

UC Santa Barbara

UC Santa Barbara Electronic Theses and Dissertations

Title

Spiders and Cobwebs Galore

Permalink

<https://escholarship.org/uc/item/9dw9r52b>

ISBN

9798186381693

Author

Mantell, Sarah

Publication Date

2026-06-05

Peer reviewed|Thesis/dissertation

University of California
Santa Barbara

Spiders and Cobwebs Galore

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Mathematics

by

Sarah P. Mantell

Committee in charge:

Professor Stephen Bigelow, Chair
Professor Xiaolei Zhao
Professor Zhenghan Wang

June 2026

The Dissertation of Sarah P. Mantell is approved.

Professor Xiaolei Zhao

Professor Zhenghan Wang

Professor Stephen Bigelow, Committee Chair

June 2026

Spiders and Cobwebs Galore

Copyright © 2026

by

Sarah P. Mantell

Dedicated to my grandfather, Ashley Lipshutz, for buying
me my first calculus book.

Acknowledgements

To my family: thank you for instilling a love for learning in me since the day I was born. Thank you for supporting me and for cheering me on every step of the way.

To my cats (past, present, and future): thank you for keeping me company during countless late nights and early mornings, and for brightening my day.

And to my Joel: thank you for the endless encouragement and love. You're a big part of why I started my PhD and a *huge* part of why I finished.

Curriculum Vitæ

Sarah P. Mantell

Education

2026	Ph.D. in Mathematics (Expected), University of California, Santa Barbara.
2024	M.A. in Mathematics, University of California, Santa Barbara.
2021	B.S. in Mathematics, California Polytechnic State University, San Luis Obispo

Supervising Faculty

Dr. Stephen Bigelow (Committee Chair)
Dr. Xiaolei Zhao
Dr. Zhenghan Wang

Abstract

Spiders and Cobwebs Galore

by

Sarah P. Mantell

The SL_n spider from [8] is a diagrammatic category that describes the representation theory of the quantum group $U_q(\mathfrak{sl}_n)$. Cobwebs, first defined in [1], are another type of diagram that encode the combinatorial data of $\mathfrak{sl}_n$. I define a new set of cobweb relations, including the pop-switch relation from [12], and prove that these relations compute the Ocneanu cells in type A_n . With this modified set of cobweb relations, I demonstrate how cobwebs provide a diagrammatic way to compute the Ocneanu cells.

The main result in [8] implies that any closed web in the SL_n spider can be reduced to a scalar multiple of the empty diagram, by repeatedly applying spider relations. Computational evidence strongly suggests that any closed web simplifies to a symmetric unimodal polynomial over the Laurent ring $\mathbb{Z}[q, q^{-1}]$. I make significant progress towards a proof of this conjecture by defining a collection of relations in the SL_n spider that preserve the symmetric and unimodal properties. Each relation must also maintain or decrease the total number of vertices present in the web. When a web can be fully reduced using this collection of relations, I prove that the result is a symmetric unimodal polynomial. I also define and implement an algorithm that evaluates any closed web diagram.

Contents

Curriculum Vitae	vi
Abstract	vii
1 Introduction	1
1.1 Lie theory	1
1.2 Representation theory in type A	9
2 Quantum groups	15
2.1 A few helpful lemmas	15
2.2 $U_q(\mathfrak{sl}_n)$	17
2.3 Representations of $U_q(\mathfrak{sl}_n)$	19
3 Spiders	22
3.1 A brief history of spiders	22
3.2 The SL_n spider	22
4 Diagrammatic computations of Ocneanu cells	27
4.1 Ocneanu cells	27
4.2 Cobwebs	28
4.3 From Webs to Cobwebs	37
4.4 Computing Ocneanu cells with Cobwebs	40
5 Evaluating Closed Webs	55
5.1 Unimodal symmetric polynomials	56
5.2 Simplifying small webs	58
5.3 Removing small faces	60
5.4 A collection of valid moves	63

5.5	Morsifying W	64
5.6	The Algorithm	66
6	Future directions	77
6.1	Extending the algorithm	77
6.2	Big-data approach	78
6.3	A geometric perspective	78

Chapter 1

Introduction

This chapter contains a brief introduction to the majority of the background necessary for the remaining chapters. The material can be found in any standard text in representation theory and Lie theory (see for instance [11, 13, 14]).

The content of my dissertation falls into two main projects. The first establishes a diagrammatic method to compute the Ocneanu cells for type A_n webs. The second describes an evaluation algorithm for any closed web diagram. Furthermore, this result is shown to be a unimodal symmetric polynomial under certain assumptions. This chapter will define the background necessary for both parts of the dissertation. Results relevant to only one of the projects will be defined later on, as needed.

1.1 Lie theory

A *Lie group* has the structure of a group and a differentiable manifold in such a way that the two structures are compatible. Lie groups are used to model *continuous symmetries*. For example, the special orthogonal group SO_3

is the group of rotations about the origin in three-dimensional Euclidean space. Common applications include particle physics, gauge theory, and statistical mechanics. However, there are symmetries in a quantum mechanical system that cannot be directly explained via Lie groups. [Chapter 2](#) defines a quantum group, constructed as the deformation of a semisimple Lie algebra.

1.1.1 Preliminary definitions

The main results in Lie theory relevant to the remainder of the dissertation are collected here. For a detailed discussion of these topics, see [\[13, 14\]](#). This paper focuses on a specific Lie group:

Definition 1.1: The *special linear group* $\mathrm{SL}_n(\mathbb{C})$, or just SL_n , is the collection of $n \times n$ matrices over $\mathbb{C}$ with determinant 1.

Every Lie group has a related structure, called a *Lie algebra*:

Definition 1.2: A *Lie algebra* is an algebra L whose multiplication map $[\cdot, \cdot] : L \times L \rightarrow L$ is called the *Lie bracket* and satisfies:

- (i) Antisymmetry: $[x, y] = -[y, x]$;
- (ii) The Jacobi identity: $[x, [y, z]] + [y, [z, x]] + [z, [x, y]] = 0$

for all $x, y, z \in L$.

The Lie algebra associated to $\mathrm{SL}_n(\mathbb{C})$ is the *special linear Lie algebra*:

Definition 1.3: The *special linear Lie algebra* $\mathfrak{sl}_n(\mathbb{C})$ is the Lie algebra of $n \times n$ matrices over $\mathbb{C}$ with trace zero. The bracket is given by

$$[X, Y] = XY - YX.$$

Example 1.4: Consider the Lie algebra $\mathfrak{sl}_2(\mathbb{C}) = \mathfrak{sl}_2$. This is the collection of 2×2 matrices over $\mathbb{C}$ with trace zero. The matrices

$$X = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \quad H = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

are a basis for $\mathfrak{sl}_2$. A quick computation shows that

$$[X, Y] = H, \quad [H, X] = 2X, \quad [H, Y] = -2Y.$$

Definition 1.5: Let $\mathfrak{g}$ and $\mathfrak{h}$ be Lie algebras. A linear map $\varphi : \mathfrak{g} \rightarrow \mathfrak{h}$ is a Lie algebra homomorphism where $\varphi([X, Y]) = [\varphi(X), \varphi(Y)]$ for all $X, Y \in \mathfrak{g}$. That is, a Lie algebra homomorphism must preserve the bracket.

