots, r$ deleted.

Definition 2.5

T is katabolizable with respect to $\lambda^{(*)}$ if

- T contains a subtableau of shape and weight $\lambda^{(1)}$.
- $\mathrm{RKS}(v'u)$ is $(\lambda^{(2)}, \ldots, \lambda^{(r)})$ katabolizable with labels shifted by r.

Katabolizable Tableau

3			
2	2	3	
1	1	1	4

is $((3,2),(2,1))$ katabolizable.

Katabolizable Tableau

3			
2	2	3	
1	1	1	4

is $((3,2),(2,1))$ katabolizable.

$$r = 2, \quad u = 3, \quad v = 2231114, \quad \text{and} \quad v' = 34.$$

Katabolizable Tableau

3			
2	2	3	
1	1	1	4

is $((3,2),(2,1))$ katabolizable.

$$r = 2, \quad u = 3, \quad v = 2231114, \quad \text{and} \quad v' = 34.$$

$\text{RKS}(v'u)$ is

4	
3	3

.

The $\mathbb{K}^{\to k}$-Operator

The $\mathbb{K}^{\to k}$-Operator

Definition 2.6

Let $\mathbb{K}^{\to k}$ be the operator on a set of tableaux with partition weight λ which acts by removing all tableau that are not katabolizable with respect to the k-split of λ and keeping all others.

The $\mathbb{K}^{\rightarrow k}$-Operator

Consider the set

3		
2	4	
1	1	5

3		
2	5	
1	1	4

4		
2	3	
1	1	5

4		
2	5	
1	1	3

The $\mathbb{K}^{\to k}$-Operator

Consider the set

3		
2	4	
1	1	5

3		
2	5	
1	1	4

4		
2	3	
1	1	5

4		
2	5	
1	1	3

The action of $\mathbb{K}^{\to 3}$ only keeps

3		
2	5	
1	1	4

Atoms

Atoms

Definition 2.7

If λ is a k-bounded partition, the k-atom $\mathbb{A}_\lambda^{(k)}$ is the set of tableaux recursively defined by

Atoms

Definition 2.7

If λ is a k-bounded partition, the k-atom $\mathbb{A}_{\lambda}^{(k)}$ is the set of tableaux recursively defined by

$$\mathbb{A}_{\lambda}^{(k)} = \mathbb{K}^{\to k}\mathbb{B}_{\lambda_1}\mathbb{A}_{(\lambda_2,\ldots,\lambda_{\ell(\lambda)})}^{(k)},$$

where $\mathbb{A}_{\varnothing}^{(k)}$ is the empty tableau.

Atoms

Definition 2.7

If λ is a k-bounded partition, the k-atom $\mathbb{A}_\lambda^{(k)}$ is the set of tableaux recursively defined by

$$\mathbb{A}_\lambda^{(k)} = \mathbb{K}^{\to k}\mathbb{B}_{\lambda_1}\mathbb{A}^{(k)}_{(\lambda_2,\ldots,\lambda_{\ell(\lambda)})},$$

where $\mathbb{A}_\varnothing^{(k)}$ is the empty tableau.

Some atoms:

Atoms

Definition 2.7

If λ is a k-bounded partition, the k-atom $\mathbb{A}_\lambda^{(k)}$ is the set of tableaux recursively defined by

$$\mathbb{A}_\lambda^{(k)} = \mathbb{K}^{\to k}\mathbb{B}_{\lambda_1}\mathbb{A}^{(k)}_{(\lambda_2,\ldots,\lambda_{\ell(\lambda)})},$$

where $\mathbb{A}_\varnothing^{(k)}$ is the empty tableau.

Some atoms:

- $\mathbb{A}_{11}^{(3)} = \begin{array}{|c|}\hline 2 \\ \hline 1 \\ \hline\end{array}$

Atoms

Definition 2.7

If λ is a k-bounded partition, the k-atom $\mathbb{A}_{\lambda}^{(k)}$ is the set of tableaux recursively defined by

$$\mathbb{A}_{\lambda}^{(k)} = \mathbb{K}^{\rightarrow k}\mathbb{B}_{\lambda_1}\mathbb{A}_{(\lambda_2,\ldots,\lambda_{\ell(\lambda)})}^{(k)},$$

where $\mathbb{A}_{\varnothing}^{(k)}$ is the empty tableau.

Some atoms:

- $\mathbb{A}_{11}^{(3)} = \begin{array}{|c|}\hline 2 \\ \hline 1 \\ \hline \end{array}$
- $\mathbb{A}_{221}^{(3)} = \begin{array}{|c|c}\hline 3 & \\ \cline{1-1} 2 & \\ \hline 1 & \multicolumn{1}{c|}{1} \\ \hline \end{array} \quad \begin{array}{|c|cc}\hline 2 & & \\ \hline 1 & \multicolumn{1}{|c|}{1} & \multicolumn{1}{c|}{3} \\ \hline \end{array}$

Atoms

Definition 2.7

If λ is a k-bounded partition, the k-atom $\mathbb{A}_\lambda^{(k)}$ is the set of tableaux recursively defined by

$$\mathbb{A}_\lambda^{(k)} = \mathbb{K}^{\to k}\mathbb{B}_{\lambda_1}\mathbb{A}_{(\lambda_2,\dots,\lambda_{\ell(\lambda)})}^{(k)},$$

where $\mathbb{A}_\varnothing^{(k)}$ is the empty tableau.

