The combinatorics of symmetric polynomials in types A, B, and C

Álvaro Gutiérrez Cáceres

Born 19th May 1999 in El Puerto de Santa María, Spain Submitted July 26, 2023, last udpated: August 14, 2023

> Master's Thesis Mathematics Advisor: Prof. Dr. Catharina Stroppel Second Advisor: Dr. Jacob Matherne MATHEMATISCHES INSTITUT

Mathematisch-Naturwissenschaftliche Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

Contents

In	troduction	2
Ι	Preliminaries	4
1	A preliminary account of symmetric polynomials	5
2	Representation theory of the classical Lie algebras	9
II	Symmetric polynomials in types A, B, and C	19
3	The Lie theoretic definition	21
4	The combinatorial definition	23
5	Pattern combinatorics	26
6	The algebraic definition (Jacobi–Trudi determinants)	31
7	The Lie theoretic and algebraic definitions coincide	38
8	Other combinatorial models for type C	42
9	A crystal structure on type C tableaux	50
10	The Lie theoretic and combinatorial definitions coincide	59
Aŗ	ppendices	62
A	Type C Bender–Knuth involutions	63
B	Crystal reflections and Bender-Knuth	66
С	Axioms for De Concini's and Kashiwara's tableaux	67
D	A SageMath library for symplectic tableaux	72
Bi	bliography	83

Introduction

«¡Qué altos los balcones de mi casa!» Rafael Alberti

This is an electronic version of the thesis which was updated beyond the submitted deadline. The newest version is available in my web page. Last updated: August 14, 2023.

Every modern account of representation theory feeds on the world of combinatorics. This is especially notable in type A representation theory, where the combinatorial models are further developed and better established. In particular, many aspects of character theory have been reduced to purely combinatorial problems.

Characters of representations in type A (that is, complex representations of S_n , GL(*n*), or $\mathfrak{sl}(n)$) may be understood as symmetric polynomials. In particular, characters of *irreducible* representations of $\mathfrak{sl}(n)$ are identified with a family of symmetric polynomials known as *Schur polynomials*. On the other hand, these agree with the generating functions of certain classes of combinatorial objects called *semistandard Young tableaux*. This enables one to study some properties of the original representation (namely, those properties encapsulated by the character) in a combinatorial way. Reciprocally, representation theory often provides results on symmetric functions and tableaux that are only later tackled combinatorially.

For the other classical types (orthogonal and symplectic Lie algebras), the combinatorics are not as well-studied. There are many reasons for this, e.g. there is more interest in the properties of S_n than in the properties of the other Weyl groups or the Brauer algebras of types B, C, and D. However, there have been many efforts to change this situation. We compare and contrast three distinct approaches.

The first approach, initiated by King [Kin76], and continued by Stanley, Sundaram [Sun86, Sun90], and others, is to mimic as closely as possible the combinatorics of type A. For this matter, the analogue of Schur polynomials in other types are defined as some generating functions of some kinds of tableaux. This may be referred to as the combinatorialist approach.

Another approach is representation theoretic in nature. The symmetric polynomials are defined via Weyl's character formula, and studied via crystal theory. The development of crystal theory by Kashiwara and Nakashima enabled them to identify their crystal bases in each type with some kind of tableaux [KN94, HK02, Lit90, BS17] (in particular, recovering the definition of semistandard Young tableaux for type A). This may be referred to as the Lie theorist approach.

A final strategy is to proceed purely algebraically: the algebraist approach. In this line, Koike and Terada [KT87] provide formulas to compute the irreducible characters of types B, C, and D in terms of usual symmetric polynomials [FH91]. In a later stage, these formulas may be explained combinatorially [Oka89, SV16, FK97].

It is our goal in this thesis to present these three approaches, as well as to establish connections between them. We try to develop the theory for all types simultaneously, but focusing on types C and B; the combinatorics of type D are less well-understood and a complete survey of the literature is beyond the scope of this work.



We therefore introduce our irreducible characters three times. We begin by giving their Lie theoretic definition in Chapter 3, as characters of the irreducible representations of the classical Lie algebras. In Chapter 4, we introduce some combinatorial models that let us define these characters as certain generating functions, as part of the combinatorialist approach. Inspired by the classical proof of well-definedness of this definition for type A, due to Bender and Knuth [BK72], we give in Chapter 5 a first self-contained and elementary proof of well-definedness of this combinatorial definition for types B and C.

For the algebraist approach, a third definition of the irreducible characters is given in Chapter 6 in the form of two theorems. This allows one to write our type B and C irreducible characters as specializations of certain type A symmetric polynomials. Mimicking type A theory [Lin73, GV85], we show that these agree with the combinatorial definitions via a lattice path argument, originally developed in [Oka89]. We show that the algebraic definition agrees with the Lie theoretic definition in Chapter 7. (In particular, note that this shows indirectly that the combinatorial and the Lie theoretic definitions agree.)

In Chapter 8, we discuss different combinatorial models for type C. In particular, we define the tableaux which arise from crystal theory (as part of the Lie theorist approach). We describe said crystal structure in Chapter 9. Notably, we state novel descriptions of a type of tableaux due to De Concini [DeC79], and we postpone the proofs to Appendix C. With this and invoking the theory of crystal bases, we are able to show in Chapter 10 that the combinatorial and the Lie theoretic definitions agree. In total, we give two proofs of this fact.

A SageMath library for the different type C combinatorial models is included in Appendix D.

Acknowledgements

I would like to thank Jacob Matherne for his many life lessons and profound conversations. I am immensely indebted to my three dear friends Giulio Ricci, Barbara Muniz, and Chris Miliónis for lending me an ear and helping me to overcome the mathematical obstacles that arose during the development of this work. Finally, I'd like to thank the people who have given, or continue to give, meaning to my life through many special moments, and without whom I would not have had such an enriching experience in Bonn.

Part I

Preliminaries

Chapter 1

A preliminary account of symmetric polynomials

In this chapter we give a self-contained introduction to the theory of symmetric polynomials via the study of three families of polynomials. In particular, we work towards some results that will be useful later in this work. We refer to [ALRS13, Sag01, Sam17, Sta99] for a more detailed and complete account of these objects.

Definition 1.1. A **symmetric polynomial** in *n* variables is a multivariate polynomial with complex coefficients that is invariant by any permutation of its variables. We call Λ_n the set of symmetric polynomials in *n* variables.

Sums and products of symmetric polynomials are again symmetric. Therefore, Λ_n has an algebra structure. Unless otherwise specified, we fix some variables $x_1, ..., x_n$ and write $\Lambda_n = \mathbb{C}[x_1, ..., x_n]^{\mathbb{S}_n}$.

Theorem 1.2. The ring Λ_n inherits a \mathbb{Z} -graded algebra structure $\Lambda_n = \bigoplus_{d \ge 0} \Lambda_n^d$ from the algebra $\mathbb{C}[x_1, ..., x_n]$, where $\Lambda_n^d := \Lambda_n \cap \mathbb{C}[x_1, ..., x_n]_d$ is the degree d part of Λ_n .

We typically represent these as polynomials in $x_1, ..., x_n$ and say that they are **evaluated at the alphabet** $X = x_1 + \cdots + x_n$. We may use the notations f, $f(x_1, ..., x_n)$, and f(X) interchangeably for elements of Λ_n . We write $\Lambda_n(X)$ whenever we want to explicitly indicate the alphabet in which the polynomials are evaluated.

Notation. We write alphabets as sums (rather than tuples) to emphasize that the order of the variables does not matter. Another feature of this notation is the following: given two alphabets $X = x_1 + \cdots + x_n$ and $Y = y_1 + \cdots + y_m$, one may write f(X + Y) to refer to a symmetric function in $\Lambda_{m+n}(X + Y) = \Lambda_{m+n}$.

We let 0 be the empty alphabet. That is, for $f \in \Lambda_n$, we have $f(X+0) := f(X) \in \Lambda_n$. For $f \in \Lambda_{n+1}$, we define f(X+1) as the polynomial $f(x_1, ..., x_n, 1) \in \Lambda_n$. In general, however, it will not be useful to think of symmetric polynomials as functions, but rather as formal sums or generating functions (this will become useful later). In particular, one should not confuse f(X) with a symmetric polynomial in one variable evaluated at a sum.

As a remark, this notation makes sense in a broader context. It is an instance of *plethystic substitution* [Sta99, ALRS13].

It is of central importance in the study of symmetric polynomials to describe, analyze, and relate various bases of Λ_n . These bases will turn out to be indexed by **partitions**, which are weakly decreasing finite sequences of non-negative integers, e.g. $\lambda = (7, 4, 4, 4, 2, 1, 0, 0)$. We say $\lambda_1 = 7, \lambda_2 = 4, ...$ are the parts of λ . By convention, tailing 0s are ignored when expressing the partition, and λ_N for $N \gg 1$ is

taken to be 0. Also, repeated parts are indicated by an exponent. For instance, $\lambda = (7, 4^3, 2, 1)$. Given a positive integer *d*, we write $\lambda \vdash d$ and say that λ is a partition of *d* if $\sum \lambda_i = d$. Reciprocally, we define by $|\lambda| := \sum \lambda_i$ the size of λ . The number of nonzero parts of λ is denoted by $l(\lambda)$ and called the length of λ . We often represent partitions through their **Young diagram**, which we obtain by drawing an array of left-justified boxes, with λ_i boxes in row *i*. The transpose λ' of a partition λ is the partition whose Young diagram is the transpose of the Young diagram of λ .

Example 1.3. Let $\lambda = (7, 4^3, 2, 1)$. We write

$$\lambda =$$
 and $\lambda' =$ (6, 5, 4², 1³).

We may now define three important families of symmetric polynomials.

Definition 1.4. Given a partition $\lambda = (\lambda_1, ..., \lambda_l)$, the **monomial symmetric polynomial** m_{λ} indexed by λ is the sum over the \mathbb{S}_n -orbit of $x^{\lambda} := x_1^{\lambda_1} x_2^{\lambda_2} \cdots x_l^{\lambda_l}$. (Here, the orbit is interpreted as a set, rather than a multiset.)

Example 1.5. Let n = 3 (that is, $X = x_1 + x_2 + x_3$). Then,

$$m_{\boxplus} = \sum_{y \in \mathbb{S}_3, x^{\boxplus}} y = x_1^2 x_2^2 + x_1^2 x_3^2 + x_2^2 x_3^2.$$

Since any symmetric polynomial $f = \sum_{\alpha} c_{\alpha} x^{\alpha}$ can be written as $f = \sum_{\lambda} c_{\lambda} m^{\lambda}$, we conclude that $\{m_{\lambda}\}_{\lambda \vdash d}$ is a basis of Λ_n^d . It is our goal now to show that the next two families of symmetric polynomials also form bases of Λ_n^d .

Definition 1.6. Let *d* be a non-negative integer. The **elementary symmetric polynomial** $e_d \in \Lambda_n^d$ indexed by *d* is the symmetric polynomial given by the sum of all square-free monomials of degree *d*. The **completely homogeneous symmetric polynomial** $h_d \in \Lambda_n^d$ indexed by *d* is the symmetric polynomial given by the sum of all monomials of degree *d*.

Given a partition λ , we let $e_{\lambda} := \prod e_{\lambda_i}$ and $h_{\lambda} := \prod h_{\lambda_i}$.

Example 1.7. Let n = 3 (that is, $X = x_1 + x_2 + x_3$).

$$\begin{split} e_{\rm FF} &= e_2 e_1 = (x_1 x_2 + x_1 x_3 + x_2 x_3)(x_1 + x_2 + x_3) \\ &= x_1^2 x_2 + x_1 x_2^2 + x_1^2 x_3 + 3 x_1 x_2 x_3 + x_2^2 x_3 + x_1 x_3^2 + x_2 x_3^2 \\ &= m_{\rm FF} + 3 m_{\rm FF} \\ h_{\rm cc} &= h_2 = x_1^2 + x_1 x_2 + x_2^2 + x_1 x_3 + x_2 x_3 + x_3^2 \\ &= m_{\rm cc} + m_{\rm H}. \end{split}$$

Theorem 1.8. The set $\{e_{\lambda}\}_{\lambda \vdash d}$ of elementary symmetric polynomials indexed by partitions of d is a basis of Λ_n^d .

For the purpose of this proof, we need to introduce a partial order on partitions of a fixed size *d*, called the **dominance order**: $\lambda \ge \mu$ if and only if $\lambda_1 + \cdots + \lambda_i \ge \mu_1 + \cdots + \mu_i$ for all *i*. An immediate property of this order is that $\lambda \le \mu$ if and only if $\mu' \le \lambda'$. Indeed, by contradiction, if $\sum_{i=1}^{i} \lambda'_i \le \sum_{i=1}^{i} \mu'_i$ for all $i = 1, ..., i_0$ but $\sum_{i=1}^{i_0} \lambda'_i > \sum_{i=1}^{i_0} \mu'_i$, then

$$\sum_{i=0}^{\mu_{i_0}} (\mu_j - i_0) = \mu'_{i_0+1} + \mu'_{i_0+2} + \dots > \lambda'_{i_0+1} + \lambda'_{i_0+2} + \dots = \sum_{i=0}^{\lambda_{i_0}} (\lambda_j - i_0) \ge \sum_{i=0}^{\mu_{i_0}} (\lambda_j - i_0),$$

CHAPTER 1. A PRELIMINARY ACCOUNT OF SYMMETRIC POLYNOMIALS

which gives $\sum_{i=0}^{\mu_{i_0}} \mu_i > \sum_{i=0}^{\mu_{i_0}} \lambda_i$, a contradiction. The reciprocal is shown similarly.

The reverse lexicographic order (\leq) defines a total order on the set of partitions of size *d* which is compatible with the dominance order. Bases of Λ_n^d will be thought as ordered bases in the following.

Proof. Fix $\lambda \vdash d$ and express e_{λ} in terms of the monomial basis, $e_{\lambda} = \sum_{\mu} a_{\lambda,\mu} m_{\mu}$. From the definition of elementary symmetric polynomials and monomial symmetric polynomials, we note that $a_{\lambda,\mu} \ge 0$ for all λ, μ . If $a_{\lambda,\mu} \ne 0$, then we claim $\lambda' \ge \mu$. Indeed, write $e_{\lambda} = e_{\lambda_1} \cdots e_{\lambda_l}$. Then, the biggest possible value of μ_1 among the partitions μ such that x^{μ} appears in e_{λ} is precisely the number of factors, $l(\lambda) = \lambda'_1$. For such a μ , the biggest possible value of μ_2 is the number of factors that are not e_1 , which is λ'_2 . This reasoning gives λ' as the biggest partition μ such that $a_{\lambda,\mu} \ne 0$. Furthermore, $a_{\lambda,\lambda'} = 1$.

This means that the matrix that transforms $\{e_{\lambda}\}_{\lambda \vdash d}$ into $\{m_{\lambda}\}_{\lambda \vdash d}$ can be written as a triangular matrix with 1s in the diagonal. In particular it is invertible. Therefore $\{e_{\lambda}\}_{\lambda \vdash d}$ is a basis of Λ_n^d .

Corollary 1.9. The set $\{e_i\}_{i>0}$ is a set of algebraically independent generators of Λ_n .

Note 1.10. The coefficients in this change of basis matrix have a combinatorial interpretation in terms of some 0-1-matrices. The interested reader may find this in [Sta99].

There is another way of thinking about these latter two families of symmetric polynomials; as generating functions. Although seemingly arbitrary, it will be apparent later how this interpretation is relevant to algebraic combinatorialists. We let [n] denote $\{1, ..., n\}$.

Proposition 1.11. *Fix an integer n (and, in turn, an alphabet X).*

- 1. The elementary symmetric polynomial e_d agrees with the generating function of strictly increasing sequences in $[n]^d$.
- 2. The completely homogeneous symmetric polynomial h_d agrees with the generating function of weakly increasing sequences in $[n]^d$.

Proof. Given a square-free monomial $x_{i_1} \cdots x_{i_d}$ consider the tuple $(i_1, ..., i_d)$ of its indexes, which we may assume to be strictly increasing without loss of generality. This gives a bijection from the set of summands of e_d to the set of strictly increasing sequences in $[n]^d$.

Given a monomial $x_{i_1} \cdots x_{i_d}$ consider the tuple $(i_1, ..., i_d)$ of its indexes, which we may assume to be weakly increasing without loss of generality. This gives a bijection from the set of summands of h_d to the set of weakly increasing sequences in $[n]^d$.

Theorem 1.12. We have the following identities:

$$E(t) := \sum_{d \ge 0} e_d t^d = \prod_{i \in [n]} (1 + x_i t) \quad and \quad H(t) := \sum_{d \ge 0} h_d t^d = \prod_{i \in [n]} \frac{1}{1 - x_i t}.$$

Proof. We show the second identity, the other one is shown similarly. Expand $(1 - x_i t)^{-1}$ as a geometric series, $1 + x_i t + x_i^2 t^2 + \cdots$. Now, a monomial in the right hand side of the identity has *t*-degree *d* if and only if it has *x*-degree *d*. We obtain such a monomial by choosing an element from each factor in the product. All *x*-monomials of degree *d* can be obtained this way and each one appears only once.

By looking at the term of *t*-degree *d* of the equation H(t)E(-t) = 1, we get the following identity.

Corollary 1.13. For each $d \ge 1$, we have $\sum_{i=0}^{d} (-1)^{i} e_{i} h_{d-i} = 0$.

The relationship between $\{e_{\lambda}\}_{\lambda}$ and $\{h_{\lambda}\}_{\lambda}$ seen in these previous results culminates in the form of an involution.

Definition-theorem 1.14. Let $\omega : \Lambda_n \to \Lambda_n$ be an algebra homomorphism defined on $\{e_i\}_{i>0}$ by $\omega(e_i) = h_i$. It is an involution, referred to as the ω involution.

Proof. We check that the ω involution is an involution. Consider Corollary 1.13. Taking ω yields

$$0 = \sum_{i=0}^{d} (-1)^{i} h_{i} \omega(h_{d-i}) = (-1)^{d} \sum_{i=0}^{d} (-1)^{i} \omega(h_{i}) h_{d-i}$$

for each *d*. But then, the generating function of $\{\omega(h_i)\}_{i>0}$ is E(t), showing that $\omega(h_i) = e_i$ for all *i*.

Since the ω involution is an homomorphism and an involution, it is in particular an automorphism. This, combined with Theorem 1.8 gives the following result.

Corollary 1.15. The set $\{h_{\lambda}\}_{\lambda \vdash d}$ of completely homogeneous symmetric polynomials indexed by partitions of *d* is a basis of Λ_n^d . The set $\{h_i\}_{i>0}$ is a set of algebraically independent generators of Λ_n .

Note 1.16. The ω involution gives, up to sign, an antipode for a Hopf algebra structure on Λ_n , in which the comultiplication can be chosen to be either the *plethysm* or the *Kronecker product* of symmetric functions. See [ALRS13] for a detailed description of the structure.

Chapter 2

Representation theory of the classical Lie algebras

We assume familiarity with Lie theory [Hum72, Bou02, Str22]. Nevertheless, when trying to study the classical Lie algebras, there are some (somewhat arbitrary) choices to be made. We dedicate this chapter to fix the notations, presentations, and realizations of these objects explicitly. In particular, our conventions will often differ from [Hum72] and [Bou02].

2.1 The classical Lie algebras

Our work will provide tools to study the so-called classical Lie algebras. There is no unique definition to what constitutes a classical Lie algebra, so clarification is needed. In this work, the classical Lie algebras are

- (A) the special linear (Lie) algebra $\mathfrak{sl}(n)$ and the general linear (Lie) algebra $\mathfrak{gl}(n)$ of degree n, for $n \in \mathbb{Z}_{>0}$,
- (B) the (odd) special orthogonal (Lie) algebra $\mathfrak{so}(2n+1)$ of degree 2n+1, for $n \in \mathbb{Z}_{>0}$,
- (C) the (even) symplectic (Lie) algebra $\mathfrak{sp}(2n)$ of degree 2n + 1, for $n \in \mathbb{Z}_{>0}$, and
- (D) the (even) special orthogonal (Lie) algebra $\mathfrak{so}(2n)$ of degree 2n, for $n \in \mathbb{Z}_{>0}$.

We may refer to them as the Lie algebras of type A, B, C, and D, respectively. In the following, we will define each of these Lie algebras and fix matrix realizations for each of them.

Definition 2.1. Let *V* be a vector space over \mathbb{C} .

- We define $\mathfrak{gl}(V)$ as the Lie algebra of endomorphisms of *V* with the commutator bracket [A, B] := AB BA, and $\mathfrak{sl}(V)$ as the Lie subalgebra of $\mathfrak{gl}(V)$ given by the trace 0 endomorphisms.
- Let (\cdot, \cdot) be a symmetric non-degenerate bilinear form on *V*. We define $\mathfrak{so}(V)$ as the Lie subalgebra of $\mathfrak{gl}(V)$ given by $\{A \in \mathfrak{gl}(V) : (Ax, y) + (x, Ay) = 0 \text{ for all } x, y \in V\}$.
- Let (\cdot, \cdot) be a skew-symmetric non-degenerate bilinear form on *V*. We define $\mathfrak{sp}(V)$ as the Lie subalgebra of $\mathfrak{gl}(V)$ given by $\{A \in \mathfrak{gl}(V) : (Ax, y) + (x, Ay) = 0 \text{ for all } x, y \in V\}$.

We write $\mathfrak{gl}(N)$, $\mathfrak{sl}(N)$, $\mathfrak{so}(N)$ and $\mathfrak{sp}(N)$ if $V = \mathbb{C}^N$.

Note 2.2. It is an easy exercise to check that these sets define Lie subalgebras. To show that the second and third items do not depend on the choice of form, we note that any two given symmetric (resp. skew-symmetric) invertible matrices are conjugate. This conjugation gives the isomorphism between two different realizations of the Lie algebras.

In what comes, we will fix bilinear forms and thus realizations for $\mathfrak{so}(N)$ and $\mathfrak{sp}(N)$. Let $J_N =$ antidiag $\{1, ..., 1\}$.

Definition 2.3. Let $A = \{a_{i,j}\}_{i,j} \in \mathfrak{gl}(N)$. We define the **anti-transpose** of A as the matrix $A^{\textcircled{}} := \{a_{N+1-j,N+1-i}\}_{i,j}$.

Example 2.4. For instance,

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}^{(t)} = \begin{pmatrix} 9 & 6 & 3 \\ 8 & 5 & 2 \\ 7 & 4 & 1 \end{pmatrix}.$$

Lemma 2.5. If we let (\cdot, \cdot) be the symmetric non-degenerate bilinear form given by J_N , then

$$\mathfrak{so}(N) = \{A \in \mathfrak{gl}(N) : A^{(t)} = -A\}$$

(In particular, note that when N is odd, the antidiagonal entries vanish.)

Proof. Let $A \in \mathfrak{so}(N)$. We have (Ax, y) = -(x, Ay) for all $x, y \in \mathbb{C}^N$. That is, $\sum_{ij} a_{N+1-j,i} x_i y_j = -\sum_{ij} a_{N+1-i,j} x_i y_j$. So we deduce $a_{ij} = -a_{N+1-j,N+1-i}$ for all i and j.

Lemma 2.6. If we let (\cdot, \cdot) be the skew-symmetric non-degenerate bilinear form given by $\begin{pmatrix} J_n \\ -J_n \end{pmatrix}$, then

$$\mathfrak{sp}(2n) = \left\{ A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathfrak{gl}(2n) : d = -a^{\textcircled{t}}, b = b^{\textcircled{t}}, c = c^{\textcircled{t}} \right\},\$$

where a, b, c and d are understood as elements of gl(n).

Proof. Set N = 2n + 1, let $A \in \mathfrak{sp}(2n)$. We have (Ax, y) = -(x, Ay) for all $x, y \in \mathbb{C}^{2n}$. That is,

$$\sum_{i < N} \sum_{j < N} (-1)^{\delta_{j \le n}} a_{N-j,i} x_i y_j = \sum_{i < N} \sum_{j < N} (-1)^{\delta_{i \le n}} a_{N-i,j} x_i y_j.$$

So we deduce $a_{ij} = (-1)^{\delta_{i \leq n}} (-1)^{\delta_{j \leq n}} a_{N-j,N-i}$ for all *i* and *j*.

The general and the special linear Lie algebras

A basis for $\mathfrak{gl}(n)$ is given by the elementary matrices E_{ij} for i, j = 1, ..., n. A basis of $\mathfrak{sl}(n)$ is given by

$$\{E_{ij} : i \neq j\} \sqcup \{E_{i,i} - E_{i+1,i+1} : i = 1, ..., n-1\}.$$

They have dimensions n^2 and $n^2 - 1$, respectively.

Lemma 2.7. The Lie algebra $\mathfrak{gl}(n)$ is a (trivial) central extension of $\mathfrak{sl}(n)$ by $\mathbb{C}I$.

We recall here the relevant definitions. We say that a Lie algebra \mathfrak{b} is a central extension of \mathfrak{c} by \mathfrak{a} if there is a short exact sequence of Lie algebra homomorphisms $\mathfrak{a} \to \mathfrak{b} \to \mathfrak{c}$ and $[\mathfrak{a}, \mathfrak{b}] = 0$. We say it is a trivial extension if the short exact sequence splits. For details, we refer to [Str22] or to [Sch08, Ch. 4].

Proof. Consider the short exact sequence $\mathbb{C}I \to \mathfrak{gl}(n) \to \mathfrak{sl}(n)$ where the first map is inclusion and the second map is $x \mapsto x - \frac{\operatorname{tr} x}{n}I$. These are both Lie algebra homomorphisms. We have a splitting $\mathfrak{sl}(n) \to \mathfrak{gl}(n)$ given by the inclusion.

In particular, $\mathfrak{gl}(n)$ is not semisimple. Below, we'll see how representations of $\mathfrak{gl}(n)$ can be studied from the representations of $\mathfrak{sl}(n)$. We now study the structure of $\mathfrak{sl}(n)$.

Lemma 2.8. A Cartan subalgebra of $\mathfrak{sl}(n)$ is given by the set $\mathfrak{h} = \mathfrak{h}_{\mathfrak{sl}(n)}$ of diagonal matrices of $\mathfrak{sl}(n)$.

Proof. We have that $[\mathfrak{h}, \mathfrak{h}] = 0$, therefore \mathfrak{h} is nilpotent. Its normalizer $N_{\mathfrak{sl}(n)}(\mathfrak{h})$ is the set of matrices $x \in \mathfrak{sl}(n)$ such that $[x, \mathfrak{h}] \subseteq \mathfrak{h}$. We have $[x, h] = xh - hx \in \mathfrak{h}$ for all $h \in \mathfrak{h}$. In particular, for $i \neq j$, we have $(xh)_{ij} = x_{ij}h_{ii} = (hx)_{ij} = x_{ij}h_{jj}$ for all $h \in \mathfrak{h}$. Necessarily, $x_{ij} = 0$ for $i \neq j$, and thus $x \in \mathfrak{h}$. So $N_{\mathfrak{sl}(n)}(\mathfrak{h}) = \mathfrak{h}$.

We will hereafter refer to \mathfrak{h} as *the* Cartan subalgebra of $\mathfrak{sl}(n)$, by abuse of language. Let $\mathfrak{h}^* = \mathfrak{h}^*_{\mathfrak{sl}(n)}$ be its dual space. It has a basis given by $\{\epsilon_{i+1} - \epsilon_i\}_{i < n}$, where $\epsilon_i(E_{ij}) = \delta_{ij}$.

Since a Cartan subalgebra $\mathfrak{h}_{\mathfrak{g}}$ of a semisimple Lie algebra \mathfrak{g} is abelian, we can write

$$\mathfrak{g} = \bigoplus_{\alpha \in \mathfrak{h}^*} \mathfrak{g}_{\alpha},$$

where $\mathfrak{g}_{\alpha} := \{x \in \mathfrak{g} : [x, h] = \alpha(h)x \ \forall h \in \mathfrak{h}_{\mathfrak{g}}\}$. We say that \mathfrak{g}_{α} is a root space if it is nonzero and α is nonzero.

Let $(\mathfrak{sl}(n))_{\alpha}$ be a root space, let $x \in (\mathfrak{sl}(n))_{\alpha}$. Write $\alpha = \sum_{i < n} a_i(\epsilon_{i+1} - \epsilon_i)$. Then, for all $h \in \mathfrak{h}_{\mathfrak{sl}(n)}$, we have $hx - xh = (\sum a_k(h_{kk} - h_{k+1,k+1}))x$. In particular, $(h_{ii} - h_{jj})x_{ij} = (\sum a_k(h_{k+1,k+1} - h_{kk}))x_{ij}$ for all h, i, j. This means that $x_{ij} = 0$ for all except one (i, j) pair, $i \neq j$, and that $\alpha = \epsilon_i - \epsilon_j$. This shows that there at most $n^2 - n$ roots.

Since $\mathfrak{sl}(n)_0 = \mathfrak{h}$ and since $\dim(\mathfrak{sl}(n)) - \dim(\mathfrak{h}_{\mathfrak{sl}(n)}) = n^2 - n$, all the previous elements are indeed roots. We have shown the following theorem.

Theorem 2.9. The Lie algebra $\mathfrak{sl}(n)$ has $n^2 - n$ roots, given by $\epsilon_i - \epsilon_j$ for $1 \le i, j \le n$ with $i \ne j$.

The abstract root system that these form is called A_{n-1} .

Theorem 2.10. The root system A_{n-1} has rank n - 1. In $\mathfrak{sl}(n)$, a basis of the root system is given by $\{\epsilon_{i+1} - \epsilon_i\}_{i < n}$.

Proof. Root systems of a Lie algebra span the dual of the Cartan subalgebra. In this case, $\dim(\mathfrak{h}_{\mathfrak{sl}(n)}) = n-1$, which shows the first statement. To show that the given set is a basis, we need to check (a) it is a basis of $\mathfrak{h}^*_{\mathfrak{sl}(n)}$, and (b) every root can be written as a nonnegative or a nonpositive linear combination of the basis. Since (a) is clear, we show (b). Consider $\alpha = \epsilon_i - \epsilon_j$ with $i \ge j$. Then, $\alpha = (\epsilon_i - \epsilon_{i-1}) + (\epsilon_{i-1} - \epsilon_{i-2}) + \cdots + (\epsilon_{j+1} - \epsilon_j)$, as desired. Proceed similarly if $i \le j$.

Hereafter, we let $\alpha_i = \epsilon_{i+1} - \epsilon_i$. The set of simple roots will be denoted by $\Delta(A_{n-1})$ or simply Δ , when the root system is clear from context.

Our previous computations also give a system of Lie algebra generators compatible with the root space decomposition. That is, we find $\mathfrak{sl}(n) = \mathfrak{n}_+ \oplus \mathfrak{h} \oplus \mathfrak{n}_-$, with \mathfrak{n}_+ generated by $\{e_\alpha\}_{\alpha \in \Delta}$, \mathfrak{h} spanned by $\{h_\alpha\}_{\alpha \in \Delta}$, and \mathfrak{n}_- generated by $\{f_\alpha\}_{\alpha \in \Delta}$, and with the additional properties that e_α spans $(\mathfrak{sp}(2n))_\alpha$ and $[e_\alpha, f_\alpha] = h_\alpha = \alpha^{\vee}$. Explicitly, these are given by

$$e_{\alpha_i} = E_{i,i+1}, \quad h_{\alpha_i} = E_{i,i} - E_{i+1,i+1}, \quad f_{\alpha_i} = e_{\alpha_i}^{\mathsf{t}} \quad \text{for } i = 1, ..., n-1.$$

By abuse of notation, we may write e_i , h_i , f_i instead of e_{α_i} , h_{α_i} , f_{α_i} .