Example 1.6: Let $\mathfrak{g}$ be a Lie algebra and let $X \in \mathfrak{g}$. Define a linear map $\text{ad}_X : \mathfrak{g} \rightarrow \mathfrak{g}$ by

$$\text{ad}_X(Y) = [X, Y].$$

Then, $\text{ad} : \mathfrak{g} \rightarrow \text{End } \mathfrak{g}$ where $\text{ad} : X \mapsto \text{ad}_X$ is a Lie algebra homomorphism, called the *adjoint map* or *adjoint representation*. The adjoint representation is discussed further in [Section 1.1.2](#).

1.1.2 Representations of Lie algebras

Because SL_n is simply connected, there is a one-to-one correspondence between the representations of SL_n and the representations of $\mathfrak{sl}_n$. Generally, it

is easier to study the representation theory of a Lie algebra rather than a Lie group and this result says that the two are one in the same. As we will not work with Lie group representations directly, this section focusses only on the representation theory of Lie algebras. Unless explicitly mentioned otherwise, all representations are assumed to be finite-dimensional and over $\mathbb{C}$.

Definition 1.7: A *representation* of a Lie algebra $\mathfrak{g}$ is a (finite-dimensional) vector space V (over $\mathbb{C}$) along with a map

$$\rho : \mathfrak{g} \rightarrow \mathfrak{gl}(V)$$

that preserves brackets. Such a representation is denoted by the pair (V, ρ) , or simply V if ρ is clear from context.

By definition, any Lie algebra $\mathfrak{g}$ is a finite-dimensional vector space over $\mathbb{C}$ and therefore can be viewed as a representation of itself. The Lie algebra homomorphism $\text{ad} : X \mapsto \text{ad}_X$ is therefore a representation of $\mathfrak{g}$ called the *adjoint representation*. The *trivial representation* of $\mathfrak{g}$ is $\pi : \mathfrak{g} \rightarrow \mathbb{C}$ given by $\pi(X) = 0$ for all $X \in \mathfrak{g}$.

Observe that the elements of $\mathfrak{sl}_n$ can be viewed as linear maps on $\mathbb{C}^n$. Then, $V = \mathbb{C}^n$ is a representation of $\mathfrak{sl}_n$ defined via matrix multiplication. This representation is the *standard representation* of $\mathfrak{sl}_n$.

Example 1.8: The *standard representation* of $\mathfrak{sl}_2$ is $V = \mathbb{C}^2$ where $\mathfrak{sl}_2$ acts via

matrix multiplication. Let $\{e_1, e_2\}$ be the standard basis for V . Then,

$$\begin{aligned} H \cdot e_1 &= e_1 & H \cdot e_2 &= -e_2 \\ X \cdot e_1 &= 0 & X \cdot e_2 &= e_1 \\ Y \cdot e_1 &= e_2 & Y \cdot e_2 &= 0 \end{aligned}$$

Observe that the subspaces $\mathbb{C}\{e_1\}$ and $\mathbb{C}\{e_2\}$ are each invariant under the action of H . Furthermore, the action of X on $\mathbb{C}\{e_2\}$ is $\mathbb{C}\{e_1\}$. Likewise, the action of Y sends $\mathbb{C}\{e_1\}$ to $\mathbb{C}\{e_2\}$.

As seen in the previous example, the subalgebra of $\mathfrak{sl}_2$ generated by H acts diagonally on the standard representation of $\mathfrak{sl}_2$. In fact, every $\mathfrak{sl}_n$ contains a subalgebra $\mathfrak{h}$ that acts diagonally on the standard representation $V = \mathbb{C}^n$.

Definition 1.9: For each $i = 1, \dots, n-1$ define

$$H_i = E_i - E_{i+1}$$

where E_i is the $n \times n$ matrix with a 1 in the (i, i) entry and zeros elsewhere. The subalgebra $\mathfrak{h} = \langle H_i \rangle$ of $\mathfrak{sl}_n$ is called the *Cartan subalgebra*.

If $V = \mathbb{C}^n$ is the standard representation with basis $\{e_1, \dots, e_n\}$, then $\mathfrak{h}$ acts diagonally on V . In [Section 1.1.4](#), the Cartan subalgebra plays an important role in characterizing the irreducible representations of $\mathfrak{sl}_n$. To characterize the irreducible representations, we need a generalized version of eigenspaces.

Definition 1.10: Let (π, V) be a representation of $\mathfrak{sl}_n$ and $\lambda : \mathfrak{h} \rightarrow \mathbb{C}$ a linear functional on the Cartan subalgebra of $\mathfrak{sl}_n$. The *weight space* of V with *weight*

λ is

$$V_\lambda = \{v \in V \mid \pi(H)v = \lambda(H)v \text{ for all } H \in \mathfrak{h}\}.$$

A *weight vector* for the weight λ is a nonzero element of V_λ .

Example 1.11: We continue with [Example 1.8](#). The standard representation $V = \mathbb{C}^2$ decomposes into a direct sum of its weight spaces:

$$V = \mathbb{C}\{e_1\} \oplus \mathbb{C}\{e_2\} = V_1 \oplus V_{-1}.$$

The weight spaces are invariant under the action of H .

Because $\mathfrak{sl}_n$ is semisimple, any representation can be written as a direct sum of irreducible representations. Therefore, to understand the representation theory of $\mathfrak{sl}_n$, it suffices to characterize the irreducible representations.

Definition 1.12: Let $(V, \pi_1), (W, \pi_2)$ be representations of $\mathfrak{sl}_n$. Then, $V \otimes W$ is also a representation of $\mathfrak{sl}_n$, with action given by

$$(\pi_1 \otimes \pi_2)(X) = \pi_1(X) \otimes I + I \otimes \pi_2(X).$$

If (V, π) is a representation of $\mathfrak{sl}_n$, then so is (V^*, π^*) with

$$\pi^*(X) = -\pi(X)^T.$$

1.1.3 Root spaces

Let $\mathfrak{h} \subseteq \mathfrak{sl}_n$ be the Cartan subalgebra, containing all diagonal matrices with trace zero.

Definition 1.13: A nonzero element $\alpha \in \mathfrak{h}$ is a *root* if there exists a nonzero $X \in \mathfrak{sl}_n$ such that

$$[H, X] = \langle \alpha, H \rangle X$$

for all $H \in \mathfrak{h}$. If α is a root, then

$$\mathfrak{g}_\alpha = \{X \in \mathfrak{g} \mid [H, X] = \langle \alpha, H \rangle X \text{ for all } H \in \mathfrak{h}\}$$

is a *root space*. The nonzero elements of $\mathfrak{g}_\alpha$ are called *roots*.

Let E_{ij} be the $n \times n$ matrix with a one in the (i, j) position and zeros elsewhere. Let $H = \text{diag}(\lambda_1, \dots, \lambda_n)$ be in $\mathfrak{h}$. Then,

$$HE_{ij} = \lambda_i E_{ij} \quad \text{and} \quad E_{ij}H = \lambda_j E_{ij} \tag{1.1}$$

implying that

$$[H, E_{ij}] = (\lambda_i - \lambda_j)E_{ij}. \tag{1.2}$$

When $i \neq j$, $E_{ij} \in \mathfrak{sl}_n$. Therefore, E_{ij} is a root vector and $\lambda_i - \lambda_j$ is a root. That is, the roots of $\mathfrak{sl}_n$ are

$$\alpha_{ij} = e_i - e_j \quad i \neq j$$

where $e_1, \dots, e_n$ are the standard basis vectors for $\mathbb{C}^n$. These roots form the A_{n-1} root system which we shall now explicitly define.