Some atoms:

- $\mathbb{A}_{11}^{(3)} = \begin{array}{|c|}\hline 2 \\ \hline 1 \\ \hline \end{array}$
- $\mathbb{A}_{221}^{(3)} = \begin{array}{|c|c}\hline 3 & \\ \cline{1-1} 2 & \\ \hline 1 & \multicolumn{1}{c|}{1} \\ \hline \end{array} \quad \begin{array}{|c|c|c|}\cline{1-1} 2 & \multicolumn{2}{c}{} \\ \hline 1 & 1 & 3 \\ \hline \end{array}$
- $\mathbb{A}_{3211}^{(4)} = \begin{array}{|c|c|c|}\cline{1-1} 4 & \multicolumn{2}{c}{} \\ \cline{1-1} 3 & \multicolumn{2}{c}{} \\ \cline{1-2} 2 & 2 & \multicolumn{1}{c}{} \\ \hline 1 & 1 & 1 \\ \hline \end{array} \quad \begin{array}{|c|c|c|c|}\cline{1-1} 3 & \multicolumn{3}{c}{} \\ \cline{1-2} 2 & 2 & \multicolumn{2}{c}{} \\ \hline 1 & 1 & 1 & 4 \\ \hline \end{array}$

k-Schur Functions

k-Schur Functions

Definition 2.8

The k-Schur function $s_\lambda^{(k)}$ is defined by

$$s_\lambda^{(k)} := \sum_{T \in \mathbb{A}_\lambda^{(k)}} t^{\mathrm{charge}(T)} s_{\mathrm{shape}(T)}.$$

k-Schur Functions

Definition 2.8

The k-Schur function $s_\lambda^{(k)}$ is defined by

$$s_\lambda^{(k)} := \sum_{T \in \mathbb{A}_\lambda^{(k)}} t^{\mathrm{charge}(T)} s_{\mathrm{shape}(T)}.$$

Take away: $s_\lambda^{(k)}$ is a sum of Schur functions with coefficients in $\mathbb{Z}[t]$.

k-Schur Functions

Definition 2.8

The k-Schur function $s_\lambda^{(k)}$ is defined by

$$s_\lambda^{(k)} := \sum_{T \in \mathbb{A}_\lambda^{(k)}} t^{\mathrm{charge}(T)} s_{\mathrm{shape}(T)}.$$

Take away: $s_\lambda^{(k)}$ is a sum of Schur functions with coefficients in $\mathbb{Z}[t]$.

Some k-Schur functions:

k-Schur Functions

Definition 2.8

The k-Schur function $s_\lambda^{(k)}$ is defined by

$$s_\lambda^{(k)} := \sum_{T \in \mathbb{A}_\lambda^{(k)}} t^{\mathrm{charge}(T)} s_{\mathrm{shape}(T)}.$$

Take away: $s_\lambda^{(k)}$ is a sum of Schur functions with coefficients in $\mathbb{Z}[t]$.

Some k-Schur functions:

- $s_{11}^{(3)} = s_{11}.$

k-Schur Functions

Definition 2.8

The k-Schur function $s_\lambda^{(k)}$ is defined by

$$s_\lambda^{(k)} := \sum_{T \in \mathbb{A}_\lambda^{(k)}} t^{\mathrm{charge}(T)} s_{\mathrm{shape}(T)}.$$

Take away: $s_\lambda^{(k)}$ is a sum of Schur functions with coefficients in $\mathbb{Z}[t]$.

Some k-Schur functions:

- $s_{11}^{(3)} = s_{11}$.
- $s_{211}^{(3)} = s_{211} + ts_{31}$.

k-Schur Functions

Definition 2.8

The k-Schur function $s_\lambda^{(k)}$ is defined by

$$s_\lambda^{(k)} := \sum_{T \in \mathbb{A}_\lambda^{(k)}} t^{\mathrm{charge}(T)} s_{\mathrm{shape}(T)}.$$

Take away: $s_\lambda^{(k)}$ is a sum of Schur functions with coefficients in $\mathbb{Z}[t]$.

Some k-Schur functions:

- $s_{11}^{(3)} = s_{11}$.
- $s_{211}^{(3)} = s_{211} + ts_{31}$.
- $s_{3211}^{(4)} = s_{3211} + ts_{421}$.

Outline

References

Thanks!

- T. Lam, L. Lapointe, J. Morse, A. Schilling, M. Shimozono, M. Zabrocki: *k*-Schur functions and affine Schubert calculus, 2012.