Example 2.11. In $\mathfrak{sl}(3)$, we have

$$A_{2} = \bigwedge_{h_{2} \to a_{1}} \alpha_{1}$$

$$e_{1} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad e_{2} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad h_{1} = \begin{pmatrix} 1 & -1 & 0 \end{pmatrix},$$

$$h_{2} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad f_{2} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

CHAPTER 2. REPRESENTATION THEORY OF THE CLASSICAL LIE ALGEBRAS

The fundamental weights ω_i , i = 1, ..., n-1, are defined by $\langle \omega_i, \alpha_j^{\vee} \rangle = \delta_{ij}$. Writing $\omega_i = \sum_{k < n} a_{i,k} (\epsilon_{k+1} - \epsilon_k)$, we obtain

$$\delta_{ij} = \langle \omega_i, \alpha_j^{\vee} \rangle = \langle \sum_{k < n} a_{i,k} (\epsilon_{k+1} - \epsilon_k), E_{j+1,j+1} - E_{jj} \rangle = -a_{i,j-1} + 2a_{i,j} - a_{i,j+1} + 2a_{i,j+1} + 2a_{i,j} - a_{i,j+1} + 2a_{i,j+1} + 2a$$

We conclude that $\omega_i = \sum_{k=i+1}^n \epsilon_k$.

Theorem 2.12. There are bijections

$$\begin{cases} \text{highest weight f.d.} \\ \text{irreps. of $\mathfrak{sl}(n)$} \end{cases} & \longleftrightarrow & \begin{cases} \text{dominant integral} \\ \text{weights of $\mathfrak{h}_{\mathfrak{sl}(n)}$} \end{cases} & \longleftrightarrow & \begin{cases} \text{partitions} \\ \text{of length $\leq n-1$} \end{cases} \\ L(\lambda) & \longleftrightarrow & \lambda = \sum_{i=1}^n \lambda_i \epsilon_i & \mapsto & (\lambda_{n-1}, ..., \lambda_1). \end{cases}$$

Proof. We know that $\mathfrak{sl}(n)$ is a complex semisimple Lie algebra. This gives the first bijection [Str22]. To see the second bijection, consider $\lambda = \sum_{i=1}^{n-1} \lambda_i \epsilon_{i+1} = \sum_{i=1}^{n-1} \tilde{\lambda}_i \omega_i$ (which we can assume, since $\epsilon_1 = -\sum_{i=2}^{n} \epsilon_i$). By definition, $\mathbb{N} \ni \langle \lambda, \alpha_i^{\vee} \rangle = \tilde{\lambda}_i$. By the above formulas for ω_i , we then get $\sum_{i=1}^{n-1} \lambda_i \epsilon_{i+1} = \sum_{i=1}^{n-1} \tilde{\lambda}_i \sum_{k=i+1}^{n} \epsilon_k$. Therefore, $\lambda_i = \tilde{\lambda}_1 + \cdots + \tilde{\lambda}_i$, and in particular, $\lambda_{n-1} \ge \cdots \ge \lambda_1$.

The Cartan matrix of $\mathfrak{sl}(n)$ is given by $\{\langle \alpha_i, \alpha_j^{\vee} \rangle\}_{ij} = \{-\delta_{i,j-1} + 2\delta_{ij} - \delta_{i,j+1}\}_{ij}$. Its Dynkin diagram is the following.

$$\alpha_1 \quad \alpha_2 \cdots \alpha_{n-2} \quad \alpha_{n-1}$$

We conclude by explicitly describing how one may study the representation theory of $\mathfrak{gl}(n)$ from that of $\mathfrak{sl}(n)$. Classical proofs of this next theorem typically require some development of Lie group theory. We choose to omit this, and refer to [FH91] for details.

Theorem 2.13. Every highest weight finite dimensional irreducible representation of $\mathfrak{gl}(n)$ is of the form

$$L^{\mathfrak{gl}(n)}(\lambda_n,...,\lambda_1) = L^{\mathfrak{sl}(n)}(\lambda_n - \lambda_1,...,\lambda_2 - \lambda_1) \otimes \lambda_n \cdot \mathrm{tr},$$

with $\lambda_n \geq \cdots \geq \lambda_1$ (not necessarily nonnegative).

We say by convention that $\mathfrak{gl}(n)$ has the same root system as $\mathfrak{sl}(n)$. But the Cartan subalgebra $\mathfrak{h}_{\mathfrak{gl}(n)}$ of $\mathfrak{gl}(n)$ is one dimension larger than that of $\mathfrak{sl}(n)$; it is spanned by $\{\epsilon_i\}_{1 \le i \le n}$, and the fundamental weights are $\omega_i = \sum_{k=i+1}^n \epsilon_k$ for i = 0, ..., n - 1.

The even symplectic Lie algebras

A basis of $\mathfrak{sp}(2n)$ is given by the matrices $E_{ij} + (-1)^{\delta(i \le n)} E_{ij}^{\textcircled{0}}$ for $i \le 2n - j$ and the matrices $E_{i,2n+1-i}$ for i = 1, ..., 2n. Therefore, $\mathfrak{sp}(2n)$ has dimension $\frac{1}{2}2n(2n-1) + 2n = 2n^2 + n$.

Lemma 2.14. A Cartan subalgebra $\mathfrak{h}_{\mathfrak{sp}(2n)}$ of $\mathfrak{sp}(2n)$ is given by the span of $\{E_{ii} - E_{ii}^{\textcircled{t}}\}_{i \leq n}$.

Proof. Analogous to the proof of Lemma 2.8 for type A.

The dual space $\mathfrak{h}^*_{\mathfrak{sp}(2n)}$ has a basis given by $\epsilon_i : E_{jj} - E_{jj}^{(1)} \mapsto \delta_{ij}$, for i, j = 1, ..., n. Suppose $(\mathfrak{sp}(2n))_{\alpha}$ is a root space, and let $x \in (\mathfrak{sp}(2n))_{\alpha}$. We have $[h, x] = \alpha(h)x$ for all $h \in \mathfrak{h}_{\mathfrak{sp}(2n)}$, with $\alpha = \sum a_i \epsilon_i$. That is, $hx - xh = (\sum a_k h_{kk}) x$. In particular, $(h_{ii} - h_{jj})x_{ij} = (\sum a_k h_{kk}) x_{ij}$ for all h, i, j. This means that $x_{i,j} = 0 = x_{2n+1-i,2n+1-j}$ for all except one (i, j) pair, and that $\alpha = \pm \epsilon_i \pm \epsilon_j$. Again, because these are $2n^2$ elements and since dim $(\mathfrak{sp}(2n)) - \dim(\mathfrak{h}_{\mathfrak{sp}(2n)}) = 2n^2$, we get that these are all roots. We have shown the following theorem.

Theorem 2.15. The root system of $\mathfrak{sp}(2n)$ has $2n^2$ roots. As a set, it is given by $\{\pm 2\epsilon_i\}_{i \le n} \sqcup \{\pm \epsilon_i \pm \epsilon_j\}_{i < j \le n}$.

The abstract root system formed by these is called C_n .

Theorem 2.16. The root system C_n has rank n. In $\mathfrak{sp}(2n)$, a basis is $\{2\epsilon_1\} \sqcup \{\epsilon_{i+1} - \epsilon_i\}_{i < n}$.

Proof. The first statement follows from dim($\mathfrak{h}_{\mathfrak{sp}(2n)}$) = *n*. For the second one, we mostly proceed as in Theorem 2.10 for type A. To express $\pm 2\epsilon_i$ in terms of the basis, one may write $\pm 2\epsilon_i = \pm 2\epsilon_1 \pm 2(\epsilon_2 - \epsilon_1) \pm \cdots \pm 2(\epsilon_{i+1} - \epsilon_i)$.

Hereafter, we will denote the simple roots by $\alpha_0 := 2\epsilon_1$ and $\alpha_i := \epsilon_{i+1} - \epsilon_i$ for i = 1, ..., n - 1. The set of simple roots will be denoted by $\Delta(C_n)$ or simply Δ , when the root system is clear from context.

Again, our previous computations also give a system of Lie algebra generators compatible with the root space decomposition. Explicitly, these are given by

$$e_{\alpha_i} = E_{i+1,i} - E_{i+1,i}^{(t)}, \quad h_{\alpha_i} = E_{i+1,i+1} - E_{ii} + E_{ii}^{(t)} - E_{i+1,i+1}^{(t)}, \quad f_{\alpha_i} = e_{\alpha_i}^t \quad \text{for } i = 1, ..., n-1,$$
$$e_{\alpha_0} = E_{1,2n}, \quad h_{\alpha_0} = E_{11} - E_{11}^{(t)}, \quad f_{\alpha_0} = e_{\alpha_0}^t.$$

By abuse of notation, we may write e_i , h_i , f_i instead of e_{α_i} , h_{α_i} , f_{α_i} .

Example 2.17. In $\mathfrak{sp}(4)$, we have

$$C_{2} = \begin{pmatrix} \alpha_{1} & & \\ & e_{0} = \begin{pmatrix} & 0 & \\ & 0 & \\ &$$

We can also compute the fundamental weights. We have

$$\langle \omega_j, \alpha_i^{\vee} \rangle = \delta_{ij} = \langle \omega_j, E_{i+1,i+1} - E_{ii} + E_{ii}^{\textcircled{\text{C}}} - E_{i+1,i+1}^{\textcircled{\text{C}}} \rangle$$

which implies $\omega_i = \sum_{k=i+1}^n \epsilon_k$ for i = 0, ..., n - 1.

Theorem 2.18. There are bijections

$$\begin{cases} \text{highest weight f.d.} \\ \text{irreps. of } \mathfrak{sp}(2n) \end{cases} & \longleftrightarrow \quad \begin{cases} \text{dominant integral} \\ \text{weights of } \mathfrak{h}_{\mathfrak{sp}(2n)} \end{cases} & \leftrightarrow \quad \begin{cases} \text{partitions} \\ \text{of length} \le n \end{cases}, \\ L(\lambda) & \longleftrightarrow \quad \lambda = \sum \lambda_i \epsilon_i \qquad \mapsto \quad (\lambda_n, ..., \lambda_1). \end{cases}$$

Proof. We know that $\mathfrak{sp}(2n)$ is a complex semisimple Lie algebra. This gives the first bijection [Str22]. To see the second bijection, we consider $\lambda = \sum_{i=1}^{n} \lambda_i \epsilon_i = \sum_{i=0}^{n-1} \tilde{\lambda}_i \omega_i$. By definition, $\mathbb{N} \ni \langle \lambda, \alpha_i^{\vee} \rangle = \tilde{\lambda}_i$. By the above formulas for ω_i , we then get $\sum_{i=1}^{n} \lambda_i \epsilon_i = \sum_{i=0}^{n-1} \tilde{\lambda}_i \sum_{k=i+1}^{n} \epsilon_k$, and thus $\lambda_i = \tilde{\lambda}_0 + \cdots + \tilde{\lambda}_{i-1} \in \mathbb{N}$. In particular, $\lambda_n \ge \cdots \ge \lambda_0$.

In the next section, we will study the Weyl group of C_n . To prepare this discussion, we need the Cartan matrix of $\mathfrak{sp}(2n)$. Since we have explicit descriptions of all simple roots and coroots, we can simply compute. For i, j = 1, ..., n - 1, we have

$$\langle \alpha_i, \alpha_j^{\vee} \rangle = \langle \epsilon_{i+1} - \epsilon_i, E_{j+1,j+1} - E_{jj} + E_{jj}^{\textcircled{0}} - E_{j+1,j+1}^{\textcircled{0}} \rangle = -\delta_{i,j+1} - \delta_{i+1,j+1}$$

When either *i* or *j* are equal to 0, similar computations give $\langle \alpha_0, \alpha_j^{\vee} \rangle = -2\delta_{1,j}, \langle \alpha_i, \alpha_0^{\vee} \rangle = -\delta_{i,1}$, and of course, $\langle \alpha_0, \alpha_0^{\vee} \rangle = 2$.

Example 2.19. For instance, for *n* = 4, we have the following Cartan matrix:

$$\begin{pmatrix} 2 & -2 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}$$

In other words, the Dynkin diagram of $W(C_n)$ is the following.

$$\alpha_0 \quad \alpha_1 \quad \alpha_2 \quad \alpha_{n-2} \quad \alpha_{n-1}$$

The odd orthogonal Lie algebras

Let N = 2n + 1. A basis of $\mathfrak{so}(N)$ is given by the matrices $\{E_{ij} - E_{ij}^{\textcircled{1}} : i \leq N - j\}$, and thus $\mathfrak{so}(N)$ has dimension $\frac{1}{2}N(N-1) = 2n^2 + n$.

Lemma 2.20. A Cartan subalgebra $\mathfrak{h}_{\mathfrak{so}(N)}$ is given by span $\{E_{ii} - E_{ii}^{\textcircled{t}}\}_{i \leq n}$.

Proof. Analogous to the proof of Lemma 2.8 for type A.

In particular, note that $h_{n+1,n+1} = 0$ for all matrices in $\mathfrak{h}_{\mathfrak{so}(N)}$. The dual space $\mathfrak{h}^*_{\mathfrak{so}(N)}$ has a basis given by $\epsilon_i : E_{jj} - E_{jj}^{\oplus} \mapsto \delta_{ij}$, for i, j = 1, ..., n. Suppose $(\mathfrak{so}(N))_{\alpha}$ is a root space, and let $x \in (\mathfrak{so}(N))_{\alpha}$. We have $[h, x] = \alpha(h)x$ for all $h \in \mathfrak{h}_{\mathfrak{so}(N)}$, with $\alpha = \sum a_i \epsilon_i$. That is, $hx - xh = (\sum a_k h_{kk}) x$. In particular, $(h_{ii} - h_{jj})x_{ij} = (\sum a_k h_{kk}) x_{ij}$ for all h, i, j. This means that $x_{i,j} = 0 = x_{N+1-i,N+1-j}$ for all except one (i, j) pair, and that $\alpha = \pm \epsilon_i \pm \epsilon_j$. If j = n + 1, we also get $\alpha = \pm \epsilon_i$. Since dim $(\mathfrak{so}(N))$ – dim $(\mathfrak{h}_{\mathfrak{so}(N)}) = 2n^2$, all of these are indeed roots. We have shown the following theorem.

Theorem 2.21. The root system of $\mathfrak{so}(2n+1)$ has $2n^2$ roots and is given, as a set, by $\{\pm \epsilon_i\}_{i \le n} \sqcup \{\pm \epsilon_i \pm \epsilon_i\}_{i \le n}$ $\epsilon_i\}_{i < j \leq n}$.

The abstract root system given by these is called B_n .

Theorem 2.22. The root system B_n has rank n. In $\mathfrak{so}(2n+1)$, a basis is $\{\epsilon_1\} \sqcup \{\epsilon_{i+1} - \epsilon_i\}_{i \le n}$.

Proof. We proceed as in Theorem 2.16.

Hereafter, we will denote the simple roots by $\alpha_0 := \epsilon_1$ and $\alpha_i := \epsilon_{i+1} - \epsilon_i$ for i = 1, ..., n - 1. The set of simple roots will be denoted by $\Delta(B_n)$ or simply Δ , when the root system is clear from context.

We can also compute the fundamental weights. We identify the *i*th coroot α_i^{\vee} with $E_{i+1,i+1} - E_{ii}$ + $E_{ii}^{(\oplus)} - E_{i+1,i+1}^{(\oplus)} \text{ for } i = 1, ..., n-1 \text{ and with } 2E_{11} - 2E_{11}^{(\oplus)} \text{ for } i = 0. \text{ We have } \langle \omega_j, \alpha_i^{\vee} \rangle = \delta_{ij} \text{, which implies } \omega_i = \sum_{k=i+1}^n \epsilon_k \text{ for } i = 1, ..., n-1, \text{ and } \langle \omega_0, \alpha_j^{\vee} \rangle = \delta_{0j} \text{ implies } \omega_0 = \frac{1}{2} \sum_{k=1}^n \epsilon_k.$

Not every highest weight finite dimensional irreducible representations of $\mathfrak{so}(2n + 1)$ is indexed by a partition. We say $(\mu_1, ..., \mu_n)$ is a half-partition of length *n* if $\mu_1 \ge \cdots \ge \mu_n$ and $\mu_i \in \mathbb{Z}_{\ge 0} + \frac{1}{2}$ for all *i*.

Theorem 2.23. There are bijections

$$\begin{cases} \text{highest weight f.d.}\\ \text{irreps. of $\mathfrak{so}(2n+1)$} \end{cases} \longleftrightarrow \begin{cases} \text{dominant integral}\\ \text{weights of } \mathfrak{h}_{\mathfrak{so}(2n+1)$} \end{cases} \longleftrightarrow \begin{cases} \text{partitions}\\ \text{of length $\leq n$} \end{cases} \cup \begin{cases} \text{half-partitions}\\ \text{of length n} \end{cases}, \\ L(\lambda) \longleftrightarrow \lambda = \sum \lambda_i \epsilon_i \longmapsto (\lambda_n, ..., \lambda_1). \end{cases}$$

Proof. We know that $\mathfrak{so}(2n+1)$ is a complex semisimple Lie algebra. This gives the first bijection [Str22]. To see the second bijection, we consider $\lambda = \sum_{i=1}^{n} \lambda_i \epsilon_i = \sum_{i=0}^{n-1} \tilde{\lambda}_i \omega_i$. By definition, $\mathbb{N} \ni \langle \lambda, \alpha_i^{\vee} \rangle = \tilde{\lambda}_i$. By the above formulas for ω_i , we then get $\sum_{i=1}^{n} \lambda_i \epsilon_i = \frac{1}{2} \tilde{\lambda}_0 \sum_{i=1}^{n} \epsilon_i + \sum_{i\geq 1} \tilde{\lambda}_i \sum_{k\geq i+1} \epsilon_k$, and thus $\lambda_i = \frac{1}{2} \tilde{\lambda}_0 + \tilde{\lambda}_1 + \dots + \tilde{\lambda}_{i-1} \in \mathbb{N} + \frac{1}{2} \mathbb{N}$. Moreover, $\lambda_n \geq \dots \geq \lambda_1$.

The Cartan matrix of $\mathfrak{sp}(2n+1)$ is the transpose of the Cartan matrix of $\mathfrak{sp}(2n)$. In other words, the Dynkin diagram of $W(B_n)$ is the following.

$$\alpha_0 \quad \alpha_1 \quad \alpha_2 \quad \alpha_{n-2} \quad \alpha_{n-1}$$

The even orthogonal Lie algebras

Let N = 2n. A basis of $\mathfrak{so}(N)$ is given by the matrices $\{E_{ij} - E_{ij}^{\textcircled{1}} : i \leq N - j\}$, and thus $\mathfrak{so}(N)$ has dimension $\frac{1}{2}N(N-1) = 2n^2 - n$.

Lemma 2.24. A Cartan subalgebra $\mathfrak{h}_{\mathfrak{so}(N)}$ is given by $\operatorname{span}\{E_{ii} - E_{ii}^{\textcircled{t}}\}_{i \leq n}$.

Proof. Analogous to the proof of Lemma 2.8 for type A.

The dual space $\mathfrak{h}_{\mathfrak{so}(N)}^*$ has a basis given by $\epsilon_i : E_{jj} - E_{jj}^{\mathbb{C}} \mapsto \delta_{ij}$, for i, j = 1, ..., n. Suppose $(\mathfrak{so}(N))_{\alpha}$ is a root space, and let $x \in (\mathfrak{so}(N))_{\alpha}$. We have $[h, x] = \alpha(h)x$ for all $h \in \mathfrak{h}_{\mathfrak{so}(N)}$, with $\alpha = \sum a_i \epsilon_i$. That is, $hx - xh = (\sum a_k h_{kk}) x$. In particular, $(h_{ii} - h_{jj})x_{ij} = (\sum a_k h_{kk}) x_{ij}$ for all h, i, j. This means that $x_{i,j} = 0 = x_{N+1-i,N+1-j}$ for all except one (i, j) pair, and that $\alpha = \pm \epsilon_i \pm \epsilon_j$. Since $\dim(\mathfrak{so}(N)) - \dim(\mathfrak{h}_{\mathfrak{so}(N)}) = 2n^2$, all of these are indeed roots. We have shown the following theorem.

Theorem 2.25. The root system of $\mathfrak{so}(2n+1)$ has $2n^2-2n$ roots and is given, as a set, by $\{\pm\epsilon_i\pm\epsilon_j\}_{i< j\leq n}$.

The abstract root system given by these is called D_n .

Theorem 2.26. The root system D_n has rank n. In $\mathfrak{so}(2n)$, a basis is $\{\epsilon_2 + \epsilon_1\} \sqcup \{\epsilon_{i+1} - \epsilon_i\}_{i \le n}$.

Proof. We proceed as in Theorem 2.16.

Hereafter, we will denote the simple roots by $\alpha_0 := \epsilon_2 + \epsilon_1$ and $\alpha_i := \epsilon_{i+1} - \epsilon_i$ for i = 1, ..., n - 1. The set of simple roots will be denoted by $\Delta(D_n)$ or simply Δ , when the root system is clear from context.

We can also compute the fundamental weights. Let α_i^{\vee} be the *i*th coroot, which we can identify with $E_{i+1,i+1} - E_{ii} + E_{ii}^{\oplus} - E_{i+1,i+1}^{\oplus}$ for i = 1, ..., n-1, and with $E_{22} + E_{11} - E_{11}^{\oplus} + E_{22}^{\oplus}$ for i = 0. Then, $\langle \omega_j, \alpha_i^{\vee} \rangle = \delta_{ij}$ implies $\omega_i = \sum_{k=i+1}^n \epsilon_k$ for i = 2, ..., n-1, whereas for i = 0, 1 we get $\omega_0 = \frac{1}{2}(\epsilon_n + \cdots + \epsilon_2 + \epsilon_1)$ and $\omega_1 = \frac{1}{2}(\epsilon_n + \cdots + \epsilon_2 - \epsilon_1)$.

Again, not every highest weight finite dimensional irreducible representations of $\mathfrak{so}(2n + 1)$ is indexed by a partition. We need to introduce yet another object to index them: we say $(\mu_1, ..., \mu_n)$ is a signed (half-)partition if $(\mu_1, ..., \mu_{n-1}, |\mu_n|)$ is a (half-)partition. We let $\operatorname{sgn}(\mu) := \operatorname{sgn}(\mu_n)$ be the sign of μ .

Theorem 2.27. There are bijections

$$\begin{cases} \text{highest weight f.d.} \\ \text{irreps. of $\mathfrak{so}(2n)$} \end{cases} & \longleftrightarrow \quad \begin{cases} \text{dominant integral} \\ \text{weights of $\mathfrak{h}_{\mathfrak{so}(2n)}$} \end{cases} & \longleftrightarrow \quad \begin{cases} \text{signed partitions} \\ \text{of length $\leq n$} \end{cases} \cup \begin{cases} \text{signed half-partitions} \\ \text{of length n} \end{cases}; \\ L(\lambda) & \longleftrightarrow \quad \lambda = \sum \lambda_i \epsilon_i \quad \mapsto \quad (\lambda_n, ..., \lambda_1). \end{cases}$$

Proof. We know that $\mathfrak{so}(2n)$ is a complex semisimple Lie algebra. This gives the first bijection [Str22]. To see the second bijection, we consider $\lambda = \sum_{i=1}^{n} \lambda_i \epsilon_i = \sum_{i=0}^{n-1} \tilde{\lambda}_i \omega_i$. By definition, $\mathbb{N} \ni \langle \lambda, \alpha_i^{\vee} \rangle = \tilde{\lambda}_i$. By the above formulas for ω_i , we then get

$$\sum_{i=1}^n \lambda_i \epsilon_i = \tilde{\lambda}_0 \frac{1}{2} (\epsilon_n + \dots + \epsilon_2 + \epsilon_1) + \tilde{\lambda}_1 \frac{1}{2} (\epsilon_n + \dots + \epsilon_2 - \epsilon_1) + \sum_{i \ge 2} \tilde{\lambda}_i (\epsilon_n + \dots + \epsilon_{i+1}).$$

Therefore, $\lambda_1 = \frac{1}{2}(\tilde{\lambda}_0 - \tilde{\lambda}_1) \in \frac{1}{2}\mathbb{Z}$, $\lambda_2 = \frac{1}{2}(\tilde{\lambda}_0 + \tilde{\lambda}_1) \in \frac{1}{2}\mathbb{N}$, and $\lambda_i = \frac{1}{2}\tilde{\lambda}_0 + \frac{1}{2}\tilde{\lambda}_1 + \tilde{\lambda}_2 + \dots + \tilde{\lambda}_{i-1} \in \mathbb{N} + \frac{1}{2}\mathbb{N}$. Moreover, $\lambda_n \ge \dots \ge \lambda_2 \ge |\lambda_1| \ge 0$. Whether $\tilde{\lambda}_0$ and $\tilde{\lambda}_1$ are simultaneously odd determines if $(\lambda_n, ..., \lambda_2, |\lambda_1|)$ is a partition or a half-partition.

When computing the Cartan matrix of $\mathfrak{so}(2n)$, note that $\langle \alpha_0, \alpha_1^{\vee} \rangle = 0$. The Dynkin diagram of $W(D_n)$ is the following.

$$\alpha_0 \bullet \ \alpha_1 \bullet \ \alpha_2 \quad \alpha_3 \cdots \bullet \ \alpha_{n-2} \quad \alpha_{n-1} \bullet$$

2.2 The classical Weyl groups

Weyl groups are Coxeter groups that can therefore be described by Coxeter graphs. Simple Lie algebras are classified by a refinement of the Coxeter graph; the Dynkin diagram (which is a mixed graph, with directed and undirected edges, whose underlying undirected graph is a Coxeter graph). The Dynkin diagrams of interest to us were computed in the last section, and are displayed in Table 2.1.

Туре	Dynkin diagram	Lie algebra
$A_{n-1} \ (n \ge 2)$	• • • • • •	$\mathfrak{sl}(n)$
$B_n \ (n \ge 2)$		$\mathfrak{so}(2n+1)$
$C_n \ (n \ge 3)$	•	$\mathfrak{sp}(2n)$
$D_n \ (n \ge 4)$		$\mathfrak{so}(2n)$

Table 2.1: Dynkin diagrams of types A, B, C, D.

The three Coxeter groups that we read off these diagrams are

$$\begin{split} W(A_{n-1}) &= \langle s_1, ..., s_{n-1} \mid s_i^2 \; \forall i, (s_i s_{i+1})^3 \; \forall i \ge 1, (s_i s_j)^2 \; \forall |i-j| \ge 2 \rangle, \\ W(B_n) &= \langle s_0, ..., s_{n-1} \mid s_i^2 \; \forall i, (s_i s_{i+1})^3 \; \forall i \ge 1, (s_i s_j)^2 \; \forall |i-j| \ge 2, (s_0 s_1)^4 \rangle, \\ W(D_n) &= \langle s_0, ..., s_{n-1} \mid s_i^2 \; \forall i, (s_i s_{i+1})^3 \; \forall i \ge 1, (s_i s_j)^2 \; \forall \{i, j\} \neq \{0, 2\}, |i-j| \ge 2, (s_0 s_1)^2, (s_0 s_2)^3 \rangle. \end{split}$$

We have $W(C_n) = W(B_n)$. We also note that $W(A_{n-1})$ is just \mathbb{S}_n . Similarly, it is sometimes more useful to think of the Weyl group $W(B_n)$ in a different way (rather than by its presentation).

Definition-theorem 2.28. Let *G* and *H* be finite groups, which we see as permutation groups. That is, there are numbers *n* and *k* and actions $G \curvearrowright [n]$ and $H \curvearrowright [k]$. We define the wreath product $G \wr H$ of *G* and *H* as the group with underlying set $G \times H^n$ and multiplication given by

$$(g; h_1, ..., h_n) \cdot (g'; h'_1, ..., h'_n) = (gg'; h_{q'.1}h'_1, ..., h_{q'.n}h'_n).$$

It acts on $[n] \times [k]$ via $(g; h_1, ..., h_n).(i, j) = (g.i, h_i.j).$

Proof. It is a group: the identity element is (e; e, ..., e), and the inverse of an element $(g; h_1, ..., h_n)$ is given by $(g^{-1}; h_{q^{-1}.1}^{-1}, ..., h_{q^{-1}.n}^{-1})$. Indeed,

$$(g; h_1, ..., h_n) \cdot (g^{-1}; h_{g^{-1}.1}^{-1}, ..., h_{g^{-1}.n}^{-1}) = (gg^{-1}; h_{g^{-1}.1}h_{g^{-1}.1}^{-1}, ..., h_{g^{-1}.n}h_{g^{-1}.n}^{-1}) = (e; e, ..., e)$$

and

$$(g^{-1}; h^{-1}_{g^{-1}.1}, ..., h^{-1}_{g^{-1}.n}) \cdot (g; h_1, ..., h_n) = (g^{-1}g; h^{-1}_{gg^{-1}.1}h_1, ..., h^{-1}_{gg^{-1}.n}h_n) = (e; e, ..., e)$$

Finally, the action is well-defined;

$$(g; h_1, ..., h_n) \cdot (g'; h'_1, ..., h'_n).(i, j) = (g; h_1, ..., h_n).(g'.i, h'_i.j) = (gg'.i, h_{g'.i}h'_i.j) = (gg'; h_{g'.1}h'_1, ..., h_{g'.n}h'_n).(i, j).$$

Example 2.29. We will mostly consider the wreath product of \mathbb{S}_n and \mathbb{Z}_2 . We see \mathbb{Z}_2 as a permutation group acting on [2], or on the set $\{+, -\}$. Therefore, the underlying set of $\mathbb{S}_n \wr \mathbb{Z}_2$ is $\mathbb{S}_n \times (\mathbb{Z}_2)^n$, and the action on $[n] \times \{+, -\}$ is given by $(\sigma; \epsilon_1, ..., \epsilon_n).(\pm i) = \pm \epsilon_i \cdot \sigma(i)$. We identify the group $\mathbb{S}_n \wr \mathbb{Z}_2$ with the group of automorphisms of the set of leaves of the tree in Figure 2.30. A generating set of $\mathbb{S}_n \wr \mathbb{Z}_2$ is $\{(1 \ 1')\} \cup \{(i \ i + 1)(i' \ i + 1') : i = 1, ..., n - 1\}$.



Figure 2.30: A tree whose leaves have automorphism group $\mathbb{S}_n \wr \mathbb{Z}_2$.

Lemma 2.31. There is an isomorphism of groups $W(B_n) \cong \mathbb{S}_n \wr \mathbb{Z}_2$.

Proof. The map $\Psi : W(B_n) \to \mathbb{S}_n \wr \mathbb{Z}_2$ defined by $s_0 \mapsto (1 \ 1')$ and $s_i \mapsto (i \ i+1)(i' \ i+1')$ is a well-defined surjective group homomorphism.

Moreover, one can show the following claim.

CLAIM. $|W(B_n)/W(B_{n-1})| \le 2n$.