Definition 1.14: Let $E \subseteq \mathbb{R}^n$ be the subspace of all vectors whose entries sum to zero. The roots of the *type A_{n-1} root system* are the vectors of the form

$$e_j - e_k, \quad j \neq k$$

where $e_1, \dots, e_n$ are the standard basis vectors. The A_{n-1} root system is generated by the vectors

$$e_1 - e_2, e_2 - e_3, \dots, e_{n-1} - e_n$$

which are called *simple roots*. The *rank* of the A_{n-1} root system is $n - 1$.

Because $\mathfrak{h}$ is abelian, the adjoint maps $\{\text{ad}_H : H \in \mathfrak{h}\}$ are simultaneously diagonalizable.

Proposition 1.15: [13, Proposition 7.16] *There is a direct sum decomposition of $\mathfrak{sl}_n$ given by*

$$\mathfrak{sl}_n = \mathfrak{h} \oplus \bigoplus_{\alpha \in R} \mathfrak{g}_\alpha$$

where R denotes the set of all roots.

1.1.4 Highest weight theory

The irreducible representations of $\mathfrak{sl}_n$ are classified by the *theory of highest weights*. Let $\mathfrak{h}$ be the Cartan subalgebra of $\mathfrak{sl}_n$ and R the associated root system. Let R^+ and R^- denote the positive and negative roots, respectively. For every $\alpha \in R$ there is a hyperplane H_α in $\mathbb{R}^n$ along with a reflection operator. The reflection operators generate the *Weyl group*. Each hyperplane H_α is a connected, open, polyhedral cone and is called a *Weyl chamber*.

For $\mathfrak{sl}_n$, the Weyl group is isomorphic to the symmetric group $\mathfrak{S}_n$. The *dominant Weyl chamber* consists of all $(n-1)$ -tuples $(\lambda_1, \dots, \lambda_{n-1})$ satisfying $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{n-1}$ and $\lambda_i - \lambda_{i+1} \in \mathbb{Z}_{\geq 0}$.

Theorem 1.16: [13, Theorem 9.4]

1. Every irreducible, finite-dimensional representation of $\mathfrak{sl}_n$ has a highest weight.
2. Two irreducible, finite-dimensional representations of $\mathfrak{sl}_n$ with the same highest weight are isomorphic.
3. There is a one-to-one correspondence between dominant integral weights and irreducible finite-dimensional representations of $\mathfrak{sl}_n$.

Theorem 1.16 says that the irreducible representations of $\mathfrak{sl}_n$ are in bijection with the dominant integral weights. Alternatively, the dominant integral weights can be viewed as integer partitions. Then, the irreducible representations are in natural bijection with Young diagrams.

Example 1.17: The irreducible representations of $\mathfrak{sl}_2$ are indexed by the natural numbers. Every dominant integral weight is some $k \in \mathbb{N}$. The corresponding irreducible representation of $\mathfrak{sl}_2$ is $\text{Sym}^k(\mathbb{C}^2)$. It is straightforward to verify that $\text{Sym}^k(\mathbb{C}^2)$ has highest weight k .

1.2 Representation theory in type A

As mentioned previously, this paper focusses on type A quantum groups. The representation theory of the quantum group $U_q(\mathfrak{sl}_n)$ is *almost* the same as the representation theory of the underlying Lie algebra $\mathfrak{sl}_n$. In particular, the

irreducible representations for the classical and quantum cases are in one-to-one correspondence, provided that q is generic. In particular, this statement is false when q is a root of unity. We shall assume that q is generic, except explicitly mentioned otherwise.

1.2.1 The representation category

It will be useful to consider the representation theory of $\mathfrak{sl}_n$ as a category, with representations as objects and intertwining maps as morphisms. If V, W are representations of $\mathfrak{sl}_n$, then $V \oplus W$, $V \otimes W$, and V^* are also representations of $\mathfrak{sl}_n$. We wish to define a category that allows for analogous operations on objects.

To take tensor products of representations, we need the representation category to have a *monoidal structure*.

Definition 1.18: A *monoidal category* is a category $\mathbf{C}$ along with

- a functor $\otimes : \mathbf{C} \times \mathbf{C} \rightarrow \mathbf{C}$ called the *monoidal product*;
- an object $\mathbb{1} \in \text{Ob}(\mathbf{C})$, called the *unit object*;
- a family of isomorphisms $\{\alpha_{X,Y,Z}\}$ called the *associativity constraint*;
- a family of isomorphisms called the *left unitality constraint*;
- a family of isomorphisms called the *right unitality constraint*.

defined so that certain coherence conditions are met. The isomorphisms in the associativity constraint are called *associators*. The isomorphisms in the left

(right) unitality constraint are called *left (right) unitors*.

In this paper, we assume that all monoidal categories are *strict*. That is, for any objects X, Y, Z in the category, $(X \otimes Y) \otimes Z = X \otimes (Y \otimes Z)$ and $\mathbb{1} \otimes X = X = X \otimes \mathbb{1}$. This means that the associators, left unitors, and right unitors are all identities.

To allow for dual representations, we require the representation category to be *rigid*:

Definition 1.19: A *rigid category* is a monoidal category in which every object A has a dual object A^* . Every object A has an *evaluation map* $\text{ev}_A : A^* \otimes A \rightarrow \mathbb{1}$ and a *coevaluation map* $\text{coev}_A : \mathbb{1} \rightarrow A \otimes A^*$.

At the level of vector spaces, $V^{**} \cong V$. Therefore, the representation category must be *pivotal*:

Definition 1.20: A *pivotal category* is a rigid category equipped with a monoidal natural isomorphism $A \rightarrow (A^*)^*$.

We are now ready to define the representation category of $\mathfrak{sl}_n$:

Definition 1.21: The *representation category* $\text{Rep}(\mathfrak{sl}_n)$ of $\mathfrak{sl}_n$ is the pivotal category whose objects are finite-dimensional representations of $\mathfrak{sl}_n$ and whose morphisms are intertwining maps.

Remark 1.22. We often denote $\text{Rep}(\mathfrak{sl}_n)$ by $\text{Rep}(\text{SL}_n)$. While this notation might appear sloppy, it is justified per the discussion in [Section 1.1.2](#). The representation theory of $\mathfrak{sl}_n$, SL_n , and $U(\mathfrak{sl}_n)$ are the “same” and therefore it

makes sense for the representation categories to have interchangeable notation. For the remainder of this paper, we shall always use the notation $\mathcal{R}ep(\mathrm{SL}_n)$.

Rather than working directly with $\mathcal{R}ep(\mathrm{SL}_n)$, we pass to a full subcategory whose idempotent completion recovers all of $\mathcal{R}ep(\mathrm{SL}_n)$ [8]. The objects in this subcategory are tensor products of the fundamental representations $\bigwedge^k \mathbb{C}^n$ of $\mathfrak{sl}_n$. The generating morphisms are

$$\bigwedge^k \mathbb{C}^n \otimes \bigwedge^l \mathbb{C}^n \rightarrow \bigwedge^{k+l} \mathbb{C}^n \quad \text{and} \quad \bigwedge^{k+l} \mathbb{C}^n \rightarrow \bigwedge^k \mathbb{C}^n \otimes \bigwedge^l \mathbb{C}^n$$

Combined with the duality, pairing, and copairing maps, these morphisms generate all SL_n -linear maps between the objects in the subcategory [17, Proposition 3.5.8].