Proof of claim. Consider the following 2*n* cosets:

$$[e], [s_{n-1}], [s_{n-2}s_{n-1}], [s_{n-3}s_{n-2}s_{n-1}], ..., [(s_1 \cdots s_{n-2}s_{n-1})], \\ [s_0(s_1 \cdots s_{n-1})], [s_1s_0(s_1 \cdots s_{n-1})], ..., [(s_{n-1} \cdots s_1)s_0(s_1 \cdots s_{n-1})].$$

Consider the product of a generator with one of these cosets. For the sake of clarity, we follow an example, say $s_i[s_{n-2}s_{n-1}]$. If i = n - 2, then this simplifies to the coset $[s_{n-1}]$. If i = n - 3, then this becomes the coset $[s_{n-3}s_{n-2}s_{n-1}]$. Any other generator s_i , i = 0, ..., n - 4 commutes with the representative of the coset and gives $[s_{n-2}s_{n-1}]$. Finally, i = n - 1 also gives $[s_{n-2}s_{n-1}]$ by the braid relations.

With this analysis, we get a diagram

$$[e] \xrightarrow{s_{n-1}} [s_{n-1}] \xleftarrow{s_{n-2}} [s_{n-2}s_{n-1}] \xleftarrow{s_{n-3}} [s_{n-3}s_{n-2}s_{n-1}] \xleftarrow{s_{n-4}} \cdots \xleftarrow{s_1} [(s_1 \cdots s_{n-2}s_{n-1})] \xrightarrow{s_0} [s_0(s_1 \cdots s_{n-1})] \xleftarrow{s_1} [s_1s_0(s_1 \cdots s_{n-1})] \xleftarrow{s_2} \cdots \xleftarrow{s_{n-1}} [(s_{n-1} \cdots s_1)s_0(s_1 \cdots s_{n-1})]$$

in which the trivial action of generators were ignored. This shows that there are at most 2n cosets in the quotient and that they are among the cosets above.

By induction, $2^n n! \ge |W(B_n)| \ge |\mathbb{S}_n \wr \mathbb{Z}_2| = 2^n n!$. This shows that Ψ is an isomorphism.

We may represent a permutation $\sigma \in S_n$ diagrammatically as a collection of *n* strands going from *i* to $\sigma(i)$, respectively, for i = 1, ..., n. For instance, see the right diagram in Figure 2.33.

Another way of looking at the Weyl group of type B is through colored or decorated permutations. A 2-colored permutation of [n] is a pair $f = f_{\sigma} = (\sigma, C)$ where σ is a permutation $\sigma \in S_n$ and C is a sequence of colors; a subset of $\{+, -\}^n$. We say f_{σ} lifts σ . Diagramatically, we represent these colored permutations by attaching a dot to the negative strands. (Diagrams are composed as usual; two dots in the same strand cancel each other.)

Example 2.32. The colored permutation $f = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3' & 1' & 4 \end{pmatrix}$ lifts $\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 1 & 4 \end{pmatrix}$. See Figure 2.33.



Figure 2.33: The colored permutation f (on the left) lifts the permutation σ (on the right).

Lemma 2.34. The set of 2-colored permutations forms a group.

Proof. We identify $\{+, -\}$ with \mathbb{Z}_2 and $\{+, -\}^n$ with \mathbb{Z}_2^n . We can then multiply 2-color permutations via $(\sigma, C) \cdot (\tau, D) = (\sigma\tau, CD)$. The identity is (e, (+, ..., +)) and the inverse is given by $(\sigma^{-1}, -C)$.

Corollary 2.35. $W(B_n)$ is isomorphic to the group of 2-colored permutations on n letters.

A signed permutation on *n* letters is a permutation σ of $\{+, -\} \times [n] = \{-n, ..., -1\} \cup \{1, ..., n\}$ such that $\sigma(-i) = -\sigma(i)$.

Corollary 2.36. $W(B_n)$ is isomorphic to the group of signed permutations on n letters.

Proof. Signed permutations form a group as a subgroup of \mathbb{S}_{2n} . We construct an isomorphism from 2-colored permutations to signed permutations by letting the color indicate whether $\sigma(i)$ is in $\{-n, ..., -1\}$ or $\{1, ..., n\}$ for each $i \ge 1$. See Figure 2.37



Figure 2.37: The signed permutation on the center corresponds unequivocally to the colored permutation on the left and the element of $\mathbb{S}_n \wr \mathbb{Z}_2$ on the right.

Part II

Symmetric polynomials in types A, B, and C

Symmetric polynomials arise naturally in representation theory in a few different ways. Showing that the definitions agree is nontrivial. This will be the content of this chapter. More precisely, we will define symmetric polynomials in three ways: Lie theoretically (1), combinatorially (2), and as a specialization of symmetric functions of type A (3). Purely algebraic proofs of the equivalence between (1) and (3) are extracted from the literature [KT87, FH91]. To show that (2) and (3) agree, we follow [Oka89, SV16, FK97], where lattice path arguments were developed. Finally, to show the equivalence between (1) and (2), we appeal to crystal theory [KN94, HK02].

Chapter 3

The Lie theoretic definition

Fix a complex semisimple Lie algebra \mathfrak{g} with Cartan subalgebra \mathfrak{h} , root system Φ , and a basis Δ of the root system. Consider the integral weight lattice $X = \{\lambda \in \mathfrak{h}^* : \langle \lambda, \alpha^{\vee} \rangle \in \mathbb{Z} \text{ for all } \alpha \in \Delta\}$. We consider the group ring $\mathbb{C}[X]$ of X over \mathbb{C} . It is defined as the complex vector space with basis $\{x^{\lambda} : \lambda \in X\}$. It becomes a ring letting $x^{\lambda}x^{\mu} = x^{\lambda+\mu}$. The Weyl group W acts on $\mathbb{C}[X]$ by letting $\sigma.x^{\lambda} = x^{\sigma(\lambda)}$ for $\sigma \in W$ and $\lambda \in \mathfrak{h}^*$.

Given a finite dimensional representation *V* of \mathfrak{g} , we define the character of *V* as the dimensiongenerating function of its weight spaces. Explicitly, $\operatorname{ch} V = \sum_{\lambda \in \mathfrak{h}^*} (\dim V_\lambda) x^\lambda \in \mathbb{C}[X]$. Recall the following theorem [Hum72, FH91, Str22].

Theorem 3.1 (Weyl's character formula). Let $L(\lambda)$ be an irreducible finite-dimensional representation of a complex semisimple Lie algebra g. Fix a Cartan subalgebra \mathfrak{h} of g. Fix a root system Φ with basis Δ . Let W be the associated Weyl group. Let ρ be the half-sum of positive roots. Then,

$$\operatorname{ch}(L(\lambda)) = \frac{\sum_{\sigma \in W} (-1)^{l(\sigma)} \sigma . x^{\lambda + \rho}}{\sum_{\sigma \in W} (-1)^{l(\sigma)} \sigma . x^{\rho}}$$

as elements of the field of fractions of the group ring $\mathbb{C}[X]$.

With this, we can compute the irreducible characters for types A, B, C, and D [FH91]. Let $L(\lambda)$ be a finite dimensional highest weight irreducible representation of $\mathfrak{sl}(n)$, $\mathfrak{so}(2n + 1)$, $\mathfrak{sp}(2n)$, or $\mathfrak{so}(2n)$, respectively. Let $x_i := x^{\epsilon_i}$ for i = 1, ..., n. The character of $L(\lambda)$ is

(A).
$$\chi_{\lambda}^{\mathfrak{sl}(n)}(x_1,...,x_n) = \frac{\det(x_j^{\lambda_i+n-i})_{ij}}{\det(x_j^{n-i})_{ij}} \in \mathbb{C}[x_1,...,x_n]^{\mathbb{S}_n},$$

(B).
$$\chi_{\lambda}^{\mathfrak{so}(2n+1)}(x_1,...,x_n) = \frac{\det(x_j^{\lambda_i+n-l+1/2}-x_j^{-(\lambda_i+n-l+1/2}))_{ij}}{\det(x_j^{n-l+1/2}-x_j^{-(n-l+1/2)})_{ij}} \in \mathbb{C}(x_1^{1/2},...,x_n^{1/2})^{W(B_n)},$$

(C).
$$\chi_{\lambda}^{\mathfrak{sp}(2n)}(x_1, ..., x_n) = \frac{\det(x_j^{\lambda_i+n-i+1} - x_j^{-(\lambda_i+n-i+1)})_{ij}}{\det(x_j^{n-i+1} - x_j^{-(n-i+1)})_{ij}} \in \mathbb{C}(x_1, ..., x_n)^{W(C_n)}$$
, or

(D).
$$\chi_{\lambda}^{\mathfrak{so}(2n)}(x_1, ..., x_n) = \frac{\det(x_j^{\lambda_i + n - i} + x_j^{-(\lambda_i + n - i)})_{ij} - \det(x_j^{\lambda_i + n - i} - x_j^{-(\lambda_i + n - i)})_{ij}}{\det(x_j^{n - i} + x_j^{-(n - i)})_{ij}} \in \mathbb{C}(x_1, ..., x_n)^{W(D_n)}.$$

We refer to these characters as symmetric polynomials of types A, B, C, and D, respectively, and they constitute the main object of study in this work. Note that they are not polynomials in the usual sense of the word (except for symmetric polynomials of type A, although this is not clear from the above formula).

Proposition 3.2. The symmetric polynomial $\chi_{\lambda}^{\mathfrak{sl}(n)}$ is a polynomial in n variables.

For the purposes of this proof, we introduce the concept of skew-symmetry: a polynomial $p(x_1, ..., x_n)$ is skew-symmetric if $\sigma.p(x_1, ..., x_n) = \operatorname{sgn}(\sigma)p(x_1, ..., x_n)$ for any permutation $\sigma \in \mathbb{S}_n$. Note that the numerator and denominator of $\chi_{\lambda}^{\mathfrak{sl}(n)}$ are skew-symmetric, since the action of \mathbb{S}_n can be thought of as permuting the rows of the matrix prior to taking the determinant. Finally, we may refer to the matrix $(x_i^{n-i})_{ij}$ in the denominator as the Vandermonde matrix.

Proof. The proof is broken up into two claims, following [Sam17]. CLAIM 1. The determinant of $(x_i^{n-i})_{ij}$ is the Vandermonde polynomial,

$$\Delta(x_1,...,x_n) := \prod_{1 \le i < j \le n} (x_i - x_j).$$

CLAIM 2. The Vandermonde polynomial $\Delta(x_1, ..., x_n)$ divides every skew-symmetric polynomial in *n* variables.

The statement now follows.

Proof of Claim 2. Let $\sigma = (i \ j)$ be a transposition, let f be a skew-symmetric polynomial. Then $\sigma \cdot f = -f$ by skew-symmetry. On the other hand, if $x_i = x_j$, then $\sigma \cdot f$ and f agree. This means f vanishes when $x_i = x_j$, or that $(x_i - x_j)$ divides f. Since this is true for every transposition, we get that Δ divides f. \Box

Proof of Claim 1. The Vandermonde polynomial divides the determinant of the Vandermonde matrix by Claim 2. But since they are both polynomials of same degree (namely, $\binom{n}{2}$) and since they agree on at least one coefficient (the coefficient of $x_1^{n-1}x_2^{n-2}\cdots x_{n-1}$ for both is 1) they are the same polynomial.

Chapter 4

The combinatorial definition

Symmetric polynomials of type A are related to very rich combinatorial objects called tableaux. In this chapter, we define these objects and explore some analogues for types B and C. In these cases, there are several combinatorial models which are "natural" to consider, depending on the properties that one wants to study. We present here one of these models for type C, originally introduced in [Kin76], and then discuss its relation to some other models in Chapter 8. We only treat one model for type B in this work, introduced in [Sun90]. For other type B combinatorial models, see for instance [Pro94, She99, BZ22].

Let $[\lambda]$ be the set of cells of a partition $\lambda = (\lambda_1, ..., \lambda_l)$, defined as $\{(i, j) \in \mathbb{Z}_{>0}^2 : j \leq \lambda_i\}$. For instance,

$$\left| \square \right| = \{ (1,1), (1,2), (2,1), (3,1) \}.$$

We will sometimes identify λ and $[\lambda]$.

Let *X* be a countable set. A tableau of shape λ in the alphabet *X* is a function $T : [\lambda] \to X$ from the set of cells of λ to *X*. (For $[\mu] \subseteq [\lambda]$, we let a skew-tableau of shape λ/μ be a function $T : [\lambda] - [\mu] \to X$.)

If X is a totally ordered set, we say a (skew-)tableau is **semistandard** if T(i, j) < T(i + 1, j) and $T(i, j) \le T(i, j + 1)$ for all *i*, *j*. A semistandard (skew-)tableau is standard (on *n* letters) if the alphabet is $X = \{1 < 2 < \cdots < n\}$ and T is bijective.

Definition 4.1. Let λ be a partition. Let $\mathcal{A} := \{1 < 1' < 2 < 2' < \dots < n < n'\}$, and $\mathcal{A}_{\infty} := \{1 < 1' < 2 < 2' < \dots < n < n'\}$, and $\mathcal{A}_{\infty} := \{1 < 1' < 2 < 2' < \dots < n < n'\}$ be two ordered sets.

- (A) A **semistandard Young tableaux** (on *n* letters) of shape λ is a semistandard tableau of shape λ in the alphabet $X = \{1 < 2 < \cdots < n\}$. We let $SSYT_n(\lambda)$ be the set of such tableaux.
- (B) A **(Sundaram) orthogonal tableau** *T* (on *n* letters) of shape λ is a semistandard tableau of shape λ in the alphabet \mathcal{A}_{∞} such that
 - the co-restriction¹ of T to \mathcal{A} defines a symplectic tableau (see below), and
 - there is at most one ∞ per row; that is, if $T(i, j) = \infty$, then $T(i, k) \neq \infty$ for all $k \neq j$.

We let $SOT_n(\lambda)$ be the set of such tableaux.

(C) A **(King) symplectic tableau** T (on n letters) of shape λ is a semistandard tableau of shape λ in the alphabet \mathcal{A} such that $T(i, j) \ge i$ for all $(i, j) \in [\lambda]$. We let $\text{KSpT}_n(\lambda)$ be the set of such tableaux.

¹The co-restriction of a map $f : A \to B$ to a subset $C \subset B$ is defined to be the restriction of f to $f^{-1}(C)$.

The weight of a tableau $T : [\lambda] \to X$ is the monomial $x^T = \prod_{a \in X} x_{T^{-1}(a)} \in \mathbb{C}[x_i : i \in X]$. For the alphabets in the above definition, we take the conventions $x_{i'} = x_i^{-1}$ and $x_{\infty} = 1$. That is, the weights of semistandard Young tableaux are polynomials in $\mathbb{C}[x_1, ..., x_n]$, whereas the weights of orthogonal and symplectic tableaux live in

$$\mathbb{C}[x_1, x_{1'}, ..., x_n, x_{n'}, x_{\infty}]/\langle x_1 x_{1'} - 1, ..., x_n x_{n'} - 1, x_{\infty} - 1 \rangle = \mathbb{C}[x_1^{\pm}, ..., x_n^{\pm}]$$

Example 4.2. Here are a semistandard Young tableau, an orthogonal tableau, and a symplectic tableau of shape $(3^2, 2)$. Below, their weights.



Definition 4.3. The Schur polynomial s_{λ} , the orthogonal polynomial o_{λ} , and the symplectic polynomial $s_{p_{\lambda}}$ on *n* letters and of shape λ are defined as the power series $\sum x^{T}$, where the sums range over semistandard Young tableaux, orthogonal tableaux, and symplectic tableaux on *n* letters of shape λ , respectively. Explicitly,

$$s_{\lambda}(x_1, ..., x_n) = \sum_{T \in \text{SSYT}_n(\lambda)} x^T, \quad o_{\lambda}(x_1, ..., x_n) = \sum_{T \in \text{SOT}_n(\lambda)} x^T, \quad \text{and} \quad sp_{\lambda}(x_1, ..., x_n) = \sum_{T \in \text{KSpT}_n(\lambda)} x^T$$

Note 4.4. The above combinatorics for type B will only model the characters of spinless irreducible representations (those indexed by partitions, rather than half-partitions). A richer combinatorial model arises from crystal theory [KN94, HK02, BS17].

Note 4.5. The definition of orthogonal tableaux gives a bijective map $SOT_n(\lambda) \rightarrow \bigcup_{\mu} KSpT_n(\mu)$ where μ ranges over the partitions which may be formed from $[\lambda]$ by removing some cells, at most one per row. Explicitly, the map is given by co-restriction to the alphabet \mathcal{A} . There is also an injective map $KSpT_n(\lambda) \rightarrow SSYT_{2n}(\lambda)$ given by post-composition with the order-preserving map $\mathcal{A} \rightarrow [2n]$. Altogether, these give an injective map $SOT_n(\lambda) \rightarrow \bigcup_{\mu} SSYT_{2n}(\mu)$, with μ as above.

Immediately from Proposition 1.11, we get the following result.

Proposition 4.6. If $\lambda = (k)$ is a row partition (a partition of length 1), then $s_{\lambda} = h_{\lambda} = h_k$; if $\lambda = (1^k)$ is a column partition (the transpose of a row partition), then $s_{\lambda} = e_{\lambda'} = e_k$.

Analogously, we have the following result.

Proposition 4.7. If $\lambda = (k)$ is a row partition, then

1.
$$sp_{\lambda}(x_1, ..., x_n) = h_k(x_1, ..., x_n, x_1^{-1}, ..., x_n^{-1})$$
, and

2.
$$o_{\lambda}(x_1, ..., x_n) = h_k(x_1, ..., x_n, x_1^{-1}, ..., x_n^{-1}) + h_{k-1}(x_1, ..., x_n, x_1^{-1}, ..., x_n^{-1})$$

It is not obvious from the definitions, however, that these (Laurent) polynomials are symmetric for any λ . The next classical proof gives well-definedness (for type A).

Proposition 4.8. Schur polynomials on *n* letters are $W(A_{n-1})$ -symmetric.

Proof. We present a classical proof due to Bender and Knuth [BK72].

Fix a partition λ . Recall that $W(A_{n-1}) \cong \mathbb{S}_n$. Therefore, it is enough to check that the set $\{x^T : T \in SSYT_n(\lambda)\}$ of semistandard Young tableaux in *n* letters is invariant by any simple transposition (*i i* + 1), i = 1, ..., n - 1. Moreover, it is enough to show that the action of a given transposition s_i can be lifted

to an involutive action on $SSYT_n(\lambda)$. The existence of such an involution shows, in particular, that the subset $\{T : x^T = x^{\alpha}\}$ has the same cardinality as the subset $\{T : x^T = s_i \cdot x^{\alpha}\}$ for each given weight x^{α} , as desired.

Fix an integer $i \in [n-1]$. Given a (semistandard Young) tableau T, we want to produce a new tableau P with $m_i(T) = m_{i+1}(P)$ and $m_{i+1}(T) = m_i(P)$, and with $m_j(T) = m_j(P)$ for every $j \neq i, i+1$. Consider the (skew-)subtableau $T^{-1}(\{i, i+1\})$. See Figure 4.9 for an example.



Figure 4.9: Let i = 3. This is $T^{-1}(\{3, 4\})$ of a tableau *T*. Highlighted, mutable entries.

We say an entry is *frozen* if it belongs to an $\{i, i+1\}$ -vertical domino. Call every other entry *mutable*. In Figure 4.9, mutable entries were highlighted. In a given row, mutable entries form a word $i^a(i+1)^b$. We construct T' from T by changing each of these words for $i^b(i+1)^a$. See Figure 4.10.

1	1	1	1	2	2	3	3	3	3	4	4	4		1	1	1	1	2	2	3	3	3	3	3	3	4
2	2	2	3	3	4	4	4	4					\mapsto	2	2	2	3	3	4	4	4	4				
3	3	3	4											4	4	4	4									

Figure 4.10: In each row, we change the mutable word $3^{a}4^{b}$ for $3^{b}4^{a}$.

This is clearly well-defined and an involution. In particular, it defines a bijection between the set of semistandard Young tableaux with weight x^{α} and weight $(i \ i + 1).x^{\alpha}$ for each α , as desired.

The map defined in the above proof is referred to as the *i*th (type A) Bender–Knuth involution. We denote it by BK_i^A .

Note 4.11. Type A Bender–Knuth involutions do not verify braid relations. That is, they do not induce an action of \mathbb{S}_n on the set $SSYT_n(\lambda)$ of semistandard Young tableaux of shape λ on n letters. For instance,

$$BK_1^A BK_2^A BK_1^A \boxed{\begin{array}{c}1 & 2 & 3\\ 2 & 3\end{array}} = \boxed{\begin{array}{c}1 & 1 & 2\\ 2 & 3\end{array}} \neq \boxed{\begin{array}{c}1 & 1 & 3\\ 2 & 2\end{array}} = BK_2^A BK_1^A BK_2^A \boxed{\begin{array}{c}1 & 2 & 3\\ 2 & 3\end{array}}$$

One may define an \mathbb{S}_n action on $SSYT_n(\lambda)$ using crystal operators. Indeed, a crystal basis for the irreducible representation $L(\lambda)$ of $\mathfrak{sl}(n)$ may be identified with $SSYT_n(\lambda)$ (see Theorem 9.1, or [KN94, HK02]). Then, suitable compositions of crystal operators induce an action of the Weyl group on the crystal basis.

The action of each generator *s_i* is usually referred to as a crystal reflection or a Lascoux–Schützenberger involution. A combinatorial description of these reflections provides an alternative proof of well-definedness.

Before showing well-definedness for types B and C, we analyze these involutions from another perspective.

Chapter 5

Pattern combinatorics

A very intimately related notion to tableaux is that of Gelfand–Tsetlin patterns (or GT patterns) [GT50, Sta99].

Definition 5.1. A **Gelfand–Tsetlin pattern** x is a triangular tuple of non-negative integers, $x = (x^{(n)}, ..., x^{(1)})$ with $x^{(k)} = (x_{1k}, ..., x_{kk})$ for $k \in [n]$, subject to the local inequalities of Figure 5.2, whenever these make sense. (If represented as a triangular array as in Figure 5.2, the local inequalities express that the pattern must be weakly decreasing along SW-NE diagonals and NW-SE diagonals.) We say $x^{(n)}, ..., x^{(1)}$ are the rows of x. Note that each row is a partition. We call $x^{(n)}$ the top row of the pattern.

Let $GT_n(\lambda)$ be the set of Gelfand–Tsetlin patterns with *n* rows and top row λ .



Figure 5.2: Left: the arrangement of a GT pattern of size 4. Right: the local inequalities.

Note 5.3. Given a Gelfand–Tsetlin pattern x, it will sometimes be useful to let $x_{ij} := \infty$ for all i < 1 and $j \ge 0$, and to let $x_{ij} := 0$ for all $i > j \ge 0$. In this manner, for instance, the local inequalities are always well-defined in rows 1, ..., n - 1. (Here, the symbol ∞ is taken to be either a number $N \gg 1$ or the formal neutral element with respect to min.)



Proposition 5.4. There is a bijection between the sets $SSYT_n(\lambda)$ and $GT_n(\lambda)$, by letting $x^{(k)}$ be defined as the shape of the co-restriction of T to [k]; i.e., $x^{(k)}$ is the shape of $T^{-1}[k]$.

CHAPTER 5. PATTERN COMBINATORICS

Example 5.5. Let $\lambda = (3, 2)$ and let n = 3. The following semistandard Young tableau corresponds to the following Gelfand–Tsetlin pattern:



Proof. Under the proposed map, $x_{i,j}$ counts the number of entries less or equal to j in the *i*th row of the tableau. Row i of the tableau is weakly increasing if and only if $x_{i1} \le x_{i2} \le \cdots \le x_{in}$.

The tableau is then semistandard if and only if $T^{-1}(k)$ is a horizontal strip for each k, i.e., contains no vertical dominoes. But to contain a domino in the *j*th column between rows *i* and *i* + 1 is exactly the condition $x_{i,j} > x_{i+1,j}$.

Under the above bijection, the Bender–Knuth involution BK_j^A translates to the following map of GT patterns [BK95]: it only affects the entries in the *j*th row $x^{(j)}$, and it sends

$$x_{i,j}$$
 to $\min\{x_{i,j+1}, x_{i-1,j-1}\} + \max\{x_{i+1,j+1}, x_{i,j-1}\} - x_{i,j},$

where the minimum and maximum simply ignore non-existing entries. (Alternatively, we could follow the convention from Note 5.3.) Indeed, that only the *j*th row is affected is clear. (This means that only $T^{-1}(j)$ and $T^{-1}(j + 1)$ change.) Now we ask, in the *i*th row of the tableau, where does the mutable $\{j, j + 1\}$ -word end? Either at the $x_{i,j+1}$ th entry, if there are no $\{j, j + 1\}$ -vertical dominoes between rows *i* and *i* + 1, or at the $x_{i-1,j-1}$ th entry, otherwise. Altogether, at entry number min $\{x_{i,j+1}, x_{i-1,j-1}\}$. Similarly, the word starts at entry number max $\{x_{i+1,j+1}, x_{i,j-1}\}$. The formula is now clear.

Example 5.6. Let i = 3. We bring back our running example from Figures 4.9 and 4.10. Let us focus on the second row of the tableau. The mutable $\{3, 4\}$ -word in this row is highlighted in the diagram below.

1	1	1	1	2	2	3	3	3	3	4	4	4
2	2	2	3	3	4	4	4	4				
3	3	3	4				-					

To compute its end, see the next diagram: the word ends at the minimum between 6 (the number of colored boxes in row 1) and 9 (the number of colored boxes in row 2).

1	1	1	1	2	2	3	3	3	3	4	4	4
2	2	2	3	3	4	4	4	4				
3	3	3	4									

Similarly, to compute its start, see the final diagram: it starts at the maximum of 3 (the number of colored boxes in row 2) and 4 (the number of colored boxes in row 3).

1	1	1	1	2	2	3	3	3	3	4	4	4
2	2	2	3	3	4	4	4	4				
3	3	3	4									

More generally, given a poset *P*, consider the set $\mathbb{Z}^P = \{f : P \to \mathbb{Z}\}$ of \mathbb{Z} -labellings of *P*. We define a map $T_v : \mathbb{Z}^P \to \mathbb{Z}^P$ by letting $T_v f$ be defined as f in $P - \{v\}$ and sending

$$v$$
 to $\min\{f(u) : u \le v\} + \max\{f(u) : v \le u\} - f(v),$

where u < v refers to v covering u. The map T_v is called the v-toggle. It is easy to see that the composition of toggles on mutually non-covering vertices commutes. We recover the above definition by considering the poset induced by the local inequalities given in Figure 5.2. Now, the *j*th Bender–Knuth involution is the composition of the v-toggles on the elements of row j of the pattern, in any order.

Definition 5.7. A (King) symplectic pattern is a Gelfand–Tsetlin pattern in which $x_{ij} = 0$ whenever 2i > j (see [Kin76]). (Symplectic patterns are thus "half-triangular" arrays.) We let $\text{KSpP}_n(\lambda)$ be the set of (King) symplectic patterns with 2n rows and top row λ .

A **(Sundaram) orthogonal pattern** is a symplectic pattern in which top row entries might be circled. Let *x* be a Sundaram orthogonal pattern on *N* rows. For each entry x_{iN} in the top row, let $\lambda_i := x_{iN} + 1$ if the entry is circled and $\lambda_i := x_{iN}$ otherwise. We call $\lambda = (\lambda_1, ..., \lambda_N)$ the shape of the pattern. We let SOP_{*n*}(λ) be the set of Sundaram's orthogonal patterns with 2*n* rows and shape λ .

Note 5.8. The maps from Note 4.5 together with the bijection from Proposition 5.4 give bijections $\text{KSpT}_n(\lambda) \leftrightarrow \text{KSpP}_n(\lambda)$ and $\text{SOT}_n(\lambda) \leftrightarrow \text{SOP}_n(\lambda)$. For instance, letting $\lambda = (3, 2)$, we get the following correspondences.

Proposition 5.9. Symplectic polynomials in *n* letters are $W(C_n)$ -symmetric.

Note 5.10. The following is the first combinatorial proof as far as we are aware. We define a type C analogue for Bender–Knuth involutions. This was done already in [Sun86], but the proof is incomplete. It is also cited erroneously in [Catalog]. See the discussion in MO:362997.

Proof. Recall from Section 2.2 that the group $W(C_n)$ is generated by the transpositions (i i') and the permutations (i i+1)(i' i+1').

The symplectic tableaux in $\text{KSpT}_n(\lambda)$ are invariant under $(i \ i')$ by type A Bender–Knuth involutions. To make this precise, we consider the injective map from Note 4.5 composed with the (2i - 1)st type A Bender–Knuth involution. Since both i and i' can each appear on exactly the same rows (the first i rows, more precisely), the resulting tableau is symplectic. (In this proof, we will denote type A Bender–Knuth involutions by the transposition they induce on $\mathcal{A} = \{1 < 1' < \cdots < n < n'\}$.)

To show that they are invariant under (i i+1)(i' i+1'), we write this permutation as a product of simple transpositions; (i' i+1)(i+1')(i i')(i' i+1). For each of these, we perform a type A Bender–Knuth involution.

$$T_0 \xrightarrow{(i' \ i+1)} T_1 \xrightarrow{(i \ i')} T_2 \xrightarrow{(i+1 \ i+1')} T_3 \xrightarrow{(i' \ i+1)} T_4.$$
(5.11)

More precisely, using Note 4.5 again, the maps above are BK_{2i}^A , BK_{2i-1}^A , BK_{2i+1}^A , and BK_{2i}^A , respectively. However, T_4 needs not be symplectic: we might find an instance of *i'* in row *i* + 1. (Finding an instance of *i* in row *i* + 1 would contradict T_4 being semistandard.)

CLAIM. If $T_4(i + 1, j) = i'$ for some *j*, then $T_4(i, j) = i$.

Proof of claim. Since $T_4(i + 1, j)$ is an element of $\{i, i', i + 1, i + 1'\}$, then also $T_0(i + 1, j)$ is. Indeed, the composite in Equation (5.11) only affects these entries.

Since T_0 is symplectic, $T_0(r, c) \ge r$ for all $(r, c) \in [\lambda]$. In particular, $T_0(i, j) \ge i$.

Altogether, using that T_0 is semistandard, we get that $T_0(i, j)$ is in $\{i, i', i+1, i+1'\}$, and thus $T_4(i, j)$ too.

Since T_4 is semistandard (because Bender–Knuth involutions are well-defined on semistandard Young tableaux), we have $T_4(i, j) < T_4(i + 1, j) = i'$. We conclude $T_4(i, j) = i$.

We have shown that, if T_4 is not symplectic, then there are some $\{i, i'\}$ -vertical dominoes between rows *i* and *i* + 1. So we compose with a final map $T_4 \mapsto T_5$, which changes every such domino for a $\{i + 1, i + 1'\}$ -vertical domino and then resorts both rows as to make them weakly increasing again. We refer to this map as "rectification".

CHAPTER 5. PATTERN COMBINATORICS

This procedure always gives a symplectic tableau T_5 of weight $(i \ i + 1)(i' \ i + 1') x^{T_0}$, as desired. We denote the composite map as $BK_i^{\mathbb{C}}$, and call it a type C Bender–Knuth involution. Explicitly (using Note 4.5 again),

$$BK_i^{C} := \text{rectification} \circ BK_{2i}^{A} \circ BK_{2i+1}^{A} \circ BK_{2i-1}^{A} \circ BK_{2i}^{A}$$

To show that this is an involution, we switch to the symplectic pattern model. The type A Bender– Knuth involution BK_j^A for Gelfand–Tsetlin patterns is described above. The last step of our proposed map, rectification, translates to the map of Gelfand–Tsetlin patterns

as to make the pattern symplectic. Explicitly, by our analysis above, the pattern *x* corresponding to T_4 (see Eq. (5.11)) may not be symplectic. By the claim, this would imply that $x_{i+1,2i}$ is non-zero. In this case, changing every $\{i, i'\}$ -vertical domino between rows *i* and *i* + 1 of T_4 corresponds to substracting $x_{i+1,2i}$ from the entries $x_{i+1,2i}$, $x_{i+2,2i+1}$, $x_{i,2i}$ and $x_{i,2i-1}$, in particular rendering the pattern symplectic.

These formulas allow one to write, in principle, the entries of $(BK_i^{\mathbb{C}})^2(x)$ as tropical rational functions in the entries of x (that is, expressions involving max, min, + and –). These formulas, however, quickly become untracktable. We will see how to bypass this in what follows.

We note, given that the rectification map is fairly localized, that most of the entries of $(BK_i^C)^2(x)$ may immediately be seen to agree with those of x, since type A Bender–Knuth involutions are involutions and since BK_{2i-1}^A commutes with BK_{2i+1}^A . But some entries of $(BK_i^C)^2(x)$ are affected by the rectification map. A close inspection of our composite reveals that only the last two non-zero entries in rows 2i + 1, 2i and 2i - 1 of the pattern are affected. So it is enough to show that BK_2^C is an involution on a generic pattern of size 6.

Finally, consider the composite

$$BK_4^A \circ BK_5^A \circ BK_3^A \circ BK_4^A \circ BK_4^A \circ BK_5^A \circ BK_3^A \circ BK_4^A$$

which is the identity since BK_5^A and BK_3^A commute. This composite was obtained from $(BK_2^C)^2$ by ignoring the rectification maps. Our strategy to show that $(BK_2^C)^2$ is the identity is to compare it with the above composite. This comparison reveals that the first rectification map introduces an "error" that is later canceled by the second rectification map. We leave the details of this computation to Appendix A.

Example 5.12. Let i = 2. We illustrate the analogue of the *i*th Bender–Knuth involution for type C as the following composite:

1	2	2'	(2' 3)	1	2	3	(2.21)	1	2	3	(3.37)	1	2	3	(2/3)	1	2	2'	rootification	1	2	2'
3	3	3'	$\xrightarrow{(2 \ 3)}$	2'	2'	3'	\mapsto	2	2'	3'	\mapsto	2	2'	3'	\mapsto	2	2'	3'		2'	3	3'
3'	3'			3'	3'			3'	3'			3	3			2'	3			3	3'	

Each of the four first maps are type A Bender–Knuth involutions, and the last map rectifies the tableau by getting rid of the $\{2, 2'\}$ -vertical domino between the second and third rows.

Note 5.13. During the development of the thesis, another proof of the above map being an involution was proposed. This proof consists on two parts. The first part is to *detropicalize* the expression for Bender–Knuth involutions on GT patterns. Detropicalization refers to a map of semifields that can be described in layman terms as follows: starting with an expression in terms of "max"s and sums, change each instance

CHAPTER 5. PATTERN COMBINATORICS

of "max" for a sum and each sum for a product. The result will be a rational function. For instance, the detropicalization of the v-toggle is the map

$$v \mapsto \frac{\sum_{v < u} f(u)}{\left(\sum_{u < v} \frac{1}{f(u)}\right) \cdot f(v)}.$$

This allows one, in turn, to detropicalize Bender–Knuth involutions. Once these Bender–Knuth involutions and the rectification map are detropicalized, the equations of the composite were fed into SageMath, which checked that the detropicalization of the composite on a pattern of order 6 is an involution (as a rational function).

A second part of this argument is to show that the order of our original map must coincide with the order of its detropicalization. Since detropicalization and composition do not commute in general, this is not a trivial statement. We were not able to show this. See [Roé13] or [GR16] for more instances of this general type of problems.

Note 5.14. Our defined type C Bender–Knuth involutions do not verify the braid relations of the Weyl group of type C. For instance, $s_1s_2s_1 = s_2s_1s_2$, but

See also Note 4.11. In the literature, combinatorial descriptions of type C Lascoux–Schützen-berger involutions are available (see e.g. [San21]), but defined on a different combinatorial model: Kashiwara's tableaux. Showing that there is a weight-preserving bijection between King's and Kashiwara's tableaux would then give an alternative proof of Proposition 5.9. See Chapter 8 for such a bijection.

Note also that our analogues for Bender–Knuth involutions do not agree with Lascoux–Schützenberger's (as expected) even for column or row tableaux. See Appendix B.

Corollary 5.15. Orthogonal polynomials in n letters are $W(B_n)$ -symmetric.

Proof. We have $W(B_n) = W(C_n)$. Consider the bijection $SOT_n(\lambda) \to \bigcup_{\mu} KSpT_n(\mu)$ from Note 4.5. It is weight-preserving. We can then write

$$o_{\lambda} = \sum_{T \in \text{SOT}_n(\lambda)} x^T = \sum_{\mu} \sum_{T \in \text{KSpT}_n(\mu)} x^T = \sum_{\mu} sp_{\mu}.$$

The result falls now from Proposition 5.9 (the analogous result for type C).

Chapter 6

The algebraic definition (Jacobi–Trudi determinants)

Another classic definition of Schur polynomials is through Jacobi–Trudi's formulas. It defines Schur polynomials in terms of another basis of the algebra of symmetric polynomials, discussed in Chapter 1. We present it in the form of a theorem.

Theorem 6.1 (Jacobi–Trudi [Jac41, Tru64, Sag01, Sta99]). Let $X = x_1 + \cdots + x_n$ be an alphabet. We have

$$s_{\lambda}(X) = \det \left(h_{\lambda_i - i + j}(X) \right)_{1 \le i, j \le n} \in \Lambda_n(X),$$

where $h_0 = 1$ and $h_{-k} = 0$ for all $k \ge 1$.

We will provide a combinatorial proof of this theorem, via the lattice path method deve-loped in [Lin73, GV85].

Lemma 6.2 (Lindström, Gessel–Viennot [Lin73, GV85]). Let $G = (V, \vec{E}, w)$ be a weighted digraph. Let $A = \{a_1, ..., a_n\}$ and $B = \{b_1, ..., b_n\}$ be two distinguished sets of nodes. Let $W(a_i, b_j)$ be the weighted sum of the set of paths from a_i to b_j . Assume furthermore that for any n paths $P_1, ..., P_n$ with $P_i : a_i \rightarrow b_{\sigma(i)}$ for some $\sigma \in S_n$, if the paths are pairwise nonintersecting, then $\sigma = id$. Then,

$$\det \left(W(a_i, b_j) \right)_{1 \le i, j \le n} = \sum_{\substack{(P_1, \dots, P_n) \\ P_i: a_i \to b_i \\ nonintersecting}} w(P_1) \cdots w(P_n).$$

Proof. We begin by expanding the determinant,

$$\det \left(W(a_i, b_j) \right)_{1 \le i, j \le n} = \sum_{\sigma \in \mathbb{S}_n} \operatorname{sgn}(\sigma) \prod_{i \in [n]} W(a_i, b_{\sigma(i)})$$
$$= \sum_{\sigma \in \mathbb{S}_n} \operatorname{sgn}(\sigma) \prod_{i \in [n]} \sum_{P: a_i \to b_{\sigma(i)}} w(P) = \sum_{\sigma \in \mathbb{S}_n} \operatorname{sgn}(\sigma) \sum_{\substack{(P_1, \dots, P_n) \\ P_i: a_i \to b_{\sigma(i)}}} w(P_1) \cdots w(P_n).$$

The result is now shown via a weight-preserving sign-reversing involution Φ : if we define such a map on the set of intersecting tuples of *n* paths, these will therefore cancel out in the sum. Only nonintersecting *n*-tuples survive, and these are all summed with a positive sign, since by hypothesis they induce the identity permutation.

To define the involution Φ , let $(P_1, ..., P_n)$ be intersecting and let (i, j) be the first tuple (with respect to the lexicographic order) such that P_i and P_j intersect. Let x be the first node in which they intersect. We write schematically $P_i : a_i \to x \to b_{\sigma(i)}$ and $P_j : a_j \to x \to b_{\sigma(j)}$. We let $\Phi((P_1, ..., P_n))$ be the tuple $(P_1, ..., P'_i, ..., P'_j, ..., P_n)$, where $P'_i : a_i \to x \to b_{\sigma(j)}$ is constructed by following P_i from a_i to x and P_j from x to $b_{\sigma(j)}$, and $P'_j : a_j \to x \to b_{\sigma(i)}$ is constructed similarly. See Figure 6.3.



Figure 6.3: The involution Φ sends P_i and P_j on the left to P'_i and P'_j on the right. (The paths on the right also intersect at *x*.)

That this is an involution is clear, thanks to the lexicographic order. It is well defined on the set of intersecting tuples of paths. It is weight-preserving, since the set of edges involved in $(P_1, ..., P_n)$ and in $\Phi((P_1, ..., P_n))$ coincide. It is sign-reversing, since the permutation associated with $\Phi((P_1, ..., P_n))$ is $\sigma \circ (i j)$.

Proof of Thm. 6.1. Let $\lambda = (\lambda_1, ..., \lambda_l)$ be a partition. Define a weighted lattice digraph $L = (V, \vec{E}, w)$ with vertex set $[N] \times [n]$ (for $N \gg 1$) and arrows going from each node (i, j) to the one immediately east (i + 1, j) and to the one immediately north (i, j + 1). The weight of a vertical arrow is 1 and the weight of a horizontal arrow is x_i if it is at height j.

Define two distinguished sets of nodes; $A = \{a_k := (k, 1) : 1 \le k \le l\}$ and $B = \{b_k := (k + \lambda_{l-k}, n) : 1 \le k \le l\}$.

We identify $h_{\lambda_i-i+j}(x_1, ..., x_n)$ with the weighted sum of all possible paths from a_i to b_j .

Applying Lindström–Gessel–Viennot's Lemma 6.2 to the set of tuples $(P_1, ..., P_l)$ of paths from A to B gives

$$\det(h_{\lambda_i-i+j}(x_1,...,x_n))_{1\leq i,j\leq n} = \sum_{\substack{(P_1,...,P_n)\\P_i:a_i\to b_i\\\text{nonintersecting}}} w(P_1)\cdots w(P_n).$$

Given such a tuple, one constructs a semistandard Young tableaux of shape λ by letting $w(P_{l+1-i})$ be the weight of the *i*th row of the tableau. See Figure 6.5.

CLAIM. This map is well-defined and bijective.

Proof of claim. Given a tuple $(P_1, ..., P_n)$ of paths, with $P_i : a_i \rightarrow b_i$, the above procedure uniquely determines a set of weakly increasing rows. To see that they indeed define a semistandard Young tableau, one must check that, once these are interpreted as a single tableau *T*, columns of *T* are strictly increasing. Note that under the above correspondence T(i, j) = m if the edge $(l-i+j, m) \rightarrow (l-i+j+1, m)$ belongs to P_{l+1-i} .

Suppose therefore that *T* is semistandard restricted to the first *i* rows, and that it is even semistandard if the first j - 1 columns of row i + 1 are considered, but that $\alpha := T(i, j) \ge T(i + 1, j) =: \beta$. If we let $\gamma := T(i + 1, j - 1)$, then

- $(l i + j, \alpha) \rightarrow (l i + j + 1, \alpha)$ belongs to P_{l+1-i} ,
- $(l i + j 1, \beta) \rightarrow (l i + j, \beta)$ belongs to P_{l-i} , and
- $(l i + j 1, \gamma) \rightarrow (l i + j, \gamma)$ belongs to P_{l+1-i} .

By the inequalities $\gamma \le \beta \le \alpha$ which we have by hypotheses, these two paths intersect at $(l - i + j - 1, \beta)$, giving a contradiction. See Figure 6.4.

$$a_{l-i} \cdots \underbrace{x_{\beta}}_{a_{l+1-i}} \cdots \underbrace{b_{l+1-i}}_{x_{\gamma}} (l-i+j-1,\beta)$$

Figure 6.4: The paths
$$P_{l+1-i}$$
 and P_{l-i} intersect at $(l - i + j - 1, \beta)$.

We may define the inverse row by row. This is well defined by the reciprocal argument to the one above: if the resulting path is intersecting, say if path $a_i \rightarrow b_i$ intersects $a_j \rightarrow b_j$ (say i > j) at (k, m), then $T(l+1-i, k-l-1+i) \ge m = T(l+1-j, k-l-1+i)$, which contradicts T being semistandard.



Figure 6.5: The classic bijection between Young tableaux and nonintersecting lattice paths.

Note 6.6. For the bijection between paths and tableaux, the set *A* can be taken as the diagonal $\{(k, l-k+1) : 1 \le k \le l\}$ instead. A different determinantal formula arises. Namely, $s_{\lambda}(X) = \det(h_{\lambda_i - i+j}(X_i))_{1 \le i,j \le n}$, where $X_i := x_i + \cdots + x_n$.

Theorem 6.7 (Dual Jacobi–Trudi). Let $X = x_1 + \cdots + x_n$ be an alphabet. We have

$$s_{\lambda}(X) = \det \left(e_{\lambda'_i - i + j}(X) \right)_{1 \le i, j \le n} \in \Lambda_n(X),$$

where $e_0 = 1$ and $e_{-k} = 0$ for all $k \ge 1$.

Sketch of proof. We sketch two different proofs. The first sketch assumes the Jacobi–Trudi formula 6.1. To see why both determinants coincide, one may use that the ω involution on symmetric polynomials (see Chapter 1) takes s_{λ} to $s_{\lambda'}$ [Sta99, Thm. 7.15.6]. Assuming this, since it is an algebra homomorphism by definition, we can write

$$s_{\lambda}(X) = \omega(s_{\lambda'}(X)) = \omega\left(\det\left(h_{\lambda'_{i}-i+j}(X)\right)_{1 \le i,j \le n}\right)$$
$$= \det\left(\omega(h_{\lambda'_{i}-i+j}(X))\right)_{1 \le i,j \le n} = \det\left(e_{\lambda'_{i}-i+j}(X)\right)_{1 \le i,j \le n}$$

/

Alternatively, one can mimic the combinatorial proof of Theorem 6.1, suitably modifying the construction. See, for instance [Sag01, Thm. 4.5.1].

In the same spirit, in [FH91, KT87], they define a family of symmetric polynomials that specialize to the irreducible characters of types B and C.

Definition 6.8. We define the symmetric polynomial

$$D_{\lambda} := \det \left(h_{\lambda_i - i + 1} \mid h_{\lambda_i - i + j} + h_{\lambda_i - i - j + 2} \right)_{1 \le i, j \le n} \in \Lambda_{2n}.$$

The notation for the determinant is $det(a_{i,1}|a_{i,j})_{i,j} = det(a_{i,j})_{i,j}$. Here, $h_0 = 1$ and $h_{-k} = 0$ for all $k \ge 1$.

Two alternative descriptions of this symmetric polynomial are given below. The first one (Theorem 6.9) can be thought of as an analogue of the dual Jacobi–Trudi formula 6.7, whereas the second one (Lemma 6.11) is analogous to the formula given in Note 6.6, and allows us to give a simple and elegant combinatorial proof similar to the one given for type A. Further descriptions of D_{λ} may be found in [FH91, Section A.3].

Theorem 6.9. We have

$$D_{\lambda} = \det(e_{\lambda'_i - i + j} - e_{\lambda'_i - i - j})_{1 \le i, j \le n} \in \Lambda_{2n}.$$

Here, $e_0 = 1$ *and* $e_{-k} = 0$ *for all* $k \ge 1$.

We will follow the proof found in [FH91, Section A.3]. We use the following technical lemma.

Lemma 6.10. Fix integers r and $k \le r$. Let A and B be $r \times r$ matrices, with AB = cI for some scalar c. Let $\sigma = (S, S')$ and $\tau = (T, T')$ be words in $[r]^k \times [r]^{r-k}$ corresponding to permutations in \mathbb{S}_r . Then,

 $\det(A) \det(B_{S',T'}) = \operatorname{sgn}(\sigma) \operatorname{sgn}(\tau) c^{r-k} \det(A_{S,T}),$

where $A_{S,T} = (A_{\sigma(i),\tau(j)})_{i,j \leq k}$ and $B_{S',T'} = (B_{\sigma(i+k),\tau(j+k)})_{i,j \leq l}$ are submatrices of A and B, respectively.

Proof. Let *P* and *Q* be the permutation matrices of σ and τ^{-1} , respectively. Then, we can write

$$PAQ = \begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix}$$
 and $Q^{-1}BP^{-1} = \begin{pmatrix} B_1 & B_2 \\ B_3 & B_4 \end{pmatrix}$

with $A_1 = A_{S,T}$ and $B_4 = B_{S',T'}$. We have

$$\begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix} \begin{pmatrix} I & B_2 \\ 0 & B_4 \end{pmatrix} = \begin{pmatrix} A_1 & 0 \\ A_3 & cI \end{pmatrix},$$

and taking determinants, $\det(P) \det(Q) \det(A) \det(B_4) = \det(A_1) \cdot c^{r-k}$, giving the result.

Proof of Thm. 6.9. The proof will follow from Lemma 6.10. Let $r = l(\lambda) + l(\lambda')$ and let $k = l(\lambda)$. To define matrices *A* and *B*, we are going to *fold* two matrices $H := (h_{i-j})_{i,j \le r}$ and $E := ((-1)^{i-j}e_{i-j})_{i,j \le r}$.

More precisely, *A* is going to be the matrix resulting from folding *H* along the *k*th column and adding to each column to the left of the fold the column which lies the same distance to the right of the fold. Similarly, *B* is constructed by folding along the *k*th *row* and *subtracting* rows. For instance, for r = 4, k = 3,

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ h_1 & 1 & 0 & 0 \\ h_2 & h_1 & 1 & 0 \\ h_3 & h_2 & h_1 & 1 \end{pmatrix} \mapsto A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ h_1 & 1 & 0 & 0 \\ h_2 & h_1 & 1 & 0 \\ h_3 & 1+h_2 & h_1 & 1 \end{pmatrix}, \quad E = \begin{pmatrix} 1 - e_1 & e_2 & -e_3 \\ 0 & 1 & -e_1 & e_2 \\ 0 & 0 & 1 & -e_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \mapsto B = \begin{pmatrix} 1 - e_1 & e_2 & -e_3 \\ 0 & 1 & -e_1 & e_2 \\ 0 & 0 & 1 & -e_1 \\ 0 & -1 & e_1 & 1-e_2 \end{pmatrix}.$$

Note that HE = I by Corollary 1.13.

Claim 3. AB = I.

Proof of claim. It suffices to show that if HE = I then the same holds for their folded versions. In this proof, $H = (H_{i,j})_{i,j}$ and $E = (E_{i,j})_{i,j}$ may be taken to be arbitrary matrices. After folding the matrices

along the kth column and row, respectively, and multiplying the resulting matrices A and B, we get a new matrix M that we claim is the identity. Indeed, we have

$$\begin{split} M_{i,j} &= \sum_{p \in [n]} A_{i,p} B_{p,j} = \sum_{p < k} A_{i,p} B_{p,j} + A_{i,k} B_{k,j} + \sum_{p > k} A_{i,p} B_{p,j} \\ &= \sum_{p < k} (H_{i,p} + H_{i,2k-p}) E_{p,j} + H_{i,k} E_{k,j} + \sum_{p > k} H_{i,p} (E_{p,j} - E_{2k-p,j}) \\ &= \sum_{p \in [n]} H_{i,p} E_{p,j} + \sum_{p < k} H_{i,2k-p} E_{p,j} - \sum_{p > k} H_{i,p} E_{2k-p,j} \\ &= (HE)_{i,j} + \sum_{p < k} H_{i,2k-p} E_{p,j} - H_{i,2k-p} E_{p,j} = \delta_{i,j}. \end{split}$$

Let $(S; S') = (\lambda_1, ..., \lambda_k; -\lambda'_1, ..., -\lambda'_l) + (k, k-1, ..., 1; k+1, k+2, ..., k+l)$. Let (T; T') = (k, k-1, ..., 1; k+1, k+2, ..., k+l). That (T, T') is a permutation is clear.

CLAIM 4. (S, S') is a permutation.

Proof of claim. We have that *S* (resp. *S'*) is an injective word. Suppose there is an *s* in $S \cap S'$. Then there exist *i* and *j* such that $\lambda_i + k - (i - 1) = s = k + j - \lambda'_i$.

Equivalently, $\lambda_i + \lambda_j = i + j - 1$. On the right hand side, we are counting the number of boxes in a subset $[(j, 1^{i-1})] = \{(1, k) : k \leq j\} \cup \{(k, 1) : k \leq i\}$ of $[\lambda]$. If the cell (i, j) is not in $[\lambda]$, then $\lambda_i \leq j - 1$ and $\lambda'_j \leq i - 1$, which gives a contradiction. If (i, j) is in $[\lambda]$, then $\lambda_i \geq j$ and $\lambda'_j \geq i$, which again gives a contradiction. We illustrate an example:

$$i + j - 1 \qquad \lambda_i + \lambda'_j \qquad \lambda_i + \lambda'_j - 1 \\ \text{for } (i, j) \notin [\lambda] \qquad \text{for } (i, j) \in [\lambda]$$

By definition, $\det(A_{S,T}) = D_{\lambda}$ and $\det(B_{S',T'}) = \det(e_{\lambda'_i - i + j} - e_{\lambda'_i - i - j})_{1 \le i,j \le n}$. Also, $\det(A) = 1$. We are now in the hypotheses of Lemma 6.10, which gives the result.

We now present D_{λ} with a formula that closely resembles the formula in Note 6.6.

Lemma 6.11. Let $\bar{X}_i = x_i + \cdots + x_n + x_i^{-1} + \cdots + x_n^{-1}$. Fix $\delta \in \{0, 1\}$. We have

$$D_{\lambda}(X_1 + \delta) = \det \left(h_{\lambda_i - i + j}(X_i + \delta) \right)_{1 \le i, j \le n} \in \Lambda_{2n}.$$

Proof. We loosely follow [Oka89, SV16]. We begin with an inductive formula.

CLAIM. $h_k(\bar{X}_{i+1} + \delta) = h_k(\bar{X}_i + \delta) - (x_i + x_i^{-1})h_{k-1}(\bar{X}_i + \delta) + h_{k-2}(\bar{X}_i + \delta).$

Proof of claim. We interpret h_k as the generating function of (semistandard) row tableaux of size k (see Proposition 4.6). We assume $\delta = 0$ for ease of notation. Assuming $\delta = 1$ gives a similar proof.

Let R_i^k be the set of row tableaux of size k in the alphabet $A_i = \{i < i' < \cdots < n < n'\}$. We construct a weight-preserving bijection

$$R_{i+1}^k \cup \left(\{[i], [i']\} \times R_i^{k-1}\right) \quad \to \quad R_i^k \cup R_i^{k-2}.$$

Start by sending any element of R_{i+1}^k to itself. An element (x, T) in the second set of the left hand side gets sent to the concatenation of the pair if this is possible. If this is not possible, then $x = \begin{bmatrix} i \end{bmatrix}$ and
T(1,1) = [i], and we send the pair to the tableau resulting from removing the first box from *T*. For example, let i = 2, k = 3. Then,

3	3	4	$ \mapsto$	3	3	4
2,	3	4	$ \mapsto$	2	3	4
2,	2'	4	\mapsto	2	2'	4
2',	2	4	\mapsto	4		

This is clearly weight-preserving. To see why it is bijective, we consider an element in the right hand side. If the element is in R_i^k and it has no *i*, then it must come from R_{i+1}^k , and the preimage is uniquely determined. If it does have an i or an i', then one can split it into its first box and the rest to recover the preimage. Finally, an element *T* of R_i^{k-2} must come from (i', T') where *T'* is the concatenation of i and *T*.

In order to simplify the notation, we let $z_k^i := h_k(\bar{X}_i + \delta)$. Furthermore, for any fixed j, let Z_j^i be the column vector $(z_{\lambda_k-k+j}^i)_{k \le n}$. The above equation gives

$$Z_j^{i+1} = Z_j^i - (x_i + x_i^{-1})Z_{j-1}^i + Z_{j-2}^i$$

Start with the matrix $(h_{\lambda_i-i+1} \mid h_{\lambda_i-i+j} + h_{\lambda_i-i-j+2})_{1 \le i,j \le n}$, which appears implicitly on the left hand side of our desired equation. We write it in the following way:

$$\left(Z_1^1 \mid Z_2^1 + Z_0^1 \mid Z_3^1 + Z_{-1}^1 \mid Z_4^1 + Z_{-2}^1 \mid \cdots \mid Z_n^1 + Z_{-n+2}^1\right).$$

Subtracting from the *j*th the (j-1)st column multiplied by $(x_1 - x_1^{-1})$ we obtain $Z_j^2 + Z_{-j}^2 - Z_{j-2}^1 - Z_{-j}^1$. Adding now the (j-2)nd column gives $Z_j^2 + Z_{-j}^2$. We do this process, in order, for j = n, n-1, ..., 2. The resulting matrix is

$$\left(Z_1^1 \mid Z_2^2 \mid Z_3^2 + Z_1^2 \mid Z_4^2 + Z_0^2 \mid \cdots \mid Z_n^2 + Z_{-n}^2\right).$$

In the resulting matrix, x_1 appears exclusively in the first column. We freeze the first column and repeat the argument, so that x_2 appears only in the first two columns. We iterate for x_3 , x_4 , etc. until we have the matrix

$$\left(Z_1^1 \middle| Z_2^2 \middle| Z_3^3 \middle| Z_4^4 \middle| \cdots \middle| Z_n^n\right)$$

which is the matrix in the right hand side of the equality we aim to show. Since column operations preserve the determinant, we are done.

Theorem 6.12. We have

(B). $D_{\lambda}(x_1, ..., x_n, x_1^{-1}, ..., x_n^{-1}, 1) = o_{\lambda}(x_1, ..., x_n)$, and

(C).
$$D_{\lambda}(x_1, ..., x_n, x_1^{-1}, ..., x_n^{-1}) = sp_{\lambda}(x_1, ..., x_n)$$

Proof. We follow the proofs in [Oka89] and [SV16]. Fully combinatorial proofs can be found in [FK97, Section 5] (without using Lemma 6.11).

We construct a weighted lattice digraph $L = (V, \vec{E}, w)$ as in the proof of Theorem 6.1, with $V = [H] \times [N]$ for $N \gg 1$, where H = 2n + 1 for type B and H = 2n for type C. The set \vec{E} consists of vertical arrows $(i, j) \rightarrow (i, j+1)$ of weight 1 whenever this makes sense, and horizontal arrows $(i, j) \rightarrow (i+1, j)$ for $1 \le j \le 2n$, which are now of weight x_i at height 2i - 1 and x_i^{-1} at height 2i. For type B exclusively, we introduce diagonal arrows of weight 1 from each (i, 2n) to (i + 1, 2n + 1).

The sets *A* and *B* are defined as follows:

$$A = \{a_k := (k, 2(l-k)+1) : 1 \le k \le l\} \text{ and } B = \{b_k := (k+\lambda_{l-k}, H) : 1 \le k \le l\}.$$

(This again resembles Note 6.6.) See Figure 6.13. We thus identify $h_{\lambda_i - i + j}(\bar{X}_i + \delta)$ (for $\delta = 0$ or 1, depending if we are in type C or B respectively) with the weighted sum of paths $a_i \rightarrow b_i$.

Applying Lindström–Gessel–Viennot's lemma and Lemma 6.11 gives the results. (By definition of the set *A*, the tableaux are symplectic. The bijection is well-defined just as in type A.) ■



Figure 6.13: The bijection between symplectic and orthogonal tableaux and nonintersecting lattice paths.

Recall the definition of the dominance order from Chapter 1: $\lambda \ge \mu$ if $\lambda_1 + \cdots + \lambda_i \ge \mu_1 + \cdots + \mu_i$ for all *i*. We now give a classic but often overlooked statement for Schur polynomials and the analog for D_{λ} . It will let us show that these latter form a basis of Λ_{2n} .

Theorem 6.14. Let $s_{\lambda} = \sum d_{\lambda,\mu}h_{\mu}$. Then $d_{\lambda,\mu} \neq 0$ implies $\lambda \geq \mu$. Moreover, $d_{\mu,\mu} = 1$.

Proof. We use the structure of the matrix of the Jacobi–Trudi formula. Imagine we are trying to build a counterexample; a partition μ that is as big as possible. The biggest value that μ_1 can get (among all indices of the entries of the matrix) is $\lambda_1 - 1 + 1 = \lambda_1$. Given that we want $\mu_1 \ge \lambda_1$, this must be the value of μ_1 .

Now, the biggest value of μ_2 in a minor of the matrix is $\lambda_2 - 2 + 2 = \lambda_2$, given that μ_1 discards the first row already. Since we want $\mu_1 + \mu_2 \ge \lambda_1 + \lambda_2$, this must be the value of μ_2 .

This reasoning iterates to obtain $\mu = \lambda$ as the biggest constituent of the sum.

To show that $d_{\mu,\mu} = 1$, we note that the biggest factor of the determinant constructed above is unique. And since it is the principal diagonal, it is counted with positive sign.

Corollary 6.15. The family $\{s_{\lambda}\}_{\lambda \vdash d}$ is a basis of Λ_n^d .

Theorem 6.16. Let $D_{\lambda} = \sum d_{\lambda,\mu}h_{\mu}$. Then $d_{\lambda,\mu} \neq 0$ implies $\lambda \geq \mu$ and $|\lambda| \equiv |\mu| \mod 2$. Moreover, $d_{\mu,\mu} = 1$. We have $D_{(k)} = h_{(k)}$ and $D_{(1^k)} = e_{(k)} - e_{(k-2)}$ for all $k \in \mathbb{N}$.

Note 6.17. In [KT87], they show that $\{D_{\lambda}\}_{\lambda}$ is a basis by invoking a similar result. Their result is the corollary of a stronger theorem: a change of basis equation between $\{D_{\lambda}\}_{\lambda}$ and $\{s_{\lambda}\}_{\lambda}$.

Proof. To show the first assertion, note that any counterexample λ would necessarily have a summand h_{μ} coming from det $(h_{\lambda_i-i+j})_{i,j}$ with $\lambda \ge \mu$. But the assertion is true for Schur polynomials. The congruence equality comes from the fact that we either have $(\lambda_i - i) + j$ or $(\lambda_i - i) - j$ for all *i* and *j*; these are all an even number apart.

The fact that $d_{\mu,\mu} = 1$ is the same argument as for Schur polynomials.

The last facts are also straightforward; in this cases, D_{λ} is the determinant of 1×1 matrices.

Corollary 6.18. The family $\{D_{\lambda}\}_{\lambda \vdash d}$ is a basis of Λ_{2n}^d .