1.2.2 Type A_1

The A_1 root system is associated to the Lie algebra $\mathfrak{sl}_2$. The generators for $\mathfrak{sl}_2$ are $H = \mathrm{diag}(1, -1)$, $X = E_{12}$, and $Y = E_{21}$. As usual, let $V = \mathbb{C}^2$ be the standard representation with basis $\{e_1, e_2\}$. Let $\alpha = e_1 - e_2$ be the positive simple root. The linear functional on $\mathfrak{h}$ corresponding to α is generated by $H \mapsto \lambda_1 - \lambda_2 = 2$.

Indeed, α is a root: $[H, X] = 2X = \alpha(H)X$. Additionally, $-\alpha$ is a root: $[H, Y] = -2Y = -\alpha(H)Y$. It is clear that $\mathfrak{sl}_2$ has the decomposition

$$\mathfrak{sl}_2 = \mathfrak{h} \oplus \mathfrak{g}_\alpha \oplus \mathfrak{g}_{-\alpha} \tag{1.3}$$

where each summand is invariant under $\mathfrak{h}$.

Remark 1.23. Given a semisimple Lie algebra $\mathfrak{g}$ and a root α , there exists a subalgebra of $\mathfrak{g}$ isomorphic to $\mathfrak{sl}_2$ [13, Theorem 7.19]. The subalgebra is generated by $\{H_\alpha, X_\alpha, Y_\alpha\}$ where $H_\alpha \in \mathfrak{h}$, $X_\alpha \in \mathfrak{g}_\alpha$, and $Y_\alpha \in \mathfrak{g}_{-\alpha}$. That is, the subalgebra is generated by a root $H_\alpha \in \mathfrak{h}$, a root vector X_α in the positive root space, and a root vector Y_α in the negative root space. As the notation suggests, H_α acts like H , X_α acts like X , and Y_α acts like Y in the previous example. Therefore, a thorough understanding of the representation theory of $\mathfrak{sl}_2$ provides valuable information about the representation theory of any semisimple Lie algebra.

Theorem 1.24: [11] *Any irreducible representation of $\mathfrak{sl}_2$ is a symmetric power of the standard representation $V \cong \mathbb{C}^2$.*

The integral elements in the dominant Weyl chamber are the nonnegative integers. By Theorem 1.16, the irreducible representations of $\mathfrak{sl}_2$ are in bijection with $\mathbb{Z}_{\geq 0}$. Explicitly, $k \in \mathbb{Z}_{\geq 0}$ is the irreducible representation $\text{Sym}^k(\mathbb{C}^2)$ with highest weight k .

1.2.3 Type A_{n-1}

Let $n \geq 2$. The A_{n-1} root system is associated to the Lie algebra $\mathfrak{sl}_n$. Let $\mathfrak{h}$ be the Cartan subalgebra and for each $i = 1, \dots, n$ define ϵ_i to be the linear functional on $\mathfrak{h}$ that sends $H \in \mathfrak{h}$ to the (i, i) entry of H . The roots for $\mathfrak{sl}_n$ are $\alpha_{i,j} = \epsilon_i - \epsilon_j$ assuming that $i \neq j$. Choose a set of simple roots $\{\alpha_1, \dots, \alpha_{n-1}\}$ defined by

$$\alpha_i = \epsilon_i - \epsilon_{i+1}.$$

Every simple root defines a triple $\{E_{i,i+1}, E_{i+1,i}, H_i\}$ which generates a subalgebra isomorphic to $\mathfrak{sl}_2$.

A basis for the *weight lattice* of $\mathfrak{sl}_n$ is given by the *fundamental weights* $\omega_1, \dots, \omega_{n-1}$. The irreducible representation of $\mathfrak{sl}_n$ corresponding to ω_k is the exterior power $\bigwedge^k(\mathbb{C}^n)$. Every irreducible representation of $\mathfrak{sl}_n$ appears as a summand of a tensor product of the fundamental representations. Given a length $(n-1)$ -tuple of nonnegative integers $(a_1, \dots, a_{n-1})$, the corresponding irreducible representation appears as a summand of

$$(V)^{\otimes a_1} \otimes (\bigwedge^2(V))^{\otimes a_2} \otimes \dots \otimes (\bigwedge^{n-1}(V))^{\otimes a_{n-1}} \quad (1.4)$$

Chapter 2

Quantum groups

A quantum group is a non-commutative, non-cocommutative Hopf algebra. Quantum groups were invented to systematically generate solutions to the Quantum Yang-Baxter equation. The classical Lie groups only give rise to trivial R -matrices, suggesting that a different mathematical framework was needed. One possible construction is the Drinfel'd-Jimbo type quantum group, which is constructed as a q -deformation of the universal enveloping algebra $U_q(\mathfrak{sl}_n)$. As $q \rightarrow 1$, the classical algebra $U_q(\mathfrak{sl}_n)$ is recovered. The Drinfel'd-Jimbo construction of $U_q(\mathfrak{sl}_n)$ is naturally equipped with a universal R -matrix that automatically satisfies the Yang-Baxter Equation. For more details and additional references, see [2]

2.1 A few helpful lemmas

When working with quantum groups, it is common to encounter quantum integers. There are related quantities in combinatorics called *Gaussian binomial coefficients* which have a number of nice properties.

Definition 2.1: The n th *quantum integer* is defined to be

$$[n]_q = \frac{q^n - q^{-n}}{q - q^{-1}}$$

or equivalently,

$$q^{n-1} + q^{n-3} + \cdots + q^{-(n-3)} + q^{-(n-1)}.$$

The *quantum factorial* is

$$[n]_q! = [n]_q [n-1]_q \cdots [2]_q [1]_q$$

and the *quantum binomial* is

$$\begin{bmatrix} n \\ k \end{bmatrix}_q = \frac{[n]_q!}{[k]_q! [n-k]_q!}$$

When q is clear from context, it can be dropped from the notation, writing $[n]$ instead of $[n]_q$. The above definition works for any non-negative integer n . If $n \leq 0$, then $[n] = [-n]$ by convention. For convenience, we collect a few lemmas that prove useful identities involving quantum integers.

Lemma 2.2: For any distinct integers a, b, c ,

$$[a+1][b-1][c] + [a][b+1][c-1] + [a-1][b][c+1] + [a+b+c] = [3][a][b][c].$$

Lemma 2.3: For each integer $n \geq 2$,

$$[2][n] = [n-1] + [n+1].$$

Proof. Observe:

$$\begin{aligned} [2][n] &= (q + q^{-1}) \left(\frac{q^n - q^{-n}}{q - q^{-1}} \right) \\ &= \frac{q^{n-1} - q^{-n+1}}{q - q^{-1}} + \frac{q^{n+1} - q^{-n-1}}{q - q^{-1}} \\ &= [n-1] + [n+1] \end{aligned}$$