Chapter 7

The Lie theoretic and algebraic definitions coincide

In this chapter, we show that the determinants given in Chapter 6 agree with the irreducible characters of the classical Lie algebras (given in Chapter 3). Explicitly, letting $X = x_1 + \cdots + x_n$ and $\overline{X} = x_1 + \cdots + x_n + x_n^{-1} + \cdots + x_n^{-1} + \cdots + x_n^{-1}$, we show:

Theorem 7.1. Let λ be a partition indexing an irreducible representation of $\mathfrak{sl}(n)$, $\mathfrak{so}(2n + 1)$, or $\mathfrak{sp}(2n)$, respectively. Then,

(A) $\chi_{\lambda}^{\mathfrak{sl}(n)}(X) = \det(h_{\lambda_i - i + j}(X))_{1 \le i, j \le n}$,

(B)
$$\chi_{\lambda}^{\mathfrak{so}(2n+1)}(X) = D_{\lambda}(\bar{X}+1), and$$

(C)
$$\chi_{\lambda}^{\mathfrak{sp}(2n)}(X) = D_{\lambda}(\bar{X}).$$

We follow [FH91, Appendix A].

The type A formula

We will start by showing the result for type A. Explicitly, we show

_ () (--)

.

$$\frac{\det(x_j^{\lambda_i+n-i})_{1\leq i,j\leq n}}{\det(x_i^{n-i})_{1\leq i,j\leq n}} = \det(h_{\lambda_i+j-i}(X))_{1\leq i,j\leq n}.$$

Recall Corollary 1.13. We have the following similar result.

Lemma 7.2. Let $X = x_1 + \cdots + x_n$ be an alphabet. For each $d \ge n$, $j \in [n]$, we have $\sum_{i=0}^n (-1)^i e_i(X) x_j^{d-i} = 0$.

Proof. Let $X = x_1 + \cdots + x_n$ and let $X' = x_1 + \cdots + x_{j-1} + x_{j+1} + \cdots + x_n$. In this proof we write E(t)(X) to specify the alphabet of e_d , $d \ge 0$. On the one hand, we have

$$\frac{E(-t)(X)}{1-x_jt} = E(-t)(X') = \sum_{d\ge 0} (-1)^d e_d(X')t^d$$

Note that $e_d(X') = 0$ for $d \ge n$. On the other hand, we have

$$\frac{E(-t)(X)}{1-x_jt} = E(-t)(X) \cdot (1+x_jt+x_j^2t^2+\cdots) = \sum_{d\geq 0} \left(\sum_{i=0}^n (-1)^i e_i(X)x_j^{d-i}\right) t^d.$$

Comparing the terms of *t*-degree *d* gives the result.

In particular, one may write x_j^d in terms of $e_1, ..., e_n$ and $x_j^1, ..., x_j^{d-1}$. By induction, one may write $x_j^d = \sum_{i=0}^{n-1} a_i^d(e_1, ..., e_n) x_j^{d-i}$ for some polynomials a_i^d , i = 0, ..., n-1, in $\mathbb{C}[e_1, ..., e_n]$. Since the equation in Corollary 1.13 is of the same form, we may also write $h_d = \sum_{i=0}^{n-1} a_i^d(e_1, ..., e_n) h_{d-i}$. For any partition λ we therefore have the matrix equations

$$(x_{j}^{\lambda_{i}+n-i})_{1 \le i,j \le n} = (a_{r}^{\lambda_{i}+n-i})_{1 \le i,r \le n}(x_{j}^{n-r})_{1 \le r,j \le n} \quad \text{and} \\ (h_{\lambda_{i}+j-i})_{1 \le i,j \le n} = (a_{r}^{\lambda_{i}+n-i})_{1 \le i,r \le n}(h_{j-r})_{1 \le r,j \le n}.$$

Note that $((-1)^{j-i}e_{j-i})_{ij} = (h_{i-j})_{ij}^{-1}$ (see Corollary 1.13). Consequently, one can manipulate the two identities above to write

$$(x_{j}^{\lambda_{i}+n-i})_{1 \le i,j \le n} = (h_{\lambda_{i}+p-i})_{1 \le i,p \le n} ((-1)^{q-p} e_{q-p})_{1 \le p,q \le n} (x_{j}^{n-q})_{1 \le q,j \le n}$$

Taking determinants shows the result (Theorem 7.1.A), since $((-1)^{q-p}e_{q-p})_{p,q}$ is a lower triangular matrix with 1s in the diagonal.

Note that, in particular, we have shown that x_j^l may be expressed as the following vector-matrix-vector product,

$$x_{j}^{l} = (h_{l-n+p}(X))_{1 \le p \le n} ((-1)^{q-p} e_{q-p}(X))_{1 \le p,q \le n} (x_{j}^{n-q})_{1 \le q \le n}.$$

$$(7.3)$$

The type C formula

In a similar fashion, we now show the formula for type C. For any j = 1, ..., n and any $p \in \mathbb{N}_0$, let z_j^p denote the polynomial $x_j^p - x_j^{-p}$. Every symmetric polynomial in this section is assumed to be in the alphabet \overline{X} . We want to show

$$\frac{\det(z_j^{\lambda_i+n-i+1})_{1\leq i,j\leq n}}{\det(z_j^{n-i+1})_{1\leq i,j\leq n}} = \det(h_{\lambda_i-i+1}\mid h_{\lambda_i-i+1+j}+h_{\lambda_i-i+1-j})_{1\leq i,j\leq n}$$

The result will fall from the following lemma.

Lemma 7.4. For any $l \ge 0$, $j \in [n]$, we can write z_j^l as the following vector-matrix-vector product,

$$z_{j}^{l} = (h_{l-n} \mid h_{l-n+p} + h_{l-n-p})_{1 \le p \le n} ((-1)^{q-p} e_{q-p})_{1 \le p,q \le n} (z_{j}^{n+1-q})_{1 \le q \le n}$$

where $(a_1 | a_p)_{1 \le p \le n}$ denotes $(a_1, a_2, ..., a_n)$.

Indeed, this allows us to write $(z_j^{\lambda_i+n-i+1})_{1 \le i,j \le n}$ as

$$(h_{\lambda_i-i+1} \mid h_{\lambda_i-i+1+p} + h_{\lambda_i-i+1-p})_{1 \le i, p \le n} ((-1)^{q-p} e_{q-p})_{1 \le p, q \le n} (z_j^{n-q+1})_{1 \le q, j \le n},$$

and taking determinants gives the result (Theorem 7.1.C), as before.

So it only remains to show the lemma.

Proof of Lemma 7.4. We use Equation (7.3) but in the alphabet $\overline{X} = x_1 + \cdots + x_n + x_n^{-1} + \cdots + x_1^{-1}$ to get

$$\begin{split} x_j^l &= (h_{l-2n+p}(\bar{X}))_{1 \le p \le 2n} ((-1)^{q-p} e_{q-p}(\bar{X}))_{1 \le p,q \le 2n} (x_j^{2n-q})_{1 \le q \le 2n} \quad \text{and} \\ x_j^{-l} &= (h_{l-2n+p}(\bar{X}))_{1 \le p \le 2n} ((-1)^{q-p} e_{q-p}(\bar{X}))_{1 \le p,q \le 2n} (x_j^{q-2n})_{1 \le q \le 2n} . \end{split}$$

Subtracting both expressions gives

$$z_j^l = \sum_{p=1}^{2n} h_{l-2n+p}(\bar{X}) S_p , \quad \text{where} \quad S_p = \sum_{q=p}^{2n} (-1)^{q-p} e_{q-p}(\bar{X}) z_j^{2n-q}.$$
(7.5)

Fix $p \in [2n]$. Lemma 7.2 for the alphabet \overline{X} and an arbitrary $d \gg 2n$, gives

$$x_j^d - e_1(\bar{X})x_j^{d-1} + e_2(\bar{X})x_j^{d-2} - \dots + e_{2n}(\bar{X})x_j^{d-2n} = 0.$$

Multiplying by x_i^{p-d} and isolating terms then results in

$$x_j^p - e_1(\bar{X})x_j^{p-1} + e_2(\bar{X})x_j^{p-2} - \dots + (-1)^p e_p x_j^0 = (-1)^{p+1} e_{p+1}(\bar{X})x_j^{-1} + \dots - e_{2n}(\bar{X})x_j^{p-2n}$$

This same expression but for j = 2n + 1 - j (that is, for the variable x_j^{-1}) is

$$x_{j}^{-p} - e_{1}(\bar{X})x_{j}^{1-p} + e_{2}(\bar{X})x_{j}^{2-p} - \dots + (-1)^{p}e_{p}x_{j}^{0} = (-1)^{p+1}e_{p+1}(\bar{X})x_{j}^{1} + \dots - e_{2n}(\bar{X})x_{j}^{2n-p}.$$

Subtracting both identities gives

$$\overbrace{z_{j}^{p} - e_{1}(\bar{X})z_{j}^{p-1} + e_{2}(\bar{X})z_{j}^{p-2} - \dots + (-1)^{p-1}e_{p-1}z_{j}^{1}}^{= (-1)^{p}e_{p+1}(\bar{X})z_{j}^{1} + \dots + e_{2n}(\bar{X})z_{j}^{2n-p}}^{= (-1)^{p}e_{p+1}(\bar{X})z_{j}^{1} + \dots + e_{2n}(\bar{X})z_{j}^{2n-p}}.$$
(7.6)

We use the following result.

CLAIM. $(-1)^{p}e_{p}(\bar{X}) = (-1)^{2n-p}e_{2n-p}(\bar{X}).$

Proof of claim. We have $\sum_{d\geq 0} e_d(\bar{X})t^d = E(-t)(\bar{X}) = \prod_{i\in[n]} (1-x_it)(1-x_i^{-1}t) = \prod_{i\in[n]} (1-(x_i+x_i^{-1})t+t^2)$. In particular, we can write $E(-t)(\bar{X}) = t^{2n}E(-t^{-1})(\bar{X})$.

Let us denote by [a]P(t) the coefficient of t^a in the polynomial P(t). We have

$$[2n-p]E(-t)(\bar{X}) = [2n-p]t^{2n}E(-t^{-1})(\bar{X}) = [-p]E(-t^{-1})(\bar{X}) = [p]E(-t)(\bar{X}).$$

It follows from the claim that the right hand side of Equation (7.6) is S_p and thus we have $S_p = S_{2n-p}$. Note that we can also write the left hand side as R_{n-p+1} , where $R_p := \sum_{q=p}^{n} (-1)^{q-p} e_{q-p}(\bar{X}) z_j^{n+1-q}$.

We bring back Equation (7.5) to obtain

$$\begin{split} z_{j}^{l} &= \sum_{p=1}^{2n} h_{l-2n+p}(\bar{X}) S_{p} \\ &= \underbrace{h_{l-2n+1}S_{1}}_{= \ 0 \ \text{by 7.2}} + \left(\sum_{p=2}^{n} h_{l-2n+p}(\bar{X}) S_{p} \right) + h_{l-n} S_{n+1} + \left(\sum_{p=n+2}^{2n} h_{l-2n+p}(\bar{X}) S_{p} \right) \\ &= \left(\sum_{p=2}^{n} h_{l-2n+p}(\bar{X}) S_{p} \right) + h_{l-n} S_{n+1} + \left(\sum_{p=2}^{n} h_{l-2n-p}(\bar{X}) S_{2n-p} \right). \end{split}$$

And now, since $S_p = S_{2n-p} = R_{n-(p-1)}$, a reparametrization gives

$$z_j^l = h_{l-n}R_0 + \sum_{p=0}^{n-2} (h_{l-n+p}(\bar{X}) + h_{l-n-p}(\bar{X}))R_{p+1}$$

This is exactly the formula we were seeking.

The type B formula

Finally, we turn our attention to the type B formula. We let $z_j^p := x_j^p - x_j^{-p}$ for j = 1, ..., n and $p \in \frac{1}{2}\mathbb{N}_0$. Every symmetric polynomial in this section is assumed to be in the alphabet $\overline{X} + 1$. We remind the reader that we only treat spinless irreducible representations (those indexed by partitions, as opposed to half-partitions). We want to show Theorem 7.1.B,

$$\frac{\det(z_j^{\lambda_i+n-i+1/2})_{1 \le i,j \le n}}{\det(z_j^{n-i+1/2})_{1 \le i,j \le n}} = \det(h_{\lambda_i-i+1} \mid h_{\lambda_i-i+1+j} + h_{\lambda_i-i+1-j})_{1 \le i,j \le n}$$

Note that $z_j^p \cdot (x_j^{1/2} + x_j^{-1/2}) = z_j^{p+1/2} + z_j^{p-1/2}$. Thus, starting from the left hand size of the above equation, one may multiply numerator and denominator by $\prod_{j=1}^n (x_j^{1/2} + x_j^{-1/2})$ to obtain

$$\frac{\det(z_j^{\lambda_i+n-i+1/2})_{1\leq i,j\leq n}}{\det(z_j^{n-i+1/2})_{1\leq i,j\leq n}} = \frac{\det(z_j^{\lambda_i+n-i+1}+z_j^{\lambda_i+n-i})_{1\leq i,j\leq n}}{\det(z_j^{n-i+1}+z_j^{n-i})_{1\leq i,j\leq n}} = \frac{\det(z_j^{\lambda_i+n-i+1}+z_j^{\lambda_i+n-i})_{1\leq i,j\leq n}}{\det(z_j^{n-i+1})_{1\leq i,j\leq n}}$$

To see the second equality, note that the matrix $(z_j^{n-i+1} + z_j^{n-i})_{ij}$ may be obtained from the Vandermonde matrix $(z_j^{n-i+1})_{ij}$ by performing row operations. (Note also $z_j^0 = 0$.) Therefore, it has the same determinant.

We now use Lemma 7.4 and the formula $h_k(\bar{X} + 1) = h_k(\bar{X}) + h_{k-1}(\bar{X})$ (see also Proposition 4.7) to conclude.

Chapter 8

Other combinatorial models for type C

In this chapter we survey the different models for symplectic tableaux and the way the different models are related.

At first glance, the most noticeable difference between the models is the alphabet that they use. Throughout this chapter, we will mainly use the following four alphabets:

 $\mathcal{A} = \{1 < 1' < 2 < 2' < \dots < n < n'\},$ $\mathcal{B} = \{n' < \dots < 2' < 1' < 1 < 2 < \dots < n\},$ $C = \{1 < 2 < \dots < n < n' < \dots < 2' < 1'\}, \text{ and}$ $\mathcal{D} = \{1 < 2 < \dots < 2n\}.$

We refer to any alphabet whose underlying set is $\{1, 2, ..., n, 1', 2', ..., n'\}$ as a **symplectic alphabet**. For instance, \mathcal{A}, \mathcal{B} , and *C* above are symplectic. Given a symplectic alphabet \mathcal{X} , we write \mathcal{X}' for the alphabet in which primed and non-primed letters are exchanged. By relabeling of a tableau $T : [\lambda] \to \mathcal{X}$ from an alphabet \mathcal{X} to an alphabet \mathcal{Y} we mean postcomposition with the order-preserving map sending \mathcal{X} to \mathcal{Y} . We denote this by $T_{\mathcal{Y}}$.

We will, in total, deal with four families of tableaux: King's, De Concini's, Kashiwara's, and split tableaux. King's tableaux were already defined in Definition 4.1. We now define De Concini's. But first, we need to introduce some terminology.

Definition 8.1. We say that *T* is a column tableau if *T* is of shape (1^k) for some *k*. Let *T* be a semistandard column tableau in the alphabet $\mathcal{A} = \{1 < 1' < 2 < 2' < \cdots < n < n'\}$. We say *T* is admissible if $T(i, j) \ge i$. More generally, we say a column tableau *T* in a symplectic alphabet \mathcal{X} is **admissible** if a reordering of its entries produces an admissible semistandard column tableau in \mathcal{A} .

In other words, a column tableau T in a symplectic alphabet X is admissible if

$$\#T^{-1}(\{1, 1', ..., i, i'\}) \le i$$

for all *i*. Note that with this definition, King's tableaux are exactly semistandard tableaux in the alphabet \mathcal{A} in which every column is admissible.

Fix a symplectic alphabet X. In what follows, we identify a semistandard column tableaux T in X with the pair of sets (A, D), where $A \subseteq [n]$ is the set $\{a : a' \in im T\}$, and $D \subseteq [n]$ is the set $\{d : d \in im T\}$. This completely determines T.

Definition 8.2 (Split version in \mathcal{B}). Let T = (A, D) be an admissible semistandard column tableau in the alphabet $\mathcal{B} = \{n' < \cdots < 2' < 1' < 1 < 2 < \cdots < n\}$. The split version of T is the pair (P, Q) of semistandard column tableaux in \mathcal{B} which we obtain as a result of the following algorithm:

- set $J = \emptyset$
- for *i* ranging over $A \cap D$ in descending order:
 - − set $j \in [n]$ to be the greatest number which is not in $A \cup D$, and not in J, and is smaller than i

- add j to J

• return $P = (A, (D - A) \cup J)$ and $Q = ((A - D) \cup J, D)$.

We identify the pair (P, Q) with a tableau with two columns. Diagrammatically, split is the map $\begin{pmatrix} A \\ D \end{pmatrix} \mapsto \begin{pmatrix} A & B \\ C & D \end{pmatrix}$, where $B := (D - A) \cup J$, $C := (A - D) \cup J$, and J is defined via the above algorithm.

The **split version** of a tableau $T : [\lambda] \to \mathcal{B}$ of shape λ is a tableau of shape $(2\lambda_1, 2\lambda_2, ...)$ in which each column is exchanged for its split version. We denote this by $\operatorname{split}(T)$. More generally, the split version $\operatorname{split}(T)$ of a tableau $T : [\lambda] \to X$ is the relabeling $\operatorname{split}(T_{\mathcal{B}})_X$.

Example 8.3. Let $T = (\{3, 4\}, \{2, 3\})$ denote an admissible semistandard column tableaux in the alphabet \mathcal{B} . We have $3 \in \{3, 4\} \cap \{2, 3\}$, and thus we find j = 1 which is not in $\{3, 4\}$, nor in $\{2, 3\}$ and it is smaller than 3.



We note that it is useful to think of *A* and *B* to be in bijection: elements in A-D are sent to themselves, and each $i \in A \cap D$ is sent to their corresponding *j* as in the loop of the algorithm. Similarly, *D* and *C* are in bijection.

Example 8.4 (Example 8.3 contd.). We have a bijection $A \rightarrow B$ given by $3 \mapsto 1$ and $4 \mapsto 4$. We have a bijection $D \rightarrow C$ given by $2 \mapsto 2$ and $3 \mapsto 1$.

We point out some properties about the algorithm that are immediate from its definition. If *T* is a column tableau, then $\operatorname{split}(T)$ is a semistandard tableau. Moreover, if an entry a is found in column 1 of $\operatorname{split}(T)$, then

- a' is not in column 1, and
- exactly one of a and a' is in column 2.

A similar analysis can be done for an entry a' in column 1. Finally, we can break split(*T*) into blocks in which, for some minimum *a* and some maximum *b*, all numbers $c \in [a, b]$ appear (as either primed or non-primed entries) in the block, exactly once in each column. This becomes apparent in the next example.

Example 8.5. Let $T = (\{3, 4, 6, 8, 9, 11\}, \{4, 5, 6, 9\})$ be a column tableau in \mathcal{B} . Then, split(T) = (($\{3, 4, 6, 8, 9, 11\}, \{1, 2, 5, 7\}$), ($\{1, 2, 3, 7, 8, 11\}, \{4, 5, 6, 9\}$)). We let arrows denote the bijections $A \rightarrow B$ and $D \rightarrow C$ in the diagram below. We divide the split tableau into three blocks $X \cup X'$ (corresponding to the interval [1, 6]), $Y \cup Y'$ (corresponding to the interval [7, 9]), and $Z \cup Z'$ (corresponding to the interval [11, 11]). In this case, Z is empty.



Definition 8.6. A **De Concini (symplectic) tableau** is a semistandard tableau *T* in the alphabet $\mathcal{B} = \{n' < \cdots < 2' < 1' < 1 < 2 < \cdots < n\}$ such that each column of *T* is admissible, and such that the split version of *T* is semistandard.

Example 8.7. The following is a De Cocini tableau and its split version.

4'	4'	1		4'	4'	4'	3'	1	1
3'	1	4	\mapsto	3'	2'	1	1	4	4
3	4			2	3	3	4		

We can similarly introduce the notions of **coadmissible** column and **cosplit version**. A column tableau is called coadmissible if after the relabeling $i \mapsto n+1-i$ (and $i' \mapsto (n+1-i)'$), the column tableau becomes admissible. That is, if $\#T^{-1}(\{n, n', ..., i+1, i+1'\}) \leq i$. On the other hand, if one writes the split algorithm diagrammatically as $\binom{A}{D} \mapsto \binom{A \ B}{C \ D}$, then the cosplit map can be written as $\binom{B}{C} \mapsto \binom{A \ B}{C \ D}$. Explicitly:

Definition 8.8 (Cosplit version in \mathcal{B}). If T = (B, C) is a coadmissible semistandard column tableau in the alphabet \mathcal{B} , the cosplit version cosplit(T) of T is the pair (P, Q) resulting from the following algorithm:

- set $J = \emptyset$
- for *i* ranging over $B \cap C$ in ascending order:

- set *j* ∈ [*n*] to be the smallest number which is not in $B \cup C$, not in *J*, and is greater than *i* - add *j* to *J*

• return $P = ((B - C) \cup J, C)$ and $Q = (B, (C - B) \cup J)$.

See [She99] for more information. Again, the cosplit version of a tableau in \mathcal{B} is created exchanging each column for its cosplit version, and the cosplit version of a tableau $T : [\lambda] \to X$ is $\operatorname{cosplit}(T_{\mathcal{B}})_X$.

Note 8.9. If the split version of *T* coincides with the cosplit version of *Q*, then the weight of *T* is the same as the weight of *Q*. This is clear, since they only differ in pairs of (i, i') entries.

The tableaux above were introduced in [DeC79]. A bijection between King's tableaux relabeled to $\mathcal{A}' = \{1' < 1 < \cdots < n' < n\}$ and De Concini's tableaux is given in [She99]. We present it at the end of this chapter.

We now turn our attention to a third model of symplectic tableaux.

Definition 8.10. A Kashiwara (symplectic) tableau is a semistandard tableau T in the alphabet $C = \{1 < \dots < n < n' < \dots < 1'\}$ such that

- (K1) if *a* and *a'* appear in the same column, say T(r, c) = a, T(s, c) = a', then (s r) + a is strictly greater than the length of column T(-, c), and
- (K2) if two adjacent columns of *T* have one of the following four configurations:

$p \rightarrow$	а		а		а	а	а	
$q \rightarrow$		b	b					
$\hat{r} \rightarrow$		b'	b'					
$s \rightarrow$		a'		a'		a'	<i>a'</i>	a'

(by which we mean that e.g. the first a entry in the first column is at row number p, etc.), then (q-p) + (s-r) < (b-a). In particular, the third and fourth configurations are impossible. Here, $p \le q < r \le s$ and $a \le b$.

Note that the third and fourth conditions of (K2) are redundant, but included since they prove to be useful in later proofs.

It is often useful to think of a reformulation of (K2). The proof of this statement is included in Appendix C.

Lemma 8.11. Let T be a tableau in the alphabet C satisfying (K1). Then, T satisfies (K2) if and only if it satisfies (K2');

(K2') if two adjacent columns of T have one of the following two configurations:

$$\begin{array}{c|cccc} p \rightarrow & a & & a \\ q \rightarrow & b & b \\ r \rightarrow & c' & c' \\ s \rightarrow & d' & & d' \end{array}$$

then $(q - p) + (r - s) < \max\{b, c\} - \min\{a, d\}$. Here, $p \le q < r \le s$, $a \le b$, and $c \le d$.

As we will see, De Concini's and Kashiwara's tableaux are closely related. We give a bijection as a corollary to the following theorem, whose proof is postponed to Appendix C.

Theorem 8.12. Let T be a semistandard tableau in the alphabet $C = \{1 < \dots < n < n' < \dots < 1'\}$. A column of T verifies (K1) if and only if it is admissible. The tableau T verifies (K2) if and only if the cosplit version of T is semistandard.

Note 8.13. A semistandard column tableau *T* in *C* is admissible if and only if $T_{\mathcal{B}}$ is coadmissible. Therefore, a tableau $T : [\lambda] \to C$ is a Kashiwara tableau if and only if each column of $T_{\mathcal{B}}$ is coadmissible and the cosplit version of $T_{\mathcal{B}}$ is semistandard.

It will be nevertheless useful to have a way of computing the cosplit version of a tableau in *C* directly.

Definition 8.14 (Cosplit in *C*). Let $T : [\lambda] \to C = \{1 < \cdots < n < n' < \cdots 1'\}$ be a column tableau, which we represent as a pair (B, C) of subsets of [n] where $B = \{i : i \in \text{im } T\}$ and $C = \{i : i' \in \text{im } T\}$. Assume *T* to be admissible. The cosplit version cosplit(*T*) of *T* is the pair (P, Q) of column tableaux resulting from the following algorithm:

- set $J = \emptyset$
- for *i* ranging over $B \cap C$ in ascending order:
 - set $j \in [n]$ to be the greatest number which is not in $B \cup C$, not in J, and is smaller than i
 - add j to J
- return $P = ((B C) \cup J, C)$ and $Q = (B, (C B) \cup J)$.

If $T : [\lambda] \to \mathcal{B}$ is a De Concini tableau, we say T_C is a De Concini_C tableau.

Corollary 8.15. The map $\operatorname{cosplit}^{-1} \circ \operatorname{split}$ gives a weight-preserving bijection between De Concini_C tableaux and Kashiwara tableaux.

Inspired by this bijection, we present two novel and different characterization of De Concini tableaux. See Appendix C for proofs.

Proposition 8.16. Let T be a tableau in the alphabet \mathcal{B} such that each column is admissible. Then, T is a De Concini tableau if and only if it satisfies (DC');

(DC') if two adjacent columns of T have one of the following two configurations:

$$\begin{array}{c|ccc} p \to & a' \\ q \to & \\ r \to & c \\ s \to & d \end{array} \begin{vmatrix} b' & & & \\ b' & & c \\ & & \\ d \end{vmatrix}$$

then $(q - p) + (r - s) < \max\{a, d\} - \min\{b, c\}$. Here, $p \le q < r \le s$, $a \le b$, and $c \le d$.

Lemma 8.17. Let T be a tableau in the alphabet \mathcal{B} such that each column is admissible. Then, T satisfies (DC') if and only if it satisfies (DC);

(DC) if two adjacent columns of T have one of the following two configurations:

$$\begin{array}{c|ccc} p \to & a' \\ q \to & \\ r \to & b \\ s \to & a \end{array} \middle| \begin{array}{c} b' & & \\ b' & & \\ b & \\ a \end{array} \right|$$

then (q - p) + (r - s) < a - b. *Here,* $p \le q < r \le s$.

Note 8.18. In the literature, e.g. in [Kra98, Lit90, LMS79], some authors make the split version of the De Concini's tableaux their main object of study. (Or, the cosplit version of Kashiwara's tableaux.) The preferred alphabet in [Kra98] is $\mathcal{D} = \{1 < 2 < \cdots < 2n\}$. We call these **split tableaux**. We refer to [Kra98, Def. A3.1] for a precise definition.

The advantage of Kashiwara's and De Concini's tableaux over King's tableaux is that we have a description of the crystal structure on them. See Chapter 9.

Summing up, we have the following bijections, where vertical arrows are just relabeling.



In particular, the implementation of the composite allows one to compute the effect of crystal operators on any of these sets of tableaux in SageMath. See Appendix D.

The composite bijection from King's tableaux to De Concini's tableaux is **weight-inverting**; that is, a tableau with weight $x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n}$ gets mapped to a tableau with weight $x_1^{-i_1}x_2^{-i_2}\cdots x_n^{-i_n}$. The composite bijection from De Concini's tableaux to Kashiwara's tableaux is **weight-reversing**; that is, a tableau with weight $x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n}$ gets mapped to a tableau with weight $x_1^{-i_n}x_2^{-i_{n-1}}\cdots x_n^{-i_1}$.



Example 8.19. Here is an example of each map in the above diagram.

We will now describe Sheats' bijection and its inverse. It is based on a modification of the classical *jeu de taquin* algorithm [Sag01]. We present the bijections algorithmically. A detailed and careful description of these maps is an arduous task and beyond the scope of this short survey. We thus refer the interested reader to in [She99]. We introduce some useful definitions first.

Definition 8.20. Let *T* be a tableau *T* : $[\lambda] \to X$ and let $c \in [\lambda]$ be a cell. The **puncture** of *T* at *c* is the pair (T^c, c) where T^c is the restriction of *T* to $[\lambda] - c$. When representing tableaux as fillings of Young diagrams, we denote the puncture by \bullet .

A puncture (T^c, c) of a De Concini column tableau T at c = (r, 1), can be represented as a triple (A, D; c) by letting $A := \{a : a' \in \operatorname{im} T^c\}$ and $D := \{d : d \in \operatorname{im} T^c\}$. That is, if $T^c = (A, D)$, then $(T^c, c) = (A, D; c)$. This allows us to define the split version of $(T^c, (r, 1))$ as the double puncture $((\operatorname{split}(T^c), (r, 1)), (r, 2))$. Therefore, the split version of a puncture (T^c, c) of a De Concini tableau is defined to be the double puncture of the tableau of shape $(2\lambda_1, 2\lambda_2, ...)$ in which each column of (T^c, c) is exchanged for its split version.

Given a King's tableau *T*, we will refer to $T_{\mathcal{H}'}$ as a King_{\mathcal{H}'} tableau. Starting from a De Concini tableau, Sheats' bijection produces a King_{\mathcal{H}'} tableau by "moving" each primed entry using a modified version of Schützenberger's *jeu de taquin* algorithm [Sag01, Section 3.7] until it verifies the semistandard rules with respect to the alphabet \mathcal{H}' . In its intermediate stages, we get tableaux with a King_{$\mathcal{H}'} part and a De$ Concini part. We refer to these as mixed tableaux in what comes.</sub>

Definition 8.21 (Sheats' bijection). To initialize the algorithm, let *T* be a De Concini tableau (a mixed tableau with an empty $King_{\mathcal{A}'}$ part).

Given a mixed tableau, let i' be the smallest value in its De Concini part (with respect to the order induced by \mathcal{B}). We add every entry greater or equal to i with respect to $\leq_{\mathcal{H}'}$ to the King part. Let $c \in [\lambda]$ be the cell of the right-most instance of i' in the De Concini part. Consider the puncture (T^c, c) .

We now perform jeu de taquin on the double puncture $((\text{split}(T^c), c'), c'')$. At any given step of the jeu de taquin algorithm, the punctures will fall into the following configuration, where *a* and *b* might not exist.

•	٠	а	*
*	b		

If both *a* and *b* do not exist, then stop for now. Otherwise, if *a* doesn't exist, then we perform a down slide (defined below), and if *b* doesn't exists, then we perform a right slide. Finally, if both exist, we compare *a* and *b*. If $a \leq_{\mathcal{B}} b$ we perform a down slide and vice versa.

A **down slide** is the map that changes



and leaves the rest of the tableau unchanged. To describe a **right slide**, consider the four columns involved in the above configuration. (These form semistandard columns with respect to \mathcal{B} .) We let them to be denoted diagrammatically as

$$\begin{pmatrix} A_1 & B_1 \\ C_1 & D_1 \end{pmatrix}$$
 and $\begin{pmatrix} A_2 & B_2 \\ C_2 & D_2 \end{pmatrix}$.

If *r* is a primed entry, then a right slide involves transferring *r* from A_2 to B_1 , and then changing the left columns for the cosplit version of $\binom{B_1 \cup \{a\}}{C_1}$, and the right columns for the split version of $\binom{A_2-\{a\}}{D_2}$. If *r* is non-primed, then they are changed for the split version of $\binom{A_1}{D_1 \cup \{a\}}$ and the cosplit version of $\binom{B_2-\{a\}}{D_2-\{a\}}$, respectively. In both cases, the punctures are later placed in the same row but on the right two columns.

We slide the punctures as long as we can. Once we are done, we consider the unsplit version of the current tableau and add back its $\operatorname{King}_{\mathcal{A}'}$ part. We place i' where the puncture is, to its $\operatorname{King}_{\mathcal{A}'}$ part. We have produced a mixed tableau that has (at least) one more entry on its $\operatorname{King}_{\mathcal{A}'}$ part and one less entry on its De Concini part.

Now we look for the next entry to move and repeat the process until the tableau is semistandard with respect to \mathcal{A}' .

It is non-obvious that the map is well-defined (in particular, that the image is a King_{\mathcal{H}'} tableau). It is also non-obvious that jeu the taquin is weight-inverting. We refer to [She99] for proofs of these facts.

Example 8.22. We hope this example helps understanding this complicated definition. We let the $\operatorname{King}_{\mathcal{H}'}$ part of the tableau to be yellow shaded.

We start with the lowest entry with respect to $\leq_{\mathcal{B}}$, which is 3'.



We consider only the De Concini part to perform jeu de taquin. In this case, two down slides are performed.



And now restore the original tableau, remove the puncture, and add all necessary entries to its King part. We pick the next entry to move too.



We focus again on the De Concini part to perform jeu de taquin: we perform two right slides.



Upon restoring the tableau, the entry to move is again a 2'. One slide later, we are done.



Let us now describe the inverse map. Given a $\operatorname{King}_{\mathcal{H}'}$ tableau, we produce a De Concini tableau by using a modified jeu de taquin algorithm until it verifies the semistandard rules with respect to the alphabet \mathcal{B} . In its intermediate stages, we get mixed tableaux.

Definition 8.23 (Sheats⁻¹). To initialize the algorithm, let *T* be a $\text{King}_{\mathcal{A}'}$ tableau seen as a mixed tableaux with empty De Concini part.

Given a mixed tableau, let i' be the smallest value in its King part (with respect to $\leq_{\mathcal{B}}$). Every entry strictly smaller than i' (with respect to $\leq_{\mathcal{R}'}$) is added to its De Concini part. We also add the left-most instance of i' in the King part to its De Concini part, and puncture the De Concini part at this same location.

Now we perform jeu de taquin on the split version of the De Concini part of the tableau. It now has two punctures. At any given step of the jeu de taquin algorithm, the punctures will fall into the following configuration (where a and b might not exist).

If both *a* and *b* do not exist, stop for now. If only one of them exist, then we slide the puncture into it (by performing either an up slide or a left slide, which we define below). If both exist, we compare them. If $a \leq_{\mathcal{B}} b$, then we perform an up slide and vice versa.

An **up slide** is the inverse of a down slide, described above.

A left slide is the inverse of a right slide, also described above.

We slide the punctures as long as we can. Once we are done, we consider the unsplit version of the current tableau and add back its $\operatorname{King}_{\mathcal{A}'}$ part. We place i' where the puncture is, to its De Concini part. Importantly, if the column in which we just placed i' has an instance of i, then it too is to be added to the De Concini part.

We have produced a mixed tableau that has (at least) one more entry on its De Concini part and one less entry on its $King_{\mathcal{A}'}$ part.

Now we look for the next entry to move and repeat the process until the tableau is semistandard with respect to \mathcal{B} .

Example 8.24. Reverse all arrows in Example 8.22.

Chapter 9

A crystal structure on type C tableaux

Semistandard Young tableaux arise naturally when the representation theory of $\mathfrak{sl}(n)$ is studied via crystal theory; more precisely:

Theorem 9.1. The crystal basis for the finite dimensional irreducible representation $L(\lambda)$ of $\mathfrak{sl}(n)$ with highest weight λ is canonically identified (as a set) with the set of Far-Eastern readings of semistandard Young tableaux of shape λ .

We will introduce below some of the necessary definitions to understand the above statement in this chapter. More background on crystal bases is given in Chapter 10. The focus of this chapter is to understand the analogous statement for type C, Theorem 9.10. We will therefore not show the above statement; the interested reader may adapt our proof below or see [KN94, HK02, SV16].

When studying the representation theory of $\mathfrak{sp}(2n)$, Kashiwara's tableaux arise. However, because of our conventions (fixed in Chapter 2), which were the natural conventions to take for developing King's tableaux, some of our definitions will be slightly awkward. A more elegant development of the theory is seen in the literature on Kashiwara's tableaux [KN94, HK02, BS17].

We begin by analyzing the natural representation V of $\mathfrak{sp}(2n)$. This is the representation $\mathfrak{sp}(2n) \rightarrow \mathfrak{gl}(2n) = \mathfrak{gl}(V)$ defined by letting any matrix act by matrix multiplication. (In particular, $V = \mathbb{C}^{2n}$ as vector spaces.)

It is enough to understand this action on a system of generators of the Lie algebra. We gave such a system in Chapter 2. The only subspaces fixed by any matrix in the Cartan are the coordinate subspaces. More specifically, the only weight spaces of V are $V_{\pm \epsilon_i}$ for i = 1, ..., n, which are

$$V_{\epsilon_1} = \left\langle \begin{pmatrix} 1\\0\\0\\\vdots\\0 \end{pmatrix} \right\rangle, \quad V_{\epsilon_2} = \left\langle \begin{pmatrix} 0\\1\\0\\\vdots\\0 \end{pmatrix} \right\rangle, \quad \dots, \quad V_{\epsilon_n} = \left\langle \begin{pmatrix} 0\\\vdots\\0\\0\\\vdots\\0 \end{pmatrix} \right\rangle, \quad V_{-\epsilon_n} = \left\langle \begin{pmatrix} 0\\\vdots\\0\\1\\\vdots\\0 \end{pmatrix} \right\rangle, \quad \dots, \quad V_{-\epsilon_1} = \left\langle \begin{pmatrix} 0\\\vdots\\0\\1\\\vdots\\0 \end{pmatrix} \right\rangle.$$

The action of e_i takes V_{ϵ_i} to $V_{\epsilon_{i+1}}$ and $V_{-\epsilon_i}$ to $V_{-\epsilon_{i-1}}$, and vanishes on the rest of the weight spaces. On the other hand, e_0 takes $V_{-\epsilon_1}$ to V_{ϵ_1} . As usual, f_i does the opposite of what its corresponding e_i does.

Example 9.2. Let n = 2; that is, we work in $\mathfrak{sp}(4)$. Let $v \in \mathbb{C}^4$ be a generic vector. We have

$$e_1.v = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ & -1 & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = \begin{pmatrix} 0 & v_1 \\ 0 \\ -v_3 \end{pmatrix}, \quad e_0.v = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = \begin{pmatrix} v_4 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

We may draw the following diagram:



Each weight space is one dimensional. One may wish to obtain a basis for the representation such that (1) each basis element is a weight vector, (2) the actions of the different e_i permute the basis elements (or act as 0), and (3) the action of f_i is inverse to that of e_i in the basis. We call such a basis a crystal basis. In this chapter, we will assume existence of crystal bases for finite dimensional highest weight representations of $\mathfrak{sp}(2n)$, even though this statement is not totally correct. We make this a bit more precise in Chapter 10. For a detailed account of the theory of crystal bases, see [HK02] or the original papers [Kas94, KN94].

If we let i be the crystal basis element spanning $V_{\epsilon_{n+1-i}}$, for i = 1, ..., n, and we let i' be the crystal basis element spanning $V_{-\epsilon_{n+1-i}}$, then the crystal graph of the natural representation is

 $\boxed{1} \xrightarrow{1} \boxed{2} \xrightarrow{2} \cdots \xrightarrow{n-1} \boxed{n} \xrightarrow{n} \boxed{n'} \xrightarrow{n-1} \cdots \boxed{2'} \xrightarrow{1} \boxed{1'},$

where an arrow labeled *i* corresponds to the action of f_{n-i} .

Just for completeness, we include the combinatorial definition of a crystal [HK02, BS17].

Definition 9.3. Let g be a semisimple Lie algebra. Let $\Delta = \{\beta_1, ..., \beta_n\}$ be a set of simple roots. Let $X = \sum_{i=1}^n \mathbb{Z}\omega_i$ be the set of integral weights. A(n abstract) **crystal** is a set *B* together with maps

- $w: B \to X$,
- $\varepsilon_i, \varphi_i : B \to \mathbb{Z} \cup \{-\infty\}, \text{ for } i = 1, ..., n,$
- $E_i, F_i : B \to B \cup \{0\}$, for i = 1, ..., n,

satisfying the following axioms:

- (C1) $\varphi_i(b) = \varepsilon_i(b) + \langle \beta_i^{\vee}, w(b) \rangle$ for all i = 1, ..., n.
- (C2) $w(E_ib) = w(b) + \beta_i$, $w(F_ib) = w(b) \beta_i$, whenever E_ib (resp. F_ib) is in *B*.
- (C3) $\varepsilon_i(E_ib) = \varepsilon_i(b) 1$, $\varphi_i(F_ib) = \varphi_i(b) 1$, whenever E_ib is in B, $\varphi_i(E_ib) = \varphi_i(b) + 1$, $\varepsilon_i(F_ib) = \varepsilon_i(b) + 1$, whenever F_ib is in B.
- (C4) $F_i a = b$ if and only if $E_i b = a$, for all $a, b \in B$, for all i = 1, ..., n.
- (C5) if $\varphi_i(b) = -\infty$ for some $b \in B$ then $E_i b = 0 = F_i b$, and $\varepsilon_i(b) = -\infty$.

(Note that φ_i is completely determined by ε_i and w.) The crystal is furthermore called **seminormal** if one can write

 $\varepsilon_i(b) = \max\{k \ge 0 : E_i^k b \in B\} \text{ and } \varphi_i(b) = \max\{k \ge 0 : F_i^k b \in B\}.$

The **crystal graph** is the weighted digraph *G* with vertex set *B* and an edge $a \rightarrow b$ with weight *i* whenever $F_i a = b$.

Warning. Do not confuse the crystal map ε_i with the weight lattice element ϵ_i .

We will now check that the above graph defines an abstract crystal. (Note that we are not checking that this crystal structure has any representation-theoretic meaning; we omit this.) Recall that we take F_i to be the crystal operator corresponding to the action of f_{n-i} .

Proposition 9.4. The graph

$$\boxed{1} \xrightarrow{1} \boxed{2} \xrightarrow{2} \cdots \xrightarrow{n-1} \boxed{n} \xrightarrow{n} \boxed{n'} \xrightarrow{n-1} \cdots \boxed{2'} \xrightarrow{1} \boxed{1'},$$

is a crystal graph, and defines a seminormal crystal of $\mathfrak{sp}(2n)$ by setting $\beta_i := \epsilon_{n+1-i} - \epsilon_{n-i}$ for i = 1, ..., nand $\beta_n := 2\epsilon_1$, and

- $w[i] = \epsilon_{n+1-i}, \quad w[i'] = -\epsilon_{n+1-i},$
- $\varphi_i[j] = \delta_{ij} = \varepsilon_i[j']$ for i = 1, ..., n 1 and j = 1, ..., n,
- $\varphi_{i-1}[j'] = \delta_{ij} = \varepsilon_{i-1}[j]$ for i = 2, ..., n and j = 1, ..., n,
- $\varphi_n[j] = \delta_{nj} = \varepsilon_n[j'], \quad \varphi_n[j'] = 0 = \varepsilon_n[j] \quad for \ j = 1, ..., n.$

Example 9.5. We record the values of φ_i and ϵ_i for the seminormal crystal of the natural representation of $\mathfrak{sp}(4)$.

	$1 \frac{1}{-}$	$\rightarrow 2$	$\xrightarrow{2}$ 2' -	$\xrightarrow{1}$ 1'
p_1	1	0	1	0
p_2	0	1	0	0
1	0	1	0	1
2	0	0	1	0

Proof. Seminormality will follow from the definitions of φ and ε . We check the axioms.

- (C1) For fixed i = 1, ..., n 1 and j = 1, ..., n, we check $\varphi_i[j] = \varepsilon_i[j] + \langle \beta_i^{\vee}, w[j] \rangle$. Computing both sides of the equation, we get $\delta_{i,j}$ on the left hand side, and $\delta_{i+1,j} + (\delta_{i,j} \delta_{i+1,j})$ on the right hand side. A similar computation gives the result if b = [j']. For i = n, we note $\langle \beta_n^{\vee}, \varepsilon_j \rangle = \delta_{1j}$, which gives the result.
- (C2) For a fixed i = 1, ..., n 1, we check $w(E_ib) = w(b) + \beta_i$, whenever $E_ib \in B$. We note that this latter condition implies $b = \underbrace{i+1}_{i+1}$ or $\underbrace{i'}_{i}$. We compare w[i] with $w[i+1] + \beta_i$. That is, ϵ_{n+1-i} with $\epsilon_{n-i} + (\epsilon_{n+1-i} \epsilon_{n-i})$, which are equal. A similar computation gives the result if $b = \underbrace{i'}_{i}$. The result involving f_i is again shown similarly. For i = n, the desired result is just one tautological formula: $w[n'] = w[n] + 2\epsilon_1$.
- (C3) This statement is clear.
- (C4) This is true by hypothesis.
- (C5) This statement is void for this particular crystal.

Definition-theorem 9.6. Given two crystals B_1 and B_2 , we define the **tensor product** of B_1 and B_2 as the crystal with underlying set $B_1 \times B_2$ and maps given by

• $w(a \otimes b) = w(a) + w(b)$,

•
$$E_i(a \otimes b) = \begin{cases} E_i a \otimes b & \text{if } \varphi_i(a) \geq \varepsilon_i(b), \\ a \otimes E_i b & \text{if } \varphi_i(a) < \varepsilon_i(b), \end{cases}$$
 $F_i(a \otimes b) = \begin{cases} F_i a \otimes b & \text{if } \varphi_i(a) > \varepsilon_i(b), \\ a \otimes F_i b & \text{if } \varphi_i(a) \leq \varepsilon_i(b), \end{cases}$

 $\begin{array}{l} \bullet \ \varepsilon_i(a\otimes b) = \max\{\varepsilon_i(a), \ \varepsilon_i(b) - \langle \beta_i^{\vee}, w(a) \rangle\}, \\ \varphi_i(a\otimes b) = \max\{\varphi_i(b), \ \varphi_i(a) + \langle \beta_i^{\vee}, w(b) \rangle\}, \end{array}$

for i = 1, ..., n. (We write $a \otimes b$ for (a, b), and we have $a \otimes 0 = 0 = 0 \otimes b$.)

- Proof. We check well-definedness of the definition.
- (C1) We aim to show $\varphi_i(a \otimes b) = \varepsilon_i(a \otimes b) + \langle \beta_i^{\vee}, w(a \otimes b) \rangle$. That is,

$$\max\{\varphi_i(b), \varphi_i(a) + \langle \beta_i^{\vee}, w(b) \rangle\} = \max\{\varepsilon_i(a), \varepsilon_i(b) - \langle \beta_i^{\vee}, w(a) \rangle\} + \langle \beta_i^{\vee}, w(a) + w(b) \rangle.$$

Using axiom (C1) for B_1 and B_2 , the formula follows.

(C2) We check $w(E_i(a \otimes b)) = w(a \otimes b) + \beta_i$; the other formula is shown similarly. Unraveling the definitions and using (C2) for B_1, B_2 , the formula we seek is

$$\begin{cases} (w(a) + \beta_i) + w(b) = w(a) + w(b) + \beta_i & \text{if } \varphi_i(a) \ge \varepsilon_i(b), \\ w(a) + (w(b) + \beta_i) = w(a) + w(b) + \beta_i & \text{if } \varphi_i(a) < \varepsilon_i(b). \end{cases}$$

But this is tautological.

(C3) We check $\varepsilon_i(E_i(a \otimes b)) = \varepsilon_i(a \otimes b) - 1$; the other three formulas are shown similarly. Using (C3) for B_1, B_2 , we rewrite the formula we want to check as

$$\begin{aligned} \max\{\varepsilon_i(a) - 1, \ \varepsilon_i(b) - \langle \beta_i^{\vee}, w(a) \rangle\} & \text{if } \varphi_i(a) \ge \varepsilon_i(b), \\ &= \max\{\varepsilon_i(a), \ \varepsilon_i(b) - \langle \beta_i^{\vee}, w(a) \rangle\} - 1 \\ \max\{\varepsilon_i(a), \ \varepsilon_i(b) - 1 - \langle \beta_i^{\vee}, w(a) \rangle\} & \text{if } \varphi_i(a) < \varepsilon_i(b). \\ &= \max\{\varepsilon_i(a), \ \varepsilon_i(b) - \langle \beta_i^{\vee}, w(a) \rangle\} - 1 \end{aligned}$$

We now use (C1) for B_1, B_2 to note that $\varphi_i(a) \ge \varepsilon_i(b)$ if and only if $\varepsilon_i(a) \ge \varepsilon_i(b) - \langle \beta_i^{\vee}, w(a) \rangle$. The formula is therefore clear.

(C4) Let $F_i(a \otimes b) = c \otimes d$. We aim to check $E_i(c \otimes d) = a \otimes b$. The reciprocal is shown similarly. We have

$$E_i(c \otimes d) = \begin{cases} E_i c \otimes d & \text{if } \varphi_i(c) \ge \varepsilon_i(d), \\ c \otimes E_i d & \text{if } \varphi_i(c) < \varepsilon_i(d). \end{cases}$$

We distinguish further into more cases: $c \otimes d$ is either $F_i a \otimes b$ or $a \otimes F_i b$. We have

$$E_{i}(c \otimes d) = \begin{cases} E_{i}F_{i}a \otimes b & \text{if } \varphi_{i}(F_{i}a) \geq \varepsilon_{i}(b), \ \varphi_{i}(a) > \varepsilon_{i}(b), \\ E_{i}a \otimes F_{i}b & \text{if } \varphi_{i}(a) \geq \varepsilon_{i}(F_{i}b), \ \varphi_{i}(a) \leq \varepsilon_{i}(b), \\ F_{i}a \otimes E_{i}b & \text{if } \varphi_{i}(F_{i}a) < \varepsilon_{i}(b), \ \varphi_{i}(a) > \varepsilon_{i}(b), \\ a \otimes E_{i}F_{i}b & \text{if } \varphi_{i}(a) < \varepsilon_{i}(F_{i}b), \ \varphi_{i}(a) \leq \varepsilon_{i}(b). \end{cases}$$

We may now use (C3) to simplify the cases. For instance, the second case becomes $\varphi_i(a) \ge \varepsilon_i(b) + 1$, $\varphi_i(a) \le \varepsilon_i(b)$, which is impossible. Similarly, the third case is impossible. But in the first and last cases, we have $E_iF_ia \otimes b = a \otimes b = a \otimes E_iF_ib$ (using (C4) for B_1, B_2). This gives the result.

(C5) If $\varphi_i(a \otimes b) = -\infty$, we deduce $\varphi_i(a) = \varphi_i(b) = -\infty$, and the result now follows using (C5) for B_1, B_2 .

Lemma 9.7. If B_1 and B_2 are seminormal crystals, then $B_1 \otimes B_2$ is seminormal.

Proof. Let K be defined as $\max\{k \ge 0 : E_i^k(a \otimes b) \in B\}$. We aim to show $K = \max\{\varepsilon_i(a), \varepsilon_i(b) - \langle \beta_i^{\vee}, w(a) \rangle\}$. Using axiom (C1), we may rewrite this as

$$K = \varepsilon_i(a) + \max\{0, \varepsilon_i(b) - \varphi_i(a)\}$$

Suppose first that $\varphi_i(a) \ge \varepsilon_i(b)$. Since B_1 is seminormal, $\varphi_i(e^k a) = \varphi_i(a) + k \ge \varepsilon_i(b)$ for all $k \ge 0$. Therefore, $E_i^K(a \otimes b) = E_i^K(a) \otimes b$ and thus $K = \varepsilon_i(a)$. Suppose now that $\varphi_i(a) < \varepsilon_i(b)$. Then $E_i^k(a \otimes b) = E_i^{k-1}(a \otimes E_i(b))$. Since B_2 is seminormal,

Suppose now that $\varphi_i(a) < \varepsilon_i(b)$. Then $E_i^{\kappa}(a \otimes b) = E_i^{\kappa-1}(a \otimes E_i(b))$. Since B_2 is seminormal, $\varphi_i(a) < \varepsilon_i(b) + k = \varepsilon_i(E_i^{k}(b))$ while $k \le \varepsilon_i(b) - \varphi_i(a)$. When applying the operator E_i to $E_i^{\varepsilon_i(b) - \varphi_i(a)}(a \otimes b) = a \otimes E_i^{\varepsilon_i(b) - \varphi_i(a)}(b)$, we act on the first factor by definition. This brings us back to the first situation, which in turn gives the formula $K = \varepsilon_i(a) + \varepsilon_i(b) - \varphi_i(a)$.

We are now almost ready to describe the crystal structure on Kashiwara's tableaux. But first, we need to introduce readings.

Definition 9.8. Given a tableau *T*, the **Far-Eastern reading** of *T* is the formal tensor of its entries, going column by column, top to bottom, and right to left.

Example 9.9. Here is a tableau and its Far-Eastern reading.

$$\begin{array}{c|c}1&2&5\\\hline 3&4\end{array}\mapsto 5\otimes 2\otimes 4\otimes 1\otimes 3.$$

Since every irreducible representation of $\mathfrak{sp}(2n)$ appears as a subrepresentation of some tensor power of the natural representation, one can interpret weight vectors of any given representation as some tensor of the weight vectors of the natural representation. We get the following theorem [HK02, Thm. 8.3.3], [BS17, Thm. 6.10], [KN94, Thm. 4.4.3].

Theorem 9.10. The crystal basis for the finite dimensional irreducible representation $L(\lambda)$ of $\mathfrak{sp}(2n)$ with highest weight λ is canonically identified (as a set) with the set of Far-Eastern readings of Kashiwara's tableaux of shape λ .

Proof. Since $L(\lambda)$ is a subrepresentation of $V^{\otimes N}$ for $N = |\lambda|$, the crystal graph of $L(\lambda)$ appears as a connected component of the crystal graph of $V^{\otimes N}$ [HK02, Thm. 4.2.10], which we now know how to compute. More specifically, the crystal graph for $L(\lambda)$ will be the connected component generated by its highest weight vector, which we aim to identify with the Kashiwara tableau $T_{\lambda} : [\lambda] \to C$ given by T(i, j) = i. For instance, for $\lambda = (3, 3, 2), n = 3$,

$$T_{\lambda} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 \end{bmatrix}.$$

So it will suffice to show that the set of Kashiwara's tableaux of shape λ is stable under the action of F_i and E_i for all *i* (when the action doesn't vanish), and that it has a unique highest weight vector T_{λ} .

Let us show stability. Let *T* be a Kashiwara tableau of shape λ . Suppose $F_iT \neq 0$, $i \neq n$. Therefore, *T* and F_iT differ either by an entry $x = \boxed{i}$ turned into $\boxed{i+1}$, or by a entry $x = \boxed{i+1'}$ turned into $\boxed{i'}$. If i = n, then *T* and F_iT differ by an entry $x = \boxed{n}$ turned into $\boxed{n'}$.

CLAIM 1. The tableau $F_i T$ is semistandard in the alphabet $C = \{1 < \cdots < n < n' < \cdots < 1'\}$.

Proof of claim. For the purpose of this proof, we will assume *T* and F_iT differ either by an entry x = [i] turned into [i+1]. The other cases are shown similarly.

Let *r* be the entry immediately to the right of *x*, if it exists; let *d* be the entry immediately below *x*, if it exists. We introduce some notation for the entries of *T* and two subreadings of the Far-Eastern reading of *T*;

$$a_1 \otimes \cdots \otimes a_k \otimes r \otimes \underbrace{b_1 \otimes \cdots \otimes b_l \otimes x \otimes \overrightarrow{d \otimes c_1 \otimes \cdots \otimes c_m}}_R.$$

Schematically, *R* and *D* define the following two highlighted parts of the tableau:

$$T = \begin{bmatrix} x & r \\ d \end{bmatrix} \implies R = \begin{bmatrix} x & r \\ d \end{bmatrix} \text{ and } D = \begin{bmatrix} x & r \\ d \end{bmatrix}$$

By hypothesis, F_i takes x = [i] to [i+1]. Therefore, $\varphi_i(r) \le \varepsilon_i(R)$ and $\varphi_i(x) > \varepsilon_i(D)$. Since x = [i], and therefore $\varphi_i(x) = 1$, we deduce $\varepsilon_i(D) = 0$.

Suppose now that F_iT is not semistandard. Since *T* is semistandard, we must have either r = [i] or d = [i+1]. Let us discuss these two cases separately.

• If d = [i+1], then

$$\begin{split} \varepsilon_i(D) &= \varepsilon_i(\overbrace{i+1} \otimes c_1 \otimes \cdots \otimes c_m) \\ &= \max\{\varepsilon_i \overbrace{i+1}, \ \varepsilon_i(c_1 \otimes \cdots \otimes c_m) - \langle \beta_i^{\vee}, w \overbrace{i+1} \rangle\} \\ &\geq \varepsilon_i(c_1 \otimes \cdots \otimes c_m) - \langle \epsilon_{n+1-i} - \epsilon_{n-i}, -\epsilon_{n-i} \rangle \geq 1. \end{split}$$

This is a contradiction to the statement above.

• If r = [i], then $\varphi_i(r) = 1$. We assume firstly that none of the entries $b_1, ..., b_l$ are in the set $\{[i], [i+1], [i'], [i+1']\}$. We will discuss later why we can make such an assumption. With this, we get $\varepsilon_i(b_j) = 0 = \langle \beta_i^{\vee}, w(b_j) \rangle$ for all j, which means

$$\begin{split} \varepsilon_i(R) &= \varepsilon_i(b_1 \otimes \cdots \otimes b_l \otimes [i] \otimes D) \\ &= \max\{\varepsilon_i(b_1), \ \varepsilon_i(b_2 \otimes \cdots \otimes b_l \otimes [i] \otimes D) - \langle \beta_i^{\vee}, w(b_1) \rangle\} \\ &= \varepsilon_i(b_2 \otimes \cdots \otimes b_l \otimes [i] \otimes D) \\ &= \cdots = \varepsilon_i([i] \otimes D) \\ &= \max\{\varepsilon_i[i], \ \varepsilon_i(D) - \langle \beta_i^{\vee}, w[i] \rangle\} = \max\{0, \ 0-1\} = 0. \end{split}$$

This contradicts the above inequality $\varphi_i(r) \leq \varepsilon_i(R)$.

Let us discuss why $b_j \notin \{[i], [i+1], [i'], [i+1']\}$ for all j = 1, ..., l. If $b_j = [i]$ for some j, then T would not be semistandard. If $b_j = [i+1]$ for some j, we note j must be equal to 1. But then, d must be equal to [i+1] too, and we get a contradiction as above. If $b_j = [i+1']$ for some j, then we would get

$$\varepsilon_{i}(b_{j} \otimes \cdots \otimes D) = \max\{\varepsilon_{i} \boxed{i+1'}, \varepsilon_{i}(b_{j+1} \otimes \cdots \otimes D) - \langle \beta_{i}^{\vee}, w \boxed{i+1'}\} \\ = \max\{0, \varepsilon_{i}(b_{j+1} \otimes \cdots \otimes D) - 1\}.$$

Therefore, the value of $\varepsilon_i(b_j \otimes \cdots \otimes D)$ would still be 0 by the above computation applied to $\varepsilon_i(b_{j+1} \otimes \cdots \otimes D)$. Finally, if $b_j = \boxed{i'}$ for some *j*, we find the following configuration in *T*:

This contradicts axiom (K2) for T.

CLAIM 2. The tableau F_iT is admissible.

Proof of claim. We only need to check admissibility of the column in which x lies. So we may assume in the following that T is a column tableau.

If i = n the statement is clear. So let $i \neq n$.

Assume firstly that *T* and F_iT differ by an entry x = [i] turned into [i+1]. This preserves admissibility in the column.

Assume now that *T* and F_iT differ by an entry x = i + 1' turned into i'.

If there is no entry i in the column of x, then turning x into i' preserves admissibility. Assume therefore that there is such an entry in the column, and write $T = A \otimes i \otimes B \otimes x \otimes C$, where $A = a_1 \otimes \cdots \otimes a_k$, $B = b_1 \otimes \cdots \otimes b_l$, and $C = c_1 \otimes \cdots \otimes c_m$ are subreadings of T. We note $b_j \notin \{i, i+1'\}, i'\}$ for any j = 1, ..., l, since the first two entries would contradict T being weakly increasing, and the third entry would contradict $F_i T$ being weakly increasing. If $b_1 = i+1$, we note admissibility is preserved when turning x into i'. So, we assume $b_j \notin \{i, i+1'\}, i'\}, i+1\}$.

Since c_1 is easily seen to be not be in $\{[i], [i+1'], [i'], [i+1]\}$, we get $\varepsilon_i(x \otimes C) = 0$.

Since F_i acts on x, we must have $1 = \overline{\varphi_i}[\underline{i}] \leq \varepsilon_i (\overline{B \otimes x \otimes C})$ and $1 = \varphi_i(x) > \varepsilon_i(C) = 0$. Finally, a computation similar to those of the proof of the previous claim yields $\varepsilon_i(B \otimes x \otimes C) = \varepsilon_i(x \otimes C) = 0$, giving a contradiction.

We would now like to check that $cosplit(F_iT)$ is semistandard. However, although more visual, such a proof involves a big amount of case-checking. Instead, we could show that F_iT satisfies axiom (K2'). This the approach taken in [HK02, BS17]. However, we have found their arguments incomplete. We instead show (K2) as in the original paper [KN94].

By definition of the cosplit algorithm, F_nT being semistandard implies $cosplit(F_nT)$ also being semistandard.

CLAIM 3. The tableau $F_i T$ verifies (K2), for i = 1, ..., n - 1.

Proof of claim. Suppose F_iT has the following configuration, the other one can be analyzed similarly:

$$\begin{array}{cccc} p \to & a \\ q \to & & b \\ r \to & & b' \\ s \to & & a' \end{array}$$

Then *T* must have one of the following two configurations:

$p \rightarrow$	а		a-1	а		a		а	
$q \rightarrow$		b		b	b-1		b		b
$\hat{r} \rightarrow$		b'		b'	b'	b	r+1'		b'
$s \rightarrow$		a'		a′	a'		a'		a+ 1'

If i < b, then the configuration of F_iT verifies (K2) from the analogous property of T. Assume $i \ge b$. We will treat all cases simultaneously except for the second one, which is slightly more delicate. We therefore assume we don't have the second configuration, for now.