□

In particular, note that

$$[2]^2 = [1] + [3] = 1 + [3].$$

Lemma 2.4:

$$[n+1] = [2][n] + [3][n-1] - [2]^2[n-1]$$

Proof. This easily follows from the identities

$$[2][n] = [n-1] + [n+1]$$

$$[2]^2 = 1 + [3]$$

□

2.2 $U_q(\mathfrak{sl}_n)$

Definition 2.5: The quantum group $U_q(\mathfrak{sl}_n)$ is a Hopf algebra generated by the symbols $E_i, F_i, K_i^\pm$ for $i = 1, \dots, n-1$ subject to the relations:

$$K_i K_j = K_j K_i, \quad K_j E_i K_j^{-1} = q^{\langle i, j \rangle} E_i, \quad K_j F_i K_j^{-1} = q^{-\langle i, j \rangle} F_i$$

$$[E_i, F_j] = \delta_{i,j} \frac{K_i - K_i^{-1}}{q - q^{-1}}$$

$$[2]E_i E_j E_i = E_i^2 E_j + E_j E_i^2 \text{ if } |i-j| = 1, \quad [E_i, E_j] = 0 \text{ if } |i-j| > 1$$

where

$$\langle i, j \rangle = \begin{cases} 2 & \text{if } i = j \\ -1 & \text{if } |i - j| = 1 \\ 0 & \text{otherwise} \end{cases}$$

The Hopf algebra structure has coproduct

$$\Delta(E_i) = E_i \otimes K_i + 1 \otimes E_i, \quad \Delta(F_i) = F_i \otimes 1 + K_i^{-1} \otimes F_i, \quad \Delta(K_i) = K_i \otimes K_i \quad (2.1)$$

antipode

$$S(K_i) = K_i^{-1}, \quad S(E_i) = -E_i K_i^{-1}, \quad S(F_i) = -K_i F_i \quad (2.2)$$

and counit

$$\epsilon(K_i) = 1, \quad \epsilon(E_i) = 0, \quad \epsilon(F_i) = 0 \quad (2.3)$$

Let $x_1, \dots, x_n$ be the standard basis of the $U_q(\mathfrak{sl}_n)$ -module $V = \mathbb{C}_q^n$. The generators for $U_q(\mathfrak{sl}_n)$ act on this basis as follows:

$$\begin{aligned} E_i(x_k) &= \begin{cases} x_{k-1} & \text{if } i = k - 1 \\ 0 & \text{otherwise} \end{cases} \\ F_i(x_k) &= \begin{cases} x_{k+1} & \text{if } i = k \\ 0 & \text{otherwise} \end{cases} \\ K_i(x_k) &= \begin{cases} qx_k & \text{if } i = k \\ q^{-1}x_k & \text{if } i = k - 1 \\ x_k & \text{otherwise} \end{cases} \end{aligned}$$

Combining these actions with the coproduct describes the action of $U_q(\mathfrak{sl}_n)$ on tensor products of the x_i . Similarly, the antipode gives an action of $U_q(\mathfrak{sl}_n)$ on the dual space V^* .

2.3 Representations of $U_q(\mathfrak{sl}_n)$

Let $\mathcal{Rep}(\mathrm{SL}_n)$ denote the representation category of $U_q(\mathfrak{sl}_n)$. The objects in $\mathcal{Rep}(\mathrm{SL}_n)$ are isomorphic to tensor products of the fundamental representations of $U_q(\mathfrak{sl}_n)$. Just as in the classical case, the fundamental representations of $U_q(\mathfrak{sl}_n)$ are $\bigwedge_q^k(\mathbb{C}_q^n)$ for $k = 1, \dots, n-1$. Morphisms in $\mathcal{Rep}(\mathrm{SL}_n)$ are built from the following generating morphisms:

$$M_{k,l} : \bigwedge_q^k(\mathbb{C}_q^n) \otimes \bigwedge_q^l(\mathbb{C}_q^n) \rightarrow \bigwedge_q^{k+l}(\mathbb{C}_q^n) \quad (2.4)$$

$$S_{k,l} : \bigwedge_q^{k+l}(\mathbb{C}_q^n) \rightarrow \bigwedge_q^k(\mathbb{C}_q^n) \otimes \bigwedge_q^l(\mathbb{C}_q^n) \quad (2.5)$$

$$D_k : \bigwedge_q^k(\mathbb{C}_q^n) \rightarrow (\bigwedge_q^{n-k}(\mathbb{C}_q^n))^* \quad (2.6)$$

Basis elements of $\bigwedge_q^k(\mathbb{C}_q^n)$ are of the form $x_{i_1} \wedge_q \dots \wedge_q x_{i_k}$ where $1 \leq i_1 < \dots < i_k \leq n$. In the quantum exterior algebra $\bigwedge_q^\bullet(\mathbb{C}_q^n)$,

$$x_i \wedge_q x_j = -q x_j \wedge_q x_i$$

whenever $i < j$ and $x_i \wedge_q x_i = 0$. By repeatedly applying this relation, any wedge product can be reordered (at the expense of some power of q) so that the indices appear in increasing order. For instance,

$$x_1 \wedge_q x_5 \wedge_q x_3 = -q^{-1} (x_1 \wedge_q x_3 \wedge_q x_5)$$

and

$$x_5 \wedge_q x_1 \wedge_q x_3 = (-q)^{-2} (x_1 \wedge_q x_3 \wedge_q x_5) = q^{-2} x_1 \wedge_q x_3 \wedge_q x_5.$$

An element of $\bigwedge_q^k(\mathbb{C}_q^n)$ with terms written in increasing order is called *standard*. The process of applying swaps to obtain a standard element is called *standardizing*.

Remark 2.6. This terminology is meant to agree with the language used in Young diagrams and Standard Monomial Theory.

More generally, let $\sigma \in \mathfrak{S}_k$ be a permutation and consider the element

$$x_{i_{\sigma(1)}} \wedge_q \cdots \wedge_q x_{i_{\sigma(k)}}$$

of $\bigwedge_q^k(\mathbb{C}_q^n)$, assuming $i_1 < \cdots < i_k$. Then,

$$x_{i_{\sigma(1)}} \wedge_q \cdots \wedge_q x_{i_{\sigma(k)}} = (-q)^{-s} x_{i_1} \wedge_q \cdots \wedge_q x_{i_k}$$

where s is the number of simple transpositions in $\mathfrak{S}_k$ needed for σ .

Example 2.7: Suppose we wish to standardize $x_5 \wedge_q x_1 \wedge_q x_3$. Let $i_1 = 1$, $i_2 = 3$, and $i_3 = 5$ and consider the permutation $\sigma = (1\ 2\ 3)^{-1} = (1\ 3\ 2) \in \mathfrak{S}_3$. Then, $\sigma = (2\ 3)(1\ 2)$ is the product of two simple transpositions and

$$\begin{aligned} x_5 \wedge_q x_1 \wedge_q x_3 &= x_{i_3} \wedge_q x_{i_1} \wedge_q x_{i_2} \\ &= x_{i_{\sigma(1)}} \wedge_q x_{i_{\sigma(2)}} \wedge_q x_{i_{\sigma(3)}}. \end{aligned}$$

Because σ is the product of two simple transpositions, there are two swaps required to sort the x_j . Therefore,

$$x_5 \wedge_q x_1 \wedge_q x_3 = (-q)^{-2} x_{i_1} \wedge_q x_{i_2} \wedge_q x_{i_3} = q^{-2} x_1 \wedge_q x_3 \wedge_q x_5.$$

Remark 2.8. The fact that any wedge product can be standardized means that elements of $\bigwedge^k(V)$ are defined by specifying a subset of $\{1, \dots, n\}$ of cardinality k . If $T = \{i_1, \dots, i_k\}$, and $i_1 < \dots < i_k$, define

$$x_T = x_{i_1} \wedge_q \dots \wedge_q x_{i_k}.$$

The merge morphism $M_{k,l} : \bigwedge_q^k(\mathbb{C}_q^n) \otimes \bigwedge_q^l(\mathbb{C}_q^n) \rightarrow \bigwedge_q^{k+l}(\mathbb{C}_q^n)$ in [Equation 2.4](#) is defined by

$$M_{k,l}(x_K \otimes x_L) \mapsto \begin{cases} (-q)^{\ell(K,L)} x_{K \sqcup L} & K \cap L = \emptyset \\ 0 & \text{otherwise} \end{cases} \quad (2.7)$$

where K and L are subsets of $\{1, \dots, n\}$ of sizes k and l , respectively. The split morphism $S_{k,l} : \bigwedge_q^{k+l}(\mathbb{C}_q^n) \rightarrow \bigwedge_q^k(\mathbb{C}_q^n) \otimes \bigwedge_q^l(\mathbb{C}_q^n)$ in [Equation 2.5](#) is

$$S_{k,l}(x_T) = (-1)^{kl} \sum_{K \subset T} (-q)^{-\ell(T \setminus K, K)} x_K \otimes x_{T \setminus K}$$

where T is a subset of size $k+l$ and $K \subset T$ ranges over all subsets of size k .

Finally, $D_k : \bigwedge_q^k(\mathbb{C}_q^n) \rightarrow (\bigwedge_q^{n-k}(\mathbb{C}_q^n))^*$ in [Equation 2.6](#) is given by

$$D_k(x_K)(x_T) = \begin{cases} (-q)^{\ell(K,T)} & \text{if } K \cap T = \emptyset \\ 0 & \text{otherwise} \end{cases}$$

where K and T are subsets of cardinalities k and $n-k$, respectively.

Chapter 3

Spiders

3.1 A brief history of spiders

Spiders were first defined in [15] for the rank 2 simple Lie algebras, A_2 , B_2 , and G_2 . This was later generalized to Lie algebras of type A_n in [8] using quantum skew Howe duality.

3.2 The SL_n spider

We begin by recalling the free spider category $\mathcal{FS}p_n$ as defined in [8]. The objects in $\mathcal{FS}p_n$ are finite sequences $\underline{k}$ in $\{1^\pm, \dots, (n-1)^\pm\}$. Morphisms are ($\mathbb{C}_q$ -linear combinations of) oriented planar graphs locally modeled on the vertices depicted in Figure 3.1.



Figure 3.1: The possible vertices in a web.

Morphisms in $\text{Hom}(\underline{k}, \underline{l})$ are called webs. The bottom boundary of a web in $\text{Hom}(\underline{k}, \underline{l})$ is $\underline{k}$ where the sign of each entry determines whether the strand points up (positive) or down (negative). The top boundary is $\underline{l}$ and is constructed similarly.

The spider category $\mathcal{S}p_n$ is the quotient of $\mathcal{F}Sp_n$ modulo the following relations, along with all mirror reflections and arrow reversals.

$$\begin{array}{c} \nearrow \\ \nearrow \\ \nearrow \\ \nearrow \\ \nearrow \end{array} \begin{array}{c} n-k \\ \\ \\ k \end{array} = (-1)^{k(n-k)} \begin{array}{c} \nearrow \\ \nearrow \\ \nearrow \\ \nearrow \\ \nearrow \end{array} \begin{array}{c} n-k \\ \\ \\ k \end{array} \quad (3.1)$$

$$\begin{array}{c} \nearrow k+l \\ \circlearrowleft \\ \nwarrow k \\ \circlearrowright \\ \nwarrow l \\ \nearrow k+l \end{array} = \left[\begin{array}{c} k+l \\ l \end{array} \right]_q \nearrow k+l \quad (3.2)$$

$$\begin{array}{c} \diagup k \\ | \\ \textcircled{\text{---}} \\ | \\ \diagdown k \end{array} \begin{array}{c} l \\ | \\ \text{---} \\ | \\ k+l \end{array} = \left[\begin{array}{c} n-k \\ l \end{array} \right]_q \begin{array}{c} \diagup \\ | \\ \text{---} \\ | \\ \diagdown k \end{array} \quad (3.3)$$

$$\begin{array}{c}
k+l+m \\
\uparrow \\
\begin{array}{ccc}
k+l & & \\
\swarrow & \searrow & \\
k & l & m
\end{array}
\end{array}
=
\begin{array}{c}
k+l+m \\
\uparrow \\
\begin{array}{ccc}
& & l+m \\
\swarrow & \searrow & \\
k & l & m
\end{array}
\end{array}
\quad (3.4)$$

$$\begin{array}{c} n-k-l \\ | \\ \text{---} \\ | \\ \text{---} \\ / \quad \backslash \\ k \quad l \end{array} \quad \begin{array}{c} k+l \\ | \\ \text{---} \\ | \\ \text{---} \\ / \quad \backslash \\ k \quad l \end{array} = \begin{array}{c} n-k-l \\ | \\ \text{---} \\ | \\ \text{---} \\ / \quad \backslash \\ k \quad l \end{array} \quad \begin{array}{c} n-l \\ | \\ \text{---} \\ | \\ \text{---} \\ / \quad \backslash \\ k \quad l \end{array} \quad (3.5)$$

$$\begin{array}{c} n-l \\ \downarrow \\ l \\ \swarrow \searrow \\ k+l \quad k \end{array} = \begin{array}{c} n-l \\ \downarrow \\ n-k-l \\ \swarrow \searrow \\ k+l \quad k \end{array} \quad (3.6)$$

$$\begin{array}{c} k-s-r \quad l+s+r \\ \uparrow \quad \uparrow \\ r \quad s \\ \downarrow \quad \downarrow \\ k \quad l \end{array} = \left[\begin{array}{c} r+s \\ r \end{array} \right]_q \begin{array}{c} k-s-r \quad l+s+r \\ \uparrow \quad \uparrow \\ r+s \\ \downarrow \quad \downarrow \\ k \quad l \end{array} \quad (3.7)$$

$$\begin{array}{c} k-s+r \quad l+s-r \\ \uparrow \quad \uparrow \\ r \quad s \\ \downarrow \quad \downarrow \\ k \quad l \end{array} = \sum_t \left[\begin{array}{c} k-l+r-s \\ t \end{array} \right]_q \begin{array}{c} k-s+r \quad l+s-r \\ \uparrow \quad \uparrow \\ s-t \quad r-t \\ \downarrow \quad \downarrow \\ k+r-t \quad l-r+t \\ k \quad l \end{array} \quad (3.8)$$

These relations can be used to deduce the following identities, which appear quite often and thus are worth explicitly mentioning:

$$\begin{array}{c} k \\ \circlearrowleft \end{array} = \left[\begin{array}{c} n \\ k \end{array} \right]_q \quad (3.9)$$

$$\begin{array}{c} k \\ \nearrow \\ n-k \\ \nwarrow \\ k \end{array} = \begin{array}{c} k \\ \nearrow \end{array} \quad (3.10)$$

Consider the functor $\Gamma : \mathcal{S}p(\mathrm{SL}_n) \rightarrow \mathcal{R}ep(\mathrm{SL}_n)$ that acts on objects by

$$(k_1^{\epsilon_1}, \dots, k_m^{\epsilon_m}) \mapsto (\wedge^{k_1} \mathbb{C}_q^n)^{\epsilon_1} \otimes \dots \otimes (\wedge^{k_m} \mathbb{C}_q^n)^{\epsilon_m}$$

and on the generating morphisms by

$$\begin{array}{c} k+l \\ \uparrow \\ \swarrow \quad \searrow \\ k \quad l \end{array} \mapsto M_{k,l} \quad \text{and} \quad \begin{array}{c} k \quad l \\ \swarrow \quad \searrow \\ \uparrow \\ k+l \end{array} \mapsto S_{k,l}$$