Without loss of generality, *T* has two columns. Let us now restrict our attention to the entries of *T* that in rows *p*, *p*+1, ..., *s*-1, *s*. This gives us a new tableau *P* in the alphabet $\{a < \cdots < n < n' < \cdots < a'\}$.

We know *T* is admissible, which means each column has a higher concentration of high numbers than low numbers. In equations, if the length of the column is *N*, then for each pair of entries x at row *k* and $\overline{x'}$ at row *l*, we get N - (l - k) < n - x. The property of admissibility is inherited by *P*.

By a previous claim, the columns of $F_i P$ must also be admissible. In equations, for each pair of entries b at row q and b' at row r, we get (s-p)-(r-q) < b-a, which is exactly what we wanted to show.

If *T* were to have the second configuration above, we proceed similarly, but changing the definition of *P*. Since $F_i = F_b$ affects the first column of *T* and not the second one, we deduce that there is an entry $\boxed{a - 1'}$ in the second column of *T*, which lies in row s + 1.

$$\begin{array}{c|ccc} p \rightarrow & a-1 & & \\ q \rightarrow & & & b' \\ r \rightarrow & & b' \\ s \rightarrow & & a' \\ s+1 \rightarrow & & a-1 \end{array}$$

Our tableau *P* will be defined by rows p, p + 1, ..., s, s + 1. The same analysis as before gives the desired inequality.

This concludes the first part of the proof. It remains to show that the crystal has a unique highest weight vector and that it is T_{λ} . (Note that $E_i T_{\lambda} = 0$ for all *i* and therefore it is a highest weight vector.)

Assume there is a different highest weight vector *T*. Let *k* be the biggest integer such that *T* and T_{λ} agree on the first *k* rows. (We therefore assume $k \neq l(\lambda)$.) Let *x* be the rightmost entry in row k + 1 of *T*. Necessarily, $x \neq k+1$. We may introduce some notation for the entries of *T*;

$$\underbrace{a_1 \otimes \cdots \otimes a_k}_A \otimes x \otimes \underbrace{b_1 \otimes \cdots \otimes b_l}_B \otimes \underbrace{c_1 \otimes \cdots \otimes c_m}_C,$$

where *B* is the rest of the column in which *x* lies. Schematically, we are breaking up the tableau in the following parts:



In particular, note that *A* is a tableau in which each entry in row *j* is j, for j = 1, ..., k. Consequently, $E_i(A) = 0$ for all i = 1, ..., n. We may also compute $\varphi_i(A) = 0$ for all $i \ge k + 1$.

Fix $i \ge k + 1$. We have $E_i(A \otimes x \otimes B \otimes C) = 0$, which combined with the tensor product formula for E_i and the above implies $E_i(A \otimes x) = 0$. Thus $E_i(x) = 0$. This means $x \notin \{ \underbrace{i+1}, \underbrace{i'}\}$. Since this is true for i = k + 1, ..., n, we conclude x is either i for i < k (which is not possible, since T is semistandard), or i' for $i \le k$. But if x = i' with $i \le k$, then $E_i(A \otimes x) \neq 0$. In any case, we reach a contradiction.

Therefore, T_{λ} is the only highest weight vector, finishing the proof.

This concludes the description of the crystal structure of Kashiwara's tableaux. However, the question still remains to describe the crystal structure on King's tableaux. For each i = 1, ..., n, we have a diagram



which we desire to make into a commutative square. Note that, since the bijection between the two classes of tableaux is weight-inverting (and shape-preserving), these missing maps will realize an S_n

action of the set of King's tableaux of a fixed shape. For the purpose of understanding the missing maps, our code (see Appendices B and D) applies crystal operators on King's tableaux by taking the composite of the three other maps.

Note 9.11. Crystal bases are implemented in SageMath [Sage] as Kashiwara's tableaux. However, two aspects of the implementation differ from what it is usually written in a textbook. Firstly, the tableaux are read by the *inverse* Far-Eastern reading. Secondly, the product rule for crystals is altered with respect to the classical rule. The rule implemented for the tensor $a \otimes b$ turns out to be describing the product rule for $b \otimes a$ instead. Consequently, both changes "cancel out", resulting in their tableaux being exactly Kashiwara's tableaux.

Chapter 10

The Lie theoretic and combinatorial definitions coincide

Our goal in this Chapter is to give a proof of the following statement.

Theorem 10.1. Let $n \ge 1$. Let λ be a partition, $l(\lambda) \le n$. Then, the character of the irreducible representation $L(\lambda)$ of $\mathfrak{sp}(2n)$ of highest weight λ agrees with the generating function of King's symplectic tableaux of shape λ on n letters; that is,

$$\chi_{\lambda}^{\mathfrak{sp}(2n)}(x_1,...,x_n) = sp_{\lambda}(x_1,...,x_n).$$

Note that we already know

$$sp_{\lambda}(x_1,...,x_n) \stackrel{6.12}{=} D_{\lambda}(x_1,...,x_n,x_1^{-1},...,x_n^{-1}) \stackrel{7.1}{=} \chi_{\lambda}^{\mathfrak{sp}(2n)}(x_1,...,x_n).$$

We present now, nevertheless, a distinct proof, relying on the theory of quantum groups and crystal bases, which we quickly sketch here. We refer to [HK02] for a proper treatment of the subject.

As we said in Chapter 9, given a semisimple Lie algebra \mathfrak{g} and a representation V of \mathfrak{g} , we would like to find a basis of V such that (1) each basis element is a weight vector, (2) the actions of the different e_i permute the basis elements (or act by 0), and (3) the action of f_i is inverse to that of e_i in the basis. However, we don't know if such a basis exists or how to construct it. In the following, we sketch how to construct a set that, although not a basis of V, has properties analogous to (1), (2), and (3).

We know a g-representation is just a $U(\mathfrak{g})$ -module, where $U(\mathfrak{g})$ denotes the universal enveloping algebra of \mathfrak{g} . One can consider deformations of $U(\mathfrak{g})$, by which we mean be Hopf algebras $U_q(\mathfrak{g})$ depending on a parameter $q \in \mathbb{C} - \{0\}$. We do this in such a way that $U_1(\mathfrak{g})$ recovers our original universal enveloping algebra.

Furthermore, it is possible to require the dimension of weight spaces to be invariant under the deformation. That is, if one has a $U_q(\mathfrak{g})$ -module V^q with a weight space decomposition $V^q = \bigoplus_{\lambda} V_{\lambda}^q$, then we can require dim V_{λ}^q to be independent of q.

then we can require $\dim V_{\lambda}^{q}$ to be independent of q. This allows us to write $\operatorname{ch} V^{1} = \sum_{\lambda} \dim V_{\lambda}^{1} x^{\lambda} = \sum_{\lambda} \dim V_{\lambda}^{q} x^{\lambda} = \operatorname{ch} V^{q}$ for any $U_{q}(\mathfrak{g})$ -module V^{q} , for any $q \in \mathbb{C} - \{0\}$.

Taking a limit¹ $q \rightarrow 0$, we find a basis of the $U_q(\mathfrak{g})$ -module with properties (1), (2), and (3) (were e_i and f_i are exchanged by suitable operators in the limit). This is what we call a crystal basis of V^q . In particular, a crystal basis is a seminormal abstract crystal in the sense of Definition 9.3, where

¹We refer to [HK02] for the precise definition of a limit in this context.

- $\Delta = \{\beta_1, ..., \beta_n\}$ is the a of simple roots of \mathfrak{g} ,
- *w* is the limit of the weight function of $U_q(\mathfrak{g})$ (which does not depend on *q*), and
- the set of crystal operators $\{E_i\}_{1 \le i \le n}$ and $\{F_i\}_{1 \le i \le n}$ have actions analogous to those of the sets of elements $\{e_i\}_{1 \le i \le n}$ and $\{f_i\}_{1 \le i \le n}$ of \mathfrak{g} , for each *i*.

Given these properties, if *B* is a crystal basis of $L(\lambda)^q$, then we can write

$$\operatorname{ch} L(\lambda) = \sum_{\mu \in \mathfrak{h}^*} \dim L(\lambda)_{\mu} x^{\mu} = \sum_{b \in B} x^{w(b)}.$$

We now attempt to compute the monomial weight of a crystal basis element.

Type A

Recall Theorem 9.1, which canonically identified a crystal basis of a representation $L(\lambda)$ of $\mathfrak{sl}(n)$ with the set $SSYT_n(\lambda)$.

We will also use the following results, which we present without proof. Note that, in the literature, distinct reformulations of these statements are found, due to differences in the conventions of Chapter 2.

Proposition 10.2. The crystal graph of the natural representation V of $\mathfrak{sl}(n)$ is

$$\boxed{n} \xrightarrow{n-1} \cdots \xrightarrow{2} \boxed{2} \xrightarrow{1} \boxed{1},$$

where F_i and E_i are given by f_i and e_i , and where $w_j = \alpha_j$.

Lemma 10.3. Let λ be a partition, $l(\lambda) \leq n - 1$. Let $L(\lambda)$ be the irreducible representation of $\mathfrak{sl}(n)$ of highest weight λ . The unique highest weight vector of $L(\lambda)$ is identified, via Theorem 9.1, with the tableau $T_{\lambda} : [\lambda] \rightarrow [n]$ given by T(i, j) = n + 1 - i.

Let $\mu(T) \in \sum_{i=1}^{n-1} \mathbb{Z}\epsilon_i$ denote the weight of a tableau *T* in the Lie theoretic sense. That is, the element of the weight lattice such that $x^T = x^{\mu(T)}$.

We note that T_{λ} is of weight $\lambda_1 \epsilon_{n-1} + \lambda_2 \epsilon_{n-2} + \cdots + \lambda_n \epsilon_1$. That is, $\mu(T_{\lambda}) = \lambda$, using the bijection of Theorem 2.12.

Then, we note that, for any $i \in [n]$ and any tableau $T \in SSYT_n(\lambda)$ such that $F_iT \neq 0$, we have that T and F_iT differ by an entry i+1 turned into i. Therefore, $x^{F_iT} = x_{i+1}^{-1}x_i \cdot x^T$. On the other hand, $F_i \in \mathfrak{sl}(n)$ takes a μ -weight vector to a $(\mu - \alpha_i)$ -weight vector, with $\alpha_i = \epsilon_{i+1} - \epsilon_i$. That is, $\mu(F_iT) = F_i \cdot \mu(T) = \mu(T) - \alpha_i$.

We have shown the following lemma.

Lemma 10.4. Let λ be a partition, $l(\lambda) \leq n - 1$. Let $L(\lambda)$ be the irreducible representation of $\mathfrak{sl}(n)$ of highest weight λ . Let B be a crystal basis of $L(\lambda)$. A crystal basis element $b \in B$ identified with the tableau T via Theorem 9.1 has weight $w(b) = \mu(T)$.

Altogether, we can show the desired theorem.

Theorem 10.5. Let $n \ge 1$, let λ be a partition. Then, the character of the irreducible representation of $\mathfrak{sl}(n)$ indexed by λ agrees with the generating function of semistandard Young tableaux of shape λ on n letters; that is,

$$\chi_{\lambda}^{\mathfrak{sl}(n)}(x_1,...,x_n) = s_{\lambda}(x_1,...,x_n).$$

Proof. Let *B* be a crystal basis of $L(\lambda)$. We have

$$\operatorname{ch} L(\lambda) = \sum_{b \in B} x^{w(b)} = \sum_{T \in \operatorname{SSYT}_n(\lambda)} x^{\mu(T)} = \sum_{T \in \operatorname{SSYT}_n(\lambda)} x^T = s_{\lambda}.$$

Type C

Similar to the type A case, we use Proposition 9.4 and Theorem 9.10, which allow us to identify the basis of the natural representation and any given irreducible representation of $\mathfrak{sp}(2n)$ with a crystal basis, respectively. We also use the fact that the tableau $T_{\lambda}(i, j) = i$ is identified with the highest weight vector of $L(\lambda)$, for any fixed λ . We showed this in the proof of Theorem 9.10.

We know that T_{λ} is of weight $x_1^{\lambda_1} \cdots x_n^{\lambda_n}$. That is, $\mu(T_{\lambda}) = \lambda_1 \epsilon_1 + \cdots + \lambda_n \epsilon_n$. According to Theorem 2.18, this doesn't correspond to the partition λ , but to its *reverse* $(\lambda_n, ..., \lambda_1)$, which we denote by $rev(\lambda)$. On the other hand, following Proposition 9.4, this tableau is identified with a crystal basis element of weight λ .

Lemma 10.6. Let λ be a partition, $l(\lambda) \leq n$. Let $L(\lambda)$ be the irreducible representation of $\mathfrak{sp}(2n)$ of highest weight λ . Let B be a crystal basis of $L(\lambda)$. A crystal basis element $b \in B$ identified with the tableau T via Theorem 9.10 has weight $w(b) = \operatorname{rev}(\mu(T))$.

Proof. We proceed by induction. For the highest weight element $b \in B$, we have shown it above.

Assume that for a given $b \in B$ and a given $i \in [n]$ we have $E_iT \neq 0$ and $w(E_ib) = -\mu(E_iT)$. If i = n, then E_iT differs from $F_iE_iT = T$ in a single entry [n] turned into [n']. Then, $\mu(T) = \mu(E_nT) - \epsilon_n + (-\epsilon_n) = \mu(E_nT) - 2\epsilon_n$. On the other hand, $w(b) = w(F_nE_nb) = w(E_nb) - \beta_n = \operatorname{rev}(\mu(E_nT)) - 2\epsilon_1 = \operatorname{rev}(\mu(E_nT) - 2\epsilon_n)$. Thus, $w(b) = \operatorname{rev}(\mu(T))$.

If $i \neq 0$, then *T* differs from E_iT on an entry i into i+1, or an entry i+1' turned into i'. In either case, we get $\mu(T) = \mu(E_iT) - \epsilon_i + \epsilon_{i+1} = \mu(E_iT) + \alpha_i$. On the other hand, $w(b) = w(F_iE_ib) = w(E_ib) - \beta_i = \operatorname{rev}(\mu(E_iT) + \alpha_i)$, as desired.

We can now proof the main theorem of this Chapter.

Proof of Thm. 10.1. Let *B* be a crystal basis of $L(\lambda)$. We have

$$\operatorname{ch} L(\lambda) = \sum_{b \in B} x^{w(b)} = \sum_{\substack{T \text{ Kashiwara} \\ \text{tab. of shape } \lambda \\ \text{on } n \text{ letters}}} x^{\operatorname{rev}(\mu(T))}.$$

Since Sheat's bijection gives a weight-inverting bijection from the set of De Concini's tableaux and the set of King's tableaux (of fixed shape and number of letters), and the bijection from De Concini's and Kashiwara's tableaux is weight reversing, we get

$$\operatorname{ch} L(\lambda) = \sum_{T \in \mathrm{KSpT}_n(\lambda)} x^T = sp_{\lambda}(x_1, ..., x_n).$$

Appendices

Appendix A

Type C Bender-Knuth involutions

To complete the proof of Proposition 5.9, we need to verify that the proposed map BK_2^C is an involution. We recall that BK_2^C is defined on a Gelfand–Tsetlin pattern with 6 rows as the following composite:

$$T_0 \xrightarrow{BK_4^A} T_1 \xrightarrow{BK_3^A} T_2 \xrightarrow{BK_5^A} T_3 \xrightarrow{BK_4^A} T_4 \xrightarrow{\text{rect.}} T_5, \tag{A.1}$$

where the first four maps are type A Bender–Knuth involutions. The *j*th type A Bender–Knuth involution, we recall, acts on every entry of the *j*th row of a Gelfand–Tsetlin pattern *x*, taking

$$x_{i,j}$$
 to $\min\{x_{i,j+1}, x_{i-1,j-1}\} + \max\{x_{i+1,j+1}, x_{i,j-1}\} - x_{i,j}$.

The rectification map acts on x_{35} , x_{24} , x_{23} and x_{34} by subtracting x_{34} .

Let $T_0 = (a^{(6)}, ..., a^{(1)})$. We introduce the following notation for the entries of the intermediate patterns in the composite:

a_{16} a_{26} a_{36} 0 0 0	a_{16} a_{26} a_{36} 0 0 0
a_{15} a_{25} a_{35} 0 0	a_{15} a_{25} a_{35} 0 0
$a_{14} a_{24} 0 0$	$2^{\prime} 3) \qquad b_{14} b_{24} b_{34} 0$
a_{13} a_{23} 0	$a_{13} a_{23} 0$
$a_{12} = 0$	$a_{12} = 0$
a_{11}	a_{11}
a_{16} a_{26} a_{36} 0 0 0	a_{16} a_{26} a_{36} 0 0 0
a_{15} a_{25} a_{35} 0 0	d_{15} d_{25} d_{35} 0 0
$(2 2') \qquad b_{14} b_{24} b_{34} 0$	$(3 3') \qquad b_{14} b_{24} b_{34} 0$
c_{13} c_{23} 0	c_{13} c_{23} 0
$a_{12} = 0$	$a_{12} = 0$
a_{11}	a_{11}
a_{16} a_{26} a_{36} 0 0 0	a_{16} a_{26} a_{36} 0 0 0
d_{15} d_{25} d_{35} 0 0	d_{15} d_{25} f_{35} 0 0
$(3 2')$ e_{14} e_{24} e_{34} 0	rect. $e_{14} f_{24} 0 0$
c_{13} c_{23} 0	c_{13} f_{23} 0
$a_{12} = 0$	$a_{12} = 0$
a_{11}	a_{11}

So, for instance, $b_{16} = a_{16}$ and therefore it is omitted, b_{34} is defined to be $\min\{a_{35}, a_{23}\}$, and f_{35} is defined to be $d_{35} - e_{34}$. We also let e.g. T_5 be $((a_{16}, ..., 0), (d_{15}, ..., 0), ..., (a_{11}))$. If instead, T_0 is changed for e.g. $(A^{(6)}, ..., A^{(1)})$, then $B_{16}, B_{26}, ..., F_{11}$ are defined the obvious way. We may rewrite Equation (A.1) as

$$a \xrightarrow{BK_4^A} b \xrightarrow{BK_3^A} c \xrightarrow{BK_5^A} d \xrightarrow{BK_4^A} e \xrightarrow{\text{rect.}} f$$

To show that BK_2^C is an involution, we will compare, step by step, the entries produced by two different composites:

$$\Psi : a \xrightarrow{(2'3)} b \xrightarrow{(2'2')} c \xrightarrow{(3'3')} d \xrightarrow{(2'3)} e =: A \xrightarrow{(2'3)} B \xrightarrow{(2'2')} C \xrightarrow{(3'3')} D \xrightarrow{(2'3)} E = a,$$

$$(BK_{2}^{C})^{2} : a \xrightarrow{(2'3)} b \xrightarrow{(2'2')} c \xrightarrow{(3'3')} d \xrightarrow{(2'3)} e \xrightarrow{\text{rect.}} A' \xrightarrow{(2'3)} B' \xrightarrow{(2'2')} C' \xrightarrow{(3'3')} D' \xrightarrow{(2'3)} E' \xrightarrow{\text{rect.}} F'$$

The second composite is created by ignoring the rectification maps from the first composite. We know Ψ is the identity, since Bender–Knuth involutions are involutions, and the Bender–Knuth involutions corresponding to (2 2') and (3 3') commute.

The first four maps of both composites are identical. The fifth pattern in each composite was relabeled to A, resp. A', according to the above convention. In the following, we will express the entries of A', B', ..., F' in terms of the entries of A, B, ..., E.

Let us start by comparing A and A'. We have $A'_{jk} = A_{jk}$ for all j, k except for

$$A'_{35} = A_{35} - e_{34}, \quad A'_{24} = A_{24} - e_{34}, \quad A'_{23} = A_{23} - e_{34}, \quad \text{and} \quad A'_{34} = A_{34} - e_{34} = 0$$

We may now turn to B and B', in which we thus find

$$B'_{14} = B_{14}, \quad B'_{24} = B_{24}, \quad \text{and} \quad B'_{34} = B_{34}$$

Indeed, we have

$$\begin{split} B'_{24} &= \min\{A'_{25}, A'_{13}\} + \max\{A'_{35}, A'_{23}\} - A'_{24} \\ &= \min\{A_{25}, A_{13}\} + \max\{A_{35} - e_{34}, A_{23} - e_{34}\} - (A_{24} - e_{34}) = B_{24} \end{split}$$

and

$$B'_{34} = \min\{A_{35} - e_{34}, A_{23} - e_{34}\}$$

= min{A₃₅, A₂₃} - e₃₄
= min{A₃₅, A₂₃} - A₃₄ = B₃₄

In the next step, when comparing C and C', we therefore note

$$C'_{13} = C_{13}$$
, and $C'_{23} = C_{23} + e_{34}$

Similarly, in D, D',

$$D'_{15} = D_{15}, \quad D'_{25} = D_{25}, \text{ and } D'_{35} = D_{35} + e_{34}.$$

Finally, comparing E and E' gives

$$E'_{14} = E_{14}, \quad E'_{24} = E_{24} + e_{34}, \quad \text{and} \quad E'_{34} = E_{34} + e_{34} = a_{34} + e_{34} = e_{34}.$$

And now, subtracting e_{34} from E'_{34} , E'_{24} , D'_{35} and C'_{23} recovers the pattern *E*. This shows F' = E = a, as desired.

Note A.2. Just for illustrative purposes, we give explicitly give the patterns A', B', ..., F' in terms of A, B, ..., C according to the computations above. To save space, we denote $x - e_{34}$ by x^- and $x + e_{34}$ by x^+ .

$A_{16} \ A_{26} \ A_{36} \ 0 \ 0 \ 0$	B_{16} B_{26} B_{36} 0 0 0
$A_{15} A_{25} A_{35}^- 0 \qquad 0$	B_{15} B_{25} B_{35}^- 0 0
$A' = \begin{array}{ccc} A_{14} & A_{24}^- & A_{34}^- & 0 \\ A_{13} & A_{23}^- & 0 \end{array} \qquad \qquad$	
$A_{12} = 0$	$B_{12} = 0$
A_{11}	B_{11}
$C_{16} C_{26} C_{36} 0 0 0$ $C_{15} C_{25} C_{35}^{-} 0 0$ $(2 \ 2') \qquad C' = \begin{array}{c} C_{14} C_{24} C_{34} 0 \\ C_{13} C_{23}^{+} 0 \\ C_{12} 0 \\ C_{11} \end{array}$	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$

Appendix B

Crystal reflections and Bender–Knuth

The following diagram of symplectic tableaux,



does not commute, even for row or column tableaux. (The horizontal maps are the composites defined in Chapter 8.)

For instance, consider the action of BK_2^C on the following King tableau *T*.

$$T = \begin{bmatrix} 2\\2'\\4\\4' \end{bmatrix} \xrightarrow{(2'3)} \begin{bmatrix} 2\\3\\4\\4' \end{bmatrix} \xrightarrow{(2'2')} \begin{bmatrix} 2'\\3\\4\\4' \end{bmatrix} \xrightarrow{(3'3')} \begin{bmatrix} 2'\\3'\\4\\4' \end{bmatrix} \xrightarrow{(2'3)} \begin{bmatrix} 3\\3'\\4\\4' \end{bmatrix} \xrightarrow{\text{rectification}} \begin{bmatrix} 3\\3'\\4\\4' \end{bmatrix} = BK_2^{\mathbb{C}}(T).$$

The corresponding Kashiwara tableaux are obtained, morally, by pushing each instance of 1', 2', 3' and 4' down. In this case,

$$T = \frac{2}{\frac{2'}{4}} \mapsto \frac{2}{\frac{4}{4'}}, \quad \text{and} \quad BK_2^{\mathbb{C}}(T) = \frac{3}{\frac{3'}{4}} \mapsto \frac{3}{\frac{4}{4'}}.$$

One may check that each of these two tableaux is fixed by s_2 .

As announced, the diagram also doesn't commute for row tableaux. For instance, we have



but again, both Kashiwara's tableaux are fixed by s_1 .

Appendix C

Axioms for De Concini's and Kashiwara's tableaux

It is our goal in this Appendix to present proofs for the equivalence statements in Chapter 8.

We recall here our main definitions. A **De Concini (symplectic) tableau** is a semistandard tableau *T* in the alphabet $\mathcal{B} = \{n' < \cdots < 2' < 1' < 1 < 2 < \cdots < n\}$ such that each column of *T* is admissible, and such that the split version of *T* is semistandard.

A Kashiwara (symplectic) tableau is a semistandard tableau *T* in the alphabet $C = \{1 < \cdots < n < n' < \cdots < 1'\}$ such that

- (K1) if *a* and *a*' appear in the same column, say T(r, c) = a, T(s, c) = a', then (s r) + a is strictly greater than the length of column T(-, c), and
- (K2) if two adjacent columns of *T* have one of the following two configurations:

(by which we mean that e.g. the first a entry in the first column is at row number p, etc.), then (q-p) + (s-r) < (b-a). Here, $p \le q < r \le s$ and $a \le b$.

The first equivalence statement we presented is the following.

Lemma C.1. Let T be a tableau in the alphabet C satisfying (K1). Then, T satisfies (K2) if and only if it satisfies (K2');

(K2') if two adjacent columns of T have one of the following two configurations:

$$\begin{array}{c|cccc} p \to & a & & a \\ q \to & & b & b \\ r \to & c' & c' \\ s \to & d' \end{array}$$

d'

then $(q - p) + (r - s) < \max\{b, c\} - \min\{a, d\}$. Here, $p \le q < r \le s$, $a \le b$, and $c \le d$.

Proof. The conditions of (K2) are particular cases of those of (K2'), so we only need to show one implication.

Assume that *T* verifies (K2). Let *T* have the first configuration of (K2') for some a, b, c, d. If *T* were to have the second configuration, a similar proof would follow.

$$\begin{array}{cccc} p \to & a \\ q \to & & b \\ r \to & & c' \\ s \to & & d' \end{array}$$

Let α be the smallest number greater than $\max\{a, d\}$ such that α appears in the first column and α' in the second column. If such a number doesn't exist, then the entries between rows p and q of the first column, together with the entries between rows r and s of the second column, are all distinct. In formulas, there are (p-q) + (s-r) entries, all lying in $[\min\{b, c\}, \max\{b, c\})$, giving the desired output. Assume therefore we have the following:

$$\begin{array}{cccc} p \to & a \\ \pi \to & \alpha \\ q \to & & b \\ r \to & & c' \\ \sigma \to & & a \\ s \to & & d' \end{array}$$

Note that, by construction, the set of entries between a and α is disjoint to the set of entries between a' and a'. This gives

$$(\pi - p) + (\sigma - s) \le \alpha - \min\{a, d\}. \tag{C.2}$$

Let β be the greatest number smaller than $\min\{b, c\}$ such that β and β appear in the second column, with $\alpha \leq \beta$. If such a number doesn't exist, then the set of entries between rows π and q in the second column is disjoint to the set of entries between rows r and σ . This gives $(q - \pi) + (\sigma - r) \leq \max\{b, c\} - \alpha$, which together with the above equation give the desired output, once again. Assume therefore we have:

$$\begin{array}{cccc} p \rightarrow & a \\ \pi \rightarrow & \alpha \\ \xi \rightarrow & \beta \\ q \rightarrow & b \\ r \rightarrow & c' \\ r \rightarrow & c' \\ \sigma \rightarrow & \alpha' \\ s \rightarrow & d' \end{array}$$

Note that, by construction, the set of entries between β and b is disjoint with the set of entries between c' and β' . This gives

$$(q - \xi) + (\rho - r) \le \max\{b, c\} - \beta.$$
 (C.3)

Applying (K2) to entries α and β , we get

$$(\xi - \pi) + (\sigma - \rho) < \beta - \alpha. \tag{C.4}$$

Summing up Equations (C.2), (C.3), and (C.4), we get the desired formula.

Inspired by this and by the bijection between the two sets of tableaux, we now give two novel characterizations of De Concini tableaux.

Proposition C.5. Let T be a tableau in the alphabet \mathcal{B} such that each column is admissible. Then, T is a De Concini tableau if and only if it satisfies (DC');

(DC') if two adjacent columns of T have one of the following two configurations:

$$\begin{array}{c|ccc} p \to & a' \\ q \to & \\ r \to & c \\ s \to & d \end{array} \middle| \begin{array}{c} b' & & c \\ & c \\ & d \end{array} \middle| \begin{array}{c} a \\ b \\ d \end{array} \right|$$

then $(q - p) + (r - s) < \max\{a, d\} - \min\{b, c\}$. Here, $p \le q < r \le s$, $a \le b$, and $c \le d$.

Proof. We show that T verifies (DC') if and only if split(T) is semistandard. The proof is similar to the second part of Theorem C.8.

If $\operatorname{split}(T)$ is semistandard, and *T* has the first configuration of axiom (DC'), then $\operatorname{split}(T)$ has the following configuration:

$$\begin{array}{ccccc} p \rightarrow & a' & * \\ q \rightarrow & & & \\ r \rightarrow & * & c \\ s \rightarrow & * & d \end{array} \qquad b' \quad * \\ \end{array}$$

In the second column, between rows p and q, we find q - p + 1 numbers in [b, a]. Between rows r and s, we find s - r - 1 numbers in (c, d). In total, there are (q - p) + (s - r) numbers in the interval $[\min\{b, c\}, \max\{a, d\}]$. Note that if both bounds are obtained, then at least one of c and d is not being counted by (q - p) + (s - r). So (DC') holds.

If T has the second configuration, a similar argument gives the result.

Suppose *T* verifies (DC') but split(*T*) is not semistandard; more precisely, assume it has the following configuration for some primed entries x', y', and b', with $y \ge b$ and x < b:

$$y' x' \mid b' *$$

(If there was an analogous configuration with all non-primed entries, a similar proof follows.)

Let [K + 1, L] be the interval corresponding to the block in which y' and x' lie; let *a* and *d* be the greatest numbers less than *K* such that a' appears in column 1 and a in column 2; let *c* be the smallest number greater than *x* such that c appears in column 2.

Figure C.6: On the left, a configuration of tableau *T*. On the right, the relative positions of some relevant entries in split(T). Dashed, the levels at which numbers *K* and *x* could appear.

We obtain equations analogous to (C.10) and (C.11) just as in the proof of Theorem C.8, giving a contradiction.

Lemma C.7. Let T be a tableau in the alphabet \mathcal{B} such that each column is admissible. Then, T satisfies (DC') if and only if it satisfies (DC);

(DC) if two adjacent columns of T have one of the following two configurations:

$$\begin{array}{c|ccc} p \to & a' \\ q \to & & \\ r \to & b \\ s \to & a \end{array} \begin{array}{c|ccc} b' & & b' \\ & b' \\ & b' \\ & a \end{array}$$

then (q - p) + (r - s) < a - b. *Here,* $p \le q < r \le s$.

Proof. It is similar to the proof of Lemma C.1 and thus omitted.

The following proof is essentially extracted from [She99, Thm. A.4], but adapted to our notation.

Theorem C.8. Let T be a semistandard tableau in the alphabet $C = \{1 < \dots < n < n' < \dots < 1'\}$. A column of T verifies (K1) if and only if it is admissible. The tableau T verifies (K2) if and only if the cosplit version of T is semistandard.