Theorem 3.1: [8] *The functor $\Gamma : \mathcal{Sp}(\mathrm{SL}_n) \rightarrow \mathcal{Rep}(\mathrm{SL}_n)$ is an equivalence of pivotal categories.*

3.2.1 A different set of relations

The relations defined in [8] are equivalent to the relations given in [1]. The “new” spider relations have a different tag convention, but all webs can be converted back and forth. The “new” relations are simpler to work with. We state them here for convenience:

Tag switch:

$$\begin{array}{c} n-k \\ \downarrow \\ \text{---} \\ \uparrow \\ k \end{array} = (-1)^{k(n-k)} \begin{array}{c} n-k \\ \downarrow \\ \text{---} \\ \downarrow \\ k \end{array} \quad (3.11)$$

Tag cancellation:

$$\begin{array}{c} k \\ \uparrow \\ \text{---} \\ \downarrow \\ \uparrow \\ k \end{array} = \begin{array}{c} k \\ \uparrow \\ \text{---} \\ \uparrow \\ k \end{array} \quad (3.12)$$

I=H relation:

$$\begin{array}{c} k \\ \swarrow \quad \searrow \\ \text{---} \uparrow \text{---} \\ \swarrow \quad \searrow \\ l \quad m \end{array} = \begin{array}{c} k \\ \swarrow \quad \searrow \\ \text{---} \text{---} \text{---} \\ \swarrow \quad \searrow \\ l \quad m \end{array} \quad (3.13)$$

Bursting a digon:

$$\begin{array}{c} k \\ \uparrow \\ \text{---} \circ \text{---} 1 \\ \downarrow \end{array} = [k] \begin{array}{c} k \\ \uparrow \\ \text{---} \\ \downarrow \end{array} \quad (3.14)$$

Bursting a square:

$$\begin{array}{c} k \quad 1 \quad 1 \\ \swarrow \quad \uparrow \quad \searrow \\ \text{---} \circ \text{---} \\ \swarrow \quad \uparrow \quad \searrow \\ k \quad 1 \quad 1 \end{array} = \begin{array}{c} k \quad 1 \\ \swarrow \quad \uparrow \quad \searrow \\ \text{---} \\ \swarrow \quad \uparrow \quad \searrow \\ k \quad 1 \end{array} + [k-1] \begin{array}{c} k \\ \uparrow \end{array} \begin{array}{c} 1 \\ \uparrow \end{array} \quad (3.15)$$

Proposition 3.2: *[1, Proposition 1] The relations listed above are equivalent to the defining relations given in [8], with a different convention for tags.*

Chapter 4

Diagrammatic computations of Ocneanu cells

4.1 Ocneanu cells

Consider SL_3 and let ρ denote the standard representation of SL_3 . As discussed in [9], there is a MacKay graph $\mathcal{G}$ associated with SL_3 whose vertices are labeled by irreducible representations of SL_3 . The number of edges between two vertices i, j in $\mathcal{G}$ encodes information about the multiplicity of i in the decomposition of $j \otimes \rho$ into irreducible representations. Because $\rho \otimes \rho \otimes \rho$ is the trivial representation, the graph $\mathcal{G}$ is built from triangles, each of which has a complex number called an *Ocneanu cell*. The Ocneanu cells for SL_3 are computed in [9] using algebraic methods. For a more detailed discussion on Ocneanu cells and their applications in physics see [18, 19, 20, 21].

More recently, a new type of diagram called a *cobweb* was defined [1]. With a slightly different set of relations, cobwebs provide a diagrammatic method to compute the Ocneanu cells for SL_n .

4.2 Cobwebs

Fix $n \geq 3$ and let $\mathcal{N} = \{1, \dots, n\}$. For each $\{i, j\} \subseteq \mathcal{N}$.

Definition 4.1: Let $K \subseteq \mathbb{N}$ be a set with $|K| \geq 2$. A *color* in K is a subset $\{i, j\} \subseteq K$ with $i \neq j$. Each color has two orientations, denoted $c_{i,j}$ and $c_{j,i}$, which should be thought of as the same color, but oriented in opposite directions.

Recall that the strands in a web in the SL_n are labeled by $k \in \{1, \dots, n-1\}$. In the representation category, these labels correspond to the fundamental representations of SL_n . We now define another type of diagram, called a cobweb where strands are labeled by colors.

Definition 4.2: A *cobweb* is a collection of oriented strands drawn in the plane. Each strand is labeled by a color $c_{i,j}$ with $i, j \in \{1, \dots, n\}$. A cobweb can have virtual crossings and bivalent vertices. In a bivalent vertex, two strands of the same color, but opposite orientations, meet. That is, a bivalent vertex has two strands, both oriented in or both oriented out, labeled $c_{i,j}$ and $c_{j,i}$ and a tag that points to one side.

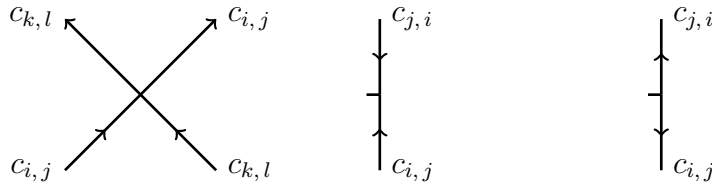


Figure 4.1: Virtual crossings and bivalent vertices in a cobweb.

Cobwebs can be simplified locally using the following relations:

$$\begin{array}{c} \uparrow \\ | \\ \hline \end{array} = (-1) \begin{array}{c} \uparrow \\ | \\ \hline \end{array} \quad (4.1)$$

$$\begin{array}{c} \uparrow \\ | \\ \hline \\ \hline \end{array} = \begin{array}{c} \uparrow \\ | \\ \hline \end{array} \quad (4.2)$$

$$\begin{array}{c} c_{i,j} \\ \circlearrowleft \end{array} + \begin{array}{c} c_{i,j} \\ \circlearrowleft \end{array} = [2] \quad (4.3)$$

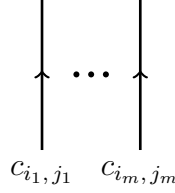
$$\begin{array}{c} c_{i,j} \\ \circlearrowleft \end{array} \begin{array}{c} c_{i,l} \\ \circlearrowleft \end{array} + \begin{array}{c} c_{i,j} \\ \circlearrowleft \end{array} \begin{array}{c} c_{j,l} \\ \circlearrowleft \end{array} + \begin{array}{c} c_{i,l} \\ \circlearrowleft \end{array} \begin{array}{c} c_{j,l} \\ \circlearrowleft \end{array} = [3] \quad (4.4)$$

$$\begin{array}{c} \circlearrowleft \\ \uparrow \end{array} \begin{array}{c} \uparrow \\ \downarrow \end{array} = \begin{array}{c} \uparrow \downarrow \end{array} \quad (4.5)$$

Equation 4.1 is called tag switching, Equation 4.2 is called tag cancellation, Equation 4.3 and Equation 4.4 are called [2]-bubble and [3]-bubble relations respectively, and Equation 4.5 is called the pop-switch relation. The pop-switch relation and [2]-bubble relation come from [12]. Tag switching and tag cancellation were defined in [1].