Proof. Let *T* be an admissible semistandard column tableau in the alphabet *C*. Say *T* is of length *l*. If *a* and *a'* appear in im *T*, in rows *r* and *s* respectively, then the number of entries in $\{1, 2, ..., a\} \cup \{a', ..., 2', 1'\}$ which appear in im *T* must not exceed *a*, by definition of admissibility. In formulas, $r + (l + 1 - s) \neq a$. That is, (s - r) + a > l. The reciprocal is clear.

Let us now discuss (K2). Assume without loss of generality T is a tableau with two only two columns, whose cosplit version is semistandard. Assume for now that we have the first of the four configurations of the definition. Then, $\operatorname{cosplit}(T)$ will have the following configuration:

$$\begin{array}{ccccccc} p \rightarrow & * & a \\ q \rightarrow & & & * & b \\ r \rightarrow & & & b' & * \\ s \rightarrow & & & a' & * \end{array}$$

In the third column, between rows p and q, we find some entries verifying the following inequalities:

$$a \leq c_1 < c_2 < \cdots < c_{q-p+1} \leq b$$

The equalities are not attained, since e.g. they both c_{q-p+1} and b' belong to the same column of a cosplit tableau. Similarly, between rows r and s, we find

$$b' < d'_1 < d'_2 < \cdots < d'_{s-r-1} < a'$$
, or, $a < d_{s-r-1} < \cdots < d_2 < d_1 < b$.

So we have found at least (q - p) + (s - r) numbers in (a, b). They are all distinct by definition of the cosplit version. Therefore, (q - p) + (s - r) < (b - a), as desired.

If T was to have the second of the four configurations, a similar argument would yield the same formula. We omit this.

It remains to show that if (K2) holds then the cosplit version of the tableau is semistandard. The cosplit version of each column is strictly increasing, by definition. We need to check if the rows are weakly increasing.

We proceed by contradiction. Assume cosplit(T) is not semistandard. More precisely, assume that, for some non-primed entries a, x, and y, we find the following configuration in cosplit(T), with a > x and $a \le y$:

 $* a \mid x y$

(If there was an analogous configuration with all primed entries, a similar proof follows. The only remaining possibilities are easily discarded by definition of the cosplit version or because T is semistandard.) We assume T to have only two columns.

Recall the analysis before Example 8.5. We know, since x and y are in the same row, that they belong to a block corresponding to some interval [L, K - 1]. In particular, we have found a number $K \in [2, n + 1]$ such that $x \le y < K$ and whose properties we will shortly exploit.

Let $b \ge y$ be the greatest number smaller than K such that b appears in the fourth column of $\operatorname{cosplit}(T)$. Let c be the greatest number smaller than K such that c' appears in the third column of $\operatorname{cosplit}(T)$. Finally, let $d \le c$ be the smallest number greater than x such that d' appears in the third column of $\operatorname{cosplit}(T)$.

We illustrate all of these choices in Figure C.9. Note that all of the entries we consider in the third and fourth column belong to the same block (corresponding to [L, K - 1]).

Figure C.9: On the left, a configuration of tableau *T*. On the right, the relative positions of some relevant entries in cosplit(T). Dashed, the levels at which numbers *K* and *x* could appear.

We claim that the corresponding entries a, b, c', d' in T, in the given configuration, do not satisfy axiom (K2'):

$$(q-p) + (s-r) \not< \max\{b,c\} - \min\{a,d\}.$$

We begin by considering the left hand side. By definition of *K*, *b*, and *c*, we get that $\max\{b, c\} = K - 1$. On the other hand, a > x and d > x, and thus we get

$$\max\{b, c\} - \min\{a, d\} < K - 1 - x. \tag{C.10}$$

We will now consider the right hand side. In the third column of $\operatorname{cosplit}(T)$, between entries $\overline{d'}$ and $\overline{c'}$ (both inclusive), and between the entry to the left of \overline{b} and \overline{x} (both inclusive), we find (q - p + 1) + (s - r + 1) entries, all distinct. Furthermore, all numbers greater than x and strictly smaller than K appear at least once, by definition of K. Therefore,

$$(q-p+1) + (s-r+1) = K - x.$$
(C.11)

Equations (C.10) and (C.11) imply $(q - p) + (s - r) \ge \max\{b, c\} - \min\{a, d\}$, contradicting (K2'), as desired.
Appendix D

A SageMath library for symplectic tableaux

(This is the documentation for a SageMath library implementing symplectic tableaux and symplectic Gelfand–Tsetlin patterns. The script is available in a GitHub repository [GitHub].)

Symplectic tableaux are type C analogues of semistandard Young tableaux. Explicitly, the generating function of symplectic tableaux of a given shape λ (a partition) is the character of the irreducible representation of $\mathfrak{sp}(2n)$ indexed by λ .

In the literature, several definitions of symplectic tableaux are available. We implement four of these; King's tableaux [Kin76], De Concini's tableaux [DeC79], split tableaux [Kra98], and Kashiwara's tableaux [KN94]. Note that some functions on Kashiwara's tableaux are already implemented in Sage as part of the "tensor product of crystals" library.

class SymplecticTableau(SageObject):

Four different combinatorial models for symplectic tableaux are implemented. Internally, these tableaux are always in the alphabet $1 < 2 < \cdots < 2n$. Later, they may be displayed with a different alphabet.

The default model is King's. They are displayed in the alphabet $1 < 1' < 2 < 2' < \cdots < n < n'$. They are well-defined if every column is admissible, that is, if for every *i* there are at most *i* numbers smaller or equal to *i'* in each column.

A second model is De Concini's. These are displayed in the alphabet $n' < \cdots < 2' < 1' < 1 < 2 < \cdots < n$. These are well-defined if every column is admissible (after reordering to the King's alphabet) and their split version is semistandard. See [She99].

A third model is that of split tableaux. These are displayed in the alphabet $1 < 2 < \cdots < 2n$. These are well-defined when they are the split version of a De Concini tableau.

A final model is Kashiwara's. These are displayed in the alphabet $1 < 2 < \cdots < n < n' < \cdots < 2' < 1'$. These are well-defined when they are admissible and their cosplit version is a split tableau.

Additionally, a custom type of tableaux can be defined by specifying the alphabet in which they should be displayed. Functionality for custom tableaux is limited.

Skew tableaux are implemented by introducing 0s.

Examples

```
sage: tab = SymplecticTableau(3, rows = [[1,1,2,3,3],[4,4,5,6]]); tab
1 1 1 2' 2'
2 2 3 3'
sage: tab.is_well_defined()
True
sage: tab.Kashiwara()
1 2 2 2' 2'
3 3 3' 1'
sage: tab.f(1)
1 1' 1' 2' 2'
2 2 3 3'
sage: print(tab)
Symplectic tableau of type King, shape [5, 4], and weight x1^{(-1)}x3^{(1)}
sage: tab.shape()
[5, 4]
sage: latex(tab)
\ytableaushort{{1}{1}{1}{2'}{2'},{2}{3}{3'}}
```

Input

- n : Positive integer
- rows : SemistandardTableau or list of lists of integers. (Default: None)
- cols : SemistandardTableau or list of lists of integers. (Default: None)
- type : Either 'King', 'DeConcini', 'split' or 'Kashiwara'. (Default: 'King')
- alphabet : a list of strings of fixed length. (Default: None)

Exactly one of rows and cols must be provided.

.rows(self):

Returns the contents of the tableau as a list of lists of entries, row by row. (The entries are in $1 < 2 < \cdots < 2n$.)

.list(self):

See .rows().

.cols(self):

Returns the contents of the tableau as a list of lists of entries, column by column. (The entries are in $1 < 2 < \cdots < 2n$.)

.transpose(self):

See .cols().

.len(self):

Returns the number of rows of the tableau.

.dict(self):

Returns the contents of the tableau as dictionary where the keys are the cells of the tableau and the entries are in $1 < 2 < \cdots < 2n$.

.is_well_defined(self):

Checks if the tableau self is a tableau of type self._type.

.n(self):

Returns *n*; half the number of variables in the alphabet.

.size(self):

See .n().

.weight(self):

Returns a string describing a monomial; the weight of the tableau. (The weight is only defined for King, De Cocini, or Kashiwara tableaux.)

.shape(self):

Returns a partition λ corresponding to the shape of the tableau.

.bender_knuth_involution(self, i, display=False):

Performs the *i*th Bender–Knuth involution on the tableau.

The *i*th Bender–Knuth involution applied to a tableau with weight x^{λ} returns a tableau of same shape and with weight $s_i.x^{\lambda}$, where $s_0, ..., s_{n-1}$ are the generators of the Weyl group of type C. These are analogs of type A Bender–Knuth involutions.

To compute the 0th Bender–Knuth involution on a King tableau, we interpret it as a semistandard tableau in the alphabet $1 < \cdots < 2n$ and then perform the 1st (type A) Bender–Knuth involution.

For i = 1, ..., n-1, the *i*th Bender–Knuth involution on a King tableau is defined as the composite of five maps. The first four maps are usual type A Bender–Knuth involutions. More precisely, if the tableau is interpreted as a (type A) semistandard tableau in the alphabet $1 < \cdots < 2n$, then the first four maps are the 2*i*th, the (2i - 1)st, the (2i + 1)st, and the 2*i*th Bender–Knuth involutions. The fifth map changes every $\{i, i'\}$ -vertical domino between rows *i* and *i* + 1 for a $\{i + 1, i + 1'\}$ -vertical domino. It then resorts rows *i* and *i* + 1 as to make them increasing.

The function converts the tableau to a King tableau, performs the involution and converts back to the original type. If display is set to True, then the intermediate stages of the algorithm are displayed.

Examples
<pre>sage: T = SymplecticTableau(3, rows = [[1,3,4],[5,5,6],[6,6]]); T 1 2 2' 3 3 3' 3' 3' sage: T.bender_knuth_involution(0) 1' 2 2' 3 3 3' 3' 3' sage: T.bender_knuth_involution(2, display = True) Now applying BenderKnuth to the tableau</pre>
T0: 1 2 2' 3 3 3' 3' 3'
T1 : 1 2 3 2' 2' 3' 3' 3'
T2: 1 2 3 2 2' 3' 3' 3'
T3 : 1 2 3 2 2' 3' 3 3
T4: 1 2 2' 2 2' 3' 2' 3
T5 : 1 2 2' 2' 3 3' 3 3'
1 2 2' 2' 3 3' 3 3'

.to_GTpattern(self):

Returns the (King) symplectic Gelfand–Tsetlin pattern associated with the tableau.

Examples

```
sage: T = SymplecticTableau(3, rows = [[1,3,4],[5,5,6],[6,6]]); T
1     2     2'
3     3     3'
3'     3'
```

```
sage: T.to_GTpattern()
3 3 2
3 2 0
3 0
2 0
1
1
1
```

.King(self):

Returns the King tableau corresponding to the tableau.

- If self is a King tableau, this returns self.
- If self is a DeConcini tableau, it applies Sheats' bijection. See . Sheats().
- If self is a split tableau, we first convert to a DeConcini tableau.
- If self is a Kashiwara tableau, we first convert to a split tableau.

.DeConcini(self):

Returns the De Concini tableau corresponding to the tableau.

- If self is a King tableau, it applies Sheats' bijection. See . Sheats().
- If self is a DeConcini tableau, this returns self.
- If self is a split tableau, it takes the inverse of the split map. See .splitInverse().
- If self is a Kashiwara tableau, it first converts to a split tableau.

.split(self):

Returns the split tableau corresponding to the tableau.

- If self is a King tableau, it first converts to a De Concini tableau.
- If self is a DeConcini tableau, it returns the split version. See .splitVersion().
- If self is a split tableau, this returns self.
- If self is a Kashiwara tableau, it returns the cosplit version. See .cosplitVersion().

.Kashiwara(self):

Returns the Kashiwara tableau corresponding to the tableau.

If self is a King tableau, it first converts to a De Concini tableau.

- If self is a DeConcini tableau, it first converts to a split tableau.
- If self is a split tableau, it takes the inverse of the cosplit map. See .cosplitInverse().
- If self is a Kashiwara tableau, this returns self.

.to_type(self, type):

Converts to the desired type. The bijections implemented are

King \leftrightarrow De Concini \leftrightarrow split \leftrightarrow Kashiwara.

Any other bijection is taken to be a composite of some of the above.

Input

• type: Either 'King', 'DeConcini', 'split', or 'Kashiwara'.

Examples
<pre>sage: T = SymplecticTableau(3, rows = [[1,3],[5,5],[6,6]]); T 1</pre>
<pre>sage: T.to_type('DeConcini') 3' 2' 2' 1' 2 3</pre>
<pre>sage: T.to_type('split') 1 1 2 2 2 3 3 3 4 5 6 6</pre>
<pre>sage: T.to_type('Kashiwara') 1 2 3 3 3' 1'</pre>

.f(self, i):

Applies the crystal operator f_i .

Crystal operators are already implemented in Sage for Kashiwara's tableaux. See the documentation for crystals.tensor_product. This functions converts the tableau to a Kashiwara tableau, performs the crystal operator, and converts back to the original type.



Applies the crystal operator e_i .

Crystal operators are already implemented in Sage for Kashiwara's tableaux. See the documentation for crystals.tensor_product. This functions converts the tableau to a Kashiwara tableau, performs the crystal operator, and converts back to the original type.

Examples



```
sage: T.e(1)
1 2
3 3
3' 3'
```

77

```
class SymplecticTableauIterator
```

```
def transpose(tab)
```

Computes the transpose of a tableau.

def splitVersion(tab)

Returns the split version of a tableau.

Takes a tableau in the $1 < 2 < \cdots < 2n$, relabels it to $n' < \cdots < 2' < 1' < 1 < 2 < \cdots < n$, and returns its split version.

The split version of a tableau of shape $\lambda = (\lambda_1, \lambda_2, ...)$ is defined as a certain tableau of shape $2\lambda = (2\lambda_1, 2\lambda_2, ...)$. The precise definition is available in [She99].

```
Examples
sage: T = SymplecticTableau(4, cols = [[1,2,6,7]], type = 'DeConcini'); T
4'
3'
2
3
sage: is_admissible(T.cols()[0], 4)
True
sage: SymplecticTableau(4, rows = splitVersion(T).list(), alphabet =
....: T._alphabet)
4' 4'
3' 1'
1 2
2 3
```

def cosplitVersion(tab):

Returns the cosplit version of a tableau.

Takes a tableau in the $1 < 2 < \cdots < 2n$, relabels it to $n' < \cdots < 2' < 1' < 1 < 2 < \cdots < n$, and returns its cosplit version. The cosplit version of a tableau of shape $\lambda = (\lambda_1, \lambda_2, ...)$ is defined as a certain tableau of shape $2\lambda = (2\lambda_1, 2\lambda_2, ...)$. The precise definition is available in [She99].

```
Examples
sage: a = ["4'", "3'", "2'", "1'", "1 ", "2 ", "3 ", "4 "]
sage: T = SymplecticTableau(4, cols = [[1,4,5,6]], alphabet = a); T
4'
1'
1
2
sage: is_coadmissible(T.cols()[0], 4)
True
sage: SymplecticTableau(4, rows = cosplitVersion(T).list(), alphabet = a)
4' 4'
3' 1'
1 2
2 3
```

def splitInverse(tab):

Takes the split version of a tableau (in the alphabet $1 < 2 < \cdots < 2n$) and returns the original tableau.

```
def cosplitInverse(tab):
```

Takes the cosplit version of a tableau (in the alphabet $1 < 2 < \cdots < 2n$) and returns the original tableau.

```
def is_admissible(c, n):
```

Takes a semistandard column in $1 < 2 < \cdots < 2n$, relabels it to $n' < \cdots < 2' < 1' < 1 < 2 < \cdots < n$ and checks admissibility.

A column of length *l* is admissible if for all *a* such that both *a'* and *a* appear in the column, say in rows *s* and *r*, one has (s - r) + a > l.

```
def is_coadmissible(c, n):
```

Takes a semistandard column in $1 < 2 < \cdots < 2n$, relabels it to $n' < \cdots < 2' < 1' < 1 < 2 < \cdots < n$ and checks coadmissibility.

A column is coadmissible if for all *a* such that both *a*' and *a* appear in the column, say in rows *s* and *r*, one has (s - r) + a > n + 1.

The use of the next functions is hopefully clear from their names. These are used in the implementation of methods such as SymplecticTableau.King().

```
def King_to_DeConcini(tab)
def DeConcini_to_King(tab)
def King_to_Kashiwara(tab)
def Kashiwara_to_King(tab)
def DeConcini_to_Kashiwara(tab)
def Kashiwara_to_DeConcini(tab)
def King_to_split(tab)
def split_to_King(tab)
def peConcini_to_split(tab)
def split_to_DeConcini(tab)
def Kashiwara_to_split(tab)
def Kashiwara_to_split(tab)
def Kashiwara_tableau_to_CrystalElement(tab)
def CrystalElement_to_KashiwaraTableau(tab, n)
```

We choose to include the implementation of the following functions in this documentation.

```
def Sheats(tab):
```

Performs Sheats' algorithm on tab.

Given a tableau in the alphabet $1 < 2 < \cdots < 2n$, it is interpreted as a De Concini tableau in the alphabet $n' < \cdots < 2' < 1' < 1 < 2 < \cdots < n$. Then, the Sheats algorithm [She99] is performed on the tableau, resulting on a King tableau in the alphabet $1 < 2 < \cdots < 2n$ (interpreted as in the alphabet $1' < 1 < \cdots < n' < n$).

```
Examples
sage: tab = SymplecticTableau(3, rows = [[1, 2], [2, 3], [5, 6]],
                              type = 'DeConcini'); tab
3' 2'
2' 1'
2 3
sage: Sheats(tab)
1 2
3
  3
3' 3'
sage: tab = SymplecticTableau(3, rows = [[1,3,3], [2,5], [4,6]],
                             type = 'DeConcini'); tab
. . . . :
3' 1' 1'
2' 2
1 3
sage: Sheats(tab)
1 2 2
2' 2'
3 3'
```

```
# we will be perfoming transformations on a subtableau and then
# adding some stuff to each column.
lam1 = tab._shape[0]
add = [[] for i in range(lam1)]
n = tab._n
cols = tab._cols
# n gives the size of the alphabet
for m in [1...n]:
   # m is the entry to apply jdtq on
    # m is taken from the alphabet 1<2<\cdots<2n</pre>
   mPrime = m
   mNormal = 2*n+1-m
   # in the new order, the entries that we moved have another relative position
   newMPrime = 2*(n-m)+1
   newMNormal = 2*(mNormal-n)
   # k is the number of times m' appears
   k = len([col for col in cols if len(col)!=0 and col[0] == mPrime])
    # the entries equal to m will stay fixed.
    # but in the new tableau, they are in a different
    # position of the alphabet, so translation is needed.
    colsFix = [[newMNormal for i in col if i == mNormal] for col in cols]
   colsFix = colsFix + [[] for i in range(lam1 - len(colsFix))]
    # everything else that is not m or m' will be movable
    colsMove = [[i for i in col if i != mNormal and i != mPrime] for col in cols]
   # we will now transform colsMove and later add some extra entries
    for i in range(lam1):
       add[i] = colsFix[i] + add[i]
    # this transformation expects a split tableau
   Cols = sum([_splitCol(col, n) for col in colsMove], [])
    while k != 0:
       # we apply _sjdt to the only inner corner
        (Cols, (p,q), n) = _sjdt(Cols, (1,k), n)
```

we keep track of the entry we need to add

```
# later to the tableau
        cols = _splitInverseCols(Cols, n)
        cols = cols + [[] for i in range(lam1-len(cols))]
        add[q-1] = [newMPrime] + add[q-1]
        for i in range(lam1):
           add[i] = [newMNormal]*cols[i].count(mNormal) + add[i]
        # we prepare for the next iteration
       k = k + len([col for col in cols if mPrime in col]) - 1
        colsMove = [[i for i in col if i != mNormal and i != mPrime] for col in cols]
        Cols = sum([_splitCol(col, n) for col in colsMove], [])
    colsMoved = _splitInverseCols(Cols, n)
    colsMoved = colsMoved + [[] for i in range(lam1- len(colsMoved))]
    # colsMoved = [[i-1 for i in col] for col in colsMoved]
   # we prepare the body for the next iteration
   cols = colsMoved
# now comes the time to add the entries that we were keeping aside
newCols = []
for i in range(lam1):
   newCols.append(colsMoved[i] + add[i])
```

```
return SymplecticTableau(tab._n, cols = newCols, type = 'King')
```

```
def _sjdt(cols, puncture, n):
```

Takes a set of columns that form a punctured split skew tableau if the puncture is set at the input location. We let 0s mark the empty skew boxes.

The puncture coordinates are in terms of the non-split version of the tableau, starting at (1,1).

```
(p,q) = puncture # starts at (1,1)
# Step 1:
# if the puncture is an outer corner, stop.
if len(cols) <= 2*q or len(cols[2*q]) < p:</pre>
    if len(cols[2*q-1]) < p:
       return (cols, (p,q), n)
# Step 2:
# if the column to the right is too short,
# just move the puncture down.
    return _sjdt(cols, (p+1, q), n)
# Step 3:
# if the current column is too short,
# move the puncture to the right.
if len(cols[2*q-1]) < p:</pre>
    return _sjdt_subroutine(cols, puncture, n)
# Step 4:
# when the algorithm reaches this step, it
# has checked that there are two possible
# locations to move the puncture to.
# We now compare entries to see where to go.
if cols[2*q-1][p-1] <= cols[2*q][p-1]:</pre>
   return _sjdt(cols, (p+1, q), n)
else:
    return _sjdt_subroutine(cols, puncture, n)
```

```
def _sjdt_subroutine(cols, puncture, n):
```

Takes a punctured split tableau (as an array of columns and a tuple indicating the puncture) and returns the punctured split tableau resulting from a right slide.

def SheatsInverse(tab)

Performs the inverse of the Sheats algorithm on tab.

Given a tableau in the alphabet $1 < 2 < \cdots < 2n$, it is interpreted as a King tableau in the alphabet $1' < 1 < 2' < 2 \dots < n' < n$. Then, the Sheats inverse algorithm is performed on the tableau, resulting on a De Concini tableau in the alphabet $1 < 2 < \cdots < 2n$ (interpreted as in the alphabet $n' < \cdots < 1' < 1 < \cdots < n$).

Examples

```
sage: tab = SymplecticTableau(3, rows = [[1, 3], [5, 5], [6, 6]]); tab
  2
3 3
3' 3'
sage: SheatsInverse(tab)
3' 2'
2' 1'
2 3
sage: tab = SymplecticTableau(3, rows = [[1, 3, 3], [4, 4], [5, 6]]); tab
1 2 2
2' 2'
3 3'
sage: SheatsInverse(tab)
3' 1' 1'
2' 2
1 3
```

def check_all_BKinvolutions(lam, i, max_entry = None):

Checks that the *i*th Bender-Knuth involution is an involution on all tableaux of shape lam and maximal entry max_entry.

class SymplecticPattern(SageObject):

A class for (King's) Gelfand-Tsetlin patterns.

Warning

This class is very rudimentary as of now.

```
.list(self):
.len(self):
.to_tableau(self):
```

```
def KingTableau_to_SymplecticPattern(T)
def SymplecticPattern_to_KingTableau(G)
```

```
Examples
```

Bibliography

- [Catalog] Per Alexandersson. Miscellaneous. The symmetric functions catalog. https://www. symmetricfunctions.com/schurMisc.htm. Accessed 06/08/2022.
- [ALRS13] Federico Ardila, Emerson León, Mercedes Rosas, and Mark Skandera. Three lectures in algebraic combinatorics II. Symmetric functions and representation theory. 2013. Original title: Tres lecciones en combinatoria algebraica. II. Las funciones simétricas y la teoría de representaciones. https://arxiv.org/1301.3988 English translation: to appear.
- [BK72] Edward Bender and Donald Knuth. Enumeration of plane partitions. *Journal of Combinatorial Theory, Series A*, 13(1):40–54, 1972. https://doi.org/10.1016/0097-3165(72)90007-6.
- [BK95] Arkady Berenstein and Anatolii Kirillov. Groups generated by involutions, Gelfand–Tsetlin patterns, and combinatorics of Young tableaux. *Algebra i Analiz*, 7:92–152, 1995. http://mi. mathnet.ru/aa491.
- [Bou02] Nicolas Bourbaki. *Lie groups and Lie algebras. Chapters 4–6.* Springer–Verlag, Berlin, 2002. Translated from the french 1968 book.
- [BS17] Daniel Bump and Anne Schilling. Crystal Bases. World Scientific, 2017.
- [BZ22] Elia Bisi and Nikos Zygouras. Transition between characters of classical groups, decomposition of Gelfand–Tsetlin patterns and last passage percolation. Advances in Mathematics, 404:108453, 2022. https://doi.org/10.1016/j.aim.2022.108453.
- [DeC79] Corrado De Concini. Symplectic standard tableaux. Advances in Mathematics, 34(1):1–27, 1979. https://doi.org/10.1016/0001-8708(79)90061-6.
- [FH91] William Fulton and Joe Harris. Representation Theory: A First Course. Graduate Texts in Mathematics. Springer New York, 1991.
- [FK97] Markus Fulmek and Christian Krattenthaler. Lattice path proofs for determinantal formulas for symplectic and orthogonal characters. *Journal of Combinatorial Theory, Series A*, 77(1):3– 50, 1997. https://doi.org/10.1006/jcta.1996.2711.
- [GR16] Darij Grinberg and Tom Roby. Iterative properties of birational rowmotion I: Generalities and skeletal posets. *The Electronic Journal of Combinatorics*, 23, 2016. https://doi.org/10.37236/ 4334.
- [GT50] Israel M. Gelfand and Michael. L. Tsetlin. Finite dimensional representations of the group of unimodular matrices. *Doklady Akad. Nauk SSSR (N.S.)*, 71:825–828, 1950.
- [GitHub] Álvaro Gutiérrez. Symplectic tableaux and patterns. https://github.com/alvgutcac/ SymplecticTableaux.

- [GV85] Ira Gessel and Gérard Viennot. Binomial determinants, paths, and hook length formulae. *Adv. in Math.*, 58(3):300–321, 1985. https://doi.org/10.1016/0001-8708(85)90121-5.
- [HK02] Jin Hong and Seok-Jin Kang. *Introduction to quantum groups and crystal bases*, volume 42. American Mathematical Soc., 2002.
- [Hum72] James Humphreys. Introduction to Lie Algebras and Representation Theory. Graduate Texts in Mathematics. Springer New York, NY, 1972. https://doi.org/10.1007/978-1-4612-6398-2.
- [Jac41] Carl G.J. Jacobi. De functionibus alternantibus earumque divisione per productum e differentiis elementorum conflatum. *Journal für die reine und angewandte Mathematik*, 22:360–371, 1841. http://eudml.org/doc/147139.
- [Kas94] Masaki Kashiwara. Crystal bases of modified quantized enveloping algebra. *Duke Mathematical Journal*, (2), 1994.
- [Kin76] Ronald King. Weight multiplicities for the classical groups. In A. Janner, T. Janssen, and M. Boon, editors, *Group Theoretical Methods in Physics*, pages 490–499, Berlin, Heidelberg, 1976. Springer Berlin Heidelberg.
- [KN94] Masaki Kashiwara and Toshiki Nakashima. Crystal Graphs for Representations of the q-Analogue of Classical Lie Algebras. Journal of Algebra, 165(2):295–345, 1994. https://doi.org/ 10.1006/jabr.1994.1114.
- [Kra98] Christian Krattenthaler. Identities for classical group characters of nearly rectangular shape. *Journal of Algebra*, 209(1):1–64, 1998. https://doi.org/10.1006/jabr.1998.7531.
- [KT87] Kazuhiko Koike and Itaru Terada. Young-diagrammatic methods for the representation theory of the classical groups of type B_n , C_n , D_n . J. Algebra, 107(2):466–511, 1987. https: //doi.org/10.1016/0021-8693(87)90099-8.
- [Lin73] Bernt Lindström. On the vector representations of induced matroids. *Bulletin of The London Mathematical Society*, 5:85–90, 1973.
- [Lit90] Peter Littelmann. A Generalization of the Littlewood–Richardson Rule. *Journal of Algebra*, 130:328–368, 1990. https://doi.org/10.1016/0021-8693(90)90086-4.
- [LMS79] Venkatraman Lakshmibai, Chitikila Musili, and Conjeevaram S. Seshadri. Geometry of G/P - IV. (Standard monomial theory theory for classical types). Proc. Indian Acad. Sci., 88A:279– 362, 1979.
- [Oka89] Soichi Okada. Lattice path methods for classical groups. Peprint, 1989.
- [Pro94] Robert Proctor. Young tableaux, Gelfand patterns, and branching rules for classical groups. *Journal of Algebra*, 164(2):299–360, 1994. https://doi.org/10.1006/jabr.1994.1064.
- [Roé13] Joaquim Roé. Involutions on a tropical line. Mathematica Scandinavica, 112(1):78–85, Mar. 2013. https://doi.org/10.7146/math.scand.a-15234.
- [Sage] The Sage Developers. Classical crystals (thematic tutorial for SageMath). https://doc. sagemath.org/html/en/thematic tutorials/lie/crystals.html. Accessed 19/02/2023.
- [Sag01] Bruce Sagan. The symmetric group: representations, combinatorial algorithms, and symmetric functions. Springer, 2001.

- [Sam17] Steven Sam. Notes for math 740 (symmetric functions), 2017. https://mathweb.ucsd.edu/ ~ssam/old/17S-740/notes.pdf.
- [San21] João Miguel Santos. Symplectic keys and Demazure atoms in type C. *The Electronic Journal of Combinatorics*, 28, 2021. https://doi.org/10.37236/9235.
- [Sch08] Martin Schottenloher. A Mathematical Introduction to Conformal Field Theory. Lecture Notes in Physics. Springer Berlin, Heidelberg, 2008. https://doi.org/10.1007/978-3-540-68628-6.
- [She99] Jeffrey Sheats. A symplectic jeu de taquin bijection between the tableaux of King and of De Concini. *Transactions of the American Mathematical Society*, 351(9):3569–3607, 1999. http://www.jstor.org/stable/117933.
- [Sta99] Richard Stanley. *Enumerative combinatorics Vol. 2.* Cambridge Studies in Advanced Mathematics, 1999.
- [Str22] Catharina Stroppel. Representation theory of Lie algebras, 2022. Class notes. Transcribed by Qi Zhu.
- [Sun86] Sheila Sundaram. On the combinatorics of representations of Sp(2n, C). PhD thesis, Massachusetts Institute of Technology, Dept. of Mathematics, 1986. https://dspace.mit.edu/ handle/1721.1/15060.
- [Sun90] Sheila Sundaram. Orthogonal tableaux and an insertion algorithm for so(2n + 1). *Journal of Combinatorial Theory, Series A*, 53(2):239–256, 1990. https://www.sciencedirect.com/science/article/pii/0097316590900596.
- [SV16] Anna Stokke and Terry Visentin. Lattice path constructions for orthosymplectic determinantal formulas. *European Journal of Combinatorics*, 58:38–51, 2016. https://doi.org/10.1016/ j.ejc.2016.05.002.
- [Tru64] Nicola Trudi. Intorno ad un determinante più generale di quello che suol dirsi determinante delle radici di una equazione ed alle funzioni simmetriche complete di queste radici. Stamperia del Fibreno, 1864.