4.2.1 Some notational conveniences

Let K, L be disjoint subsets of $\{1, \dots, n\}$. It will be convenient to have a diagrammatic shorthand for a collection of strands in a cobweb



where c_{i_s, j_s} ranges over all $i_s \in K$ and $j_s \in L$. The abbreviated diagram for this cobweb is



The shorthand when c_{i_s, j_s} ranges over all $i_s \in K$ and $j_s \notin L$ is



where the shaded background stands for the complement of the set.

To emphasize that the previous diagram has several strands, the lines are drawn thicker. The diagram should be interpreted as a collection of strands in a cobweb, all with the same orientation, with labels $c_{k, l}$ ranging over all $k \in K$, $l \in L$. If the strands have labels of the form $c_{i, j}$ where $i \in K$ and $j \notin K$, then the diagram



can be used instead.

Remark 4.3. The diagrammatic shorthand introduced in this section is well defined because strands that are labeled by distinct colors can freely pass over each other.

It will also be convenient to have shorthand for a specific kind of diagram. Let $i, j \in \{1, \dots, n\}$ be distinct. Let $(i\ j)$ denote the clockwise cobweb bubble with label $c_{i,j}$. The bubble bursting relation immediately implies that

$$(i\ j) + (j\ i) = [2]$$

since a bubble's orientation can be reversed if the order of i and j is swapped.

Let D be the diagram with two positively-oriented bubbles, labeled $c_{i,j}$ and $c_{k,l}$. Assuming that the bubbles commute, we can denote D by $(i\ j)(k\ l)$. Recall that the pair of bubbles commutes as long as they are different colors, or whenever $\{i, j\} \neq \{k, l\}$. This notation extends to allow for a diagram with any number of commuting bubbles.

Note that this is *not* the same thing as cycle notation in a permutation group. Namely, for transpositions, $(1\ 2)(1\ 3) \neq (1\ 3)(1\ 2)$ but the diagrams corresponding to $(1\ 2)(1\ 3)$ and $(1\ 3)(1\ 2)$ are equivalent.

If a diagram D contains k bubbles, we say that D has *degree* k .

4.2.2 A few more relations

Cobwebs are highly combinatorial. For instance, there are $\binom{n}{2}$ relations of the form

$$(a\ b) + (b\ a) = [2]$$

and $\binom{n}{3}$ relations of the form

$$(b\ a)(c\ a) + (a\ b)(c\ b) + (a\ c)(b\ c) = [3].$$

These relations are indexed by subsets of $\mathcal{N}$ of cardinality 2 and 3, respectively. We denote the $[2]$ -bubble relation on $\{a, b\}$ by $\text{Rel}([2], \{a, b\})$. Likewise, the $[3]$ -bubble relation on $\{a, b, c\}$ is denoted by $\text{Rel}([3], \{a, b, c\})$. For each $4 \leq k \leq n$ and subset $K \subseteq \mathcal{N}$ of cardinality k , there is a $[k]$ -bubble relation,

$$\sum_{a \in K} \prod_{\substack{i \in K \\ i \neq a}} (i\ a) = [k]$$

denoted by $\text{Rel}([k], K)$. The remainder of this section is dedicated to proving that any $\text{Rel}([k], K)$ follows from the cobweb relations. Before doing so, we introduce some helpful terminology and notation.

Definition 4.4: Let $2 \leq k \leq n$ and $K \subseteq \mathcal{N}$ be a subset of k . The $[k]$ -bubble relation, denoted $\text{Rel}([k], K)$ is

$$\sum_{a \in K} \prod_{\substack{i \in K \\ i \neq a}} (i\ a) = [k].$$

Observe that $\text{Rel}([k], K)$ has k terms, each of degree $k - 1$.

Lemma 4.5: Let $n = 4$. Then,

$$\sum_{a \in K} \prod_{\substack{i \in K \\ i \neq a}} (i\ a) = [k] \tag{4.6}$$

for any subset $K \subseteq \{1, 2, 3, 4\}$ of cardinality 2, 3, or 4.

Proof. If $k = 2$ and $K = \{a, b\}$, the left side of Equation 4.14 is

$$(a \ b) + (b \ a)$$

which equals $[2]$ following directly from the cobweb $[2]$ -bubble relation $\text{Rel}([2], \{a, b\})$.

When $k = 3$ and K is a subset of cardinality k , the result is implied by the $[3]$ -bubble relation $\text{Rel}([3], K)$.

Let $K = \{1, 2, 3, 4\}$. We must show $\text{Rel}([4], K)$ holds. That is,

$$(2 \ 1)(3 \ 1)(4 \ 1) + (1 \ 2)(3 \ 2)(4 \ 2) + (1 \ 3)(2 \ 3)(4 \ 3) + (1 \ 4)(2 \ 4)(3 \ 4) = [4].$$

There are four terms on the left side of the equation, and each term is the product of three bubbles. Let c_i denote the i th term of this sum. For each subset of $\{1, 2, 3, 4\}$ with cardinality three, there is a corresponding $[3]$ -bubble relation. For instance, the $\text{Rel}([3], \{1, 2, 3\})$ can be applied to $c_1 = (2 \ 1)(3 \ 1)(4 \ 1)$ to see that

$$(2 \ 1)(3 \ 1) = [3] - (1 \ 2)(3 \ 2) - (1 \ 3)(2 \ 3).$$

By applying $\text{Rel}([3], K')$ for various sets $K' \subseteq K$ of cardinality 3, we find that

$$c_1 = (4 \ 1)([3] - (1 \ 2)(3 \ 2) - (1 \ 3)(2 \ 3)) \tag{4.7}$$

$$c_2 = (3 \ 2)([3] - (2 \ 1)(4 \ 1) - (2 \ 4)(1 \ 4)) \tag{4.8}$$

$$c_3 = (2 \ 3)([3] - (3 \ 1)(4 \ 1) - (3 \ 4)(1 \ 4)) \tag{4.9}$$

$$c_4 = (1 \ 4)([3] - (4 \ 2)(3 \ 2) - (4 \ 3)(2 \ 3)) \tag{4.10}$$

Notice that each expression now has a term of the form $(a \ b)[3]$. Grouping

these terms together and applying the $[2]$ -bubble relations yields

$$((4\ 1) + (1\ 4) + (3\ 2) + (2\ 3))[3] = 2[2][3].$$

Then,

$$\sum_{i=1}^4 c_i = 2[2][3] - (4\ 1)(3\ 2)((1\ 2) + (2\ 1)) - (2\ 3)(1\ 4)((3\ 4) + (4\ 3)) \quad (4.11)$$

$$\begin{aligned} & - (4\ 1)(2\ 3)((1\ 3) + (3\ 1)) - (1\ 4)(3\ 2)((2\ 4) + (4\ 2)) \\ & = 2[2][3] - [2]((1\ 4) + (4\ 1))((2\ 3) + (3\ 2)) \end{aligned} \quad (4.12)$$

$$= 2[2][3] - [2]^3 \quad (4.13)$$

By [Lemma 2.4](#), the proof is complete. $\square$

Lemma 4.6: Fix $n \geq 3$. Then, for each $k \in \{1, \dots, n\}$ and any subset $K \subseteq \{1, \dots, n\}$ of cardinality k the equation

$$\sum_{a \in K} \prod_{\substack{i \in K \\ i \neq a}} (i\ a) = [k] \quad (4.14)$$

is satisfied.

Remark 4.7. Without loss of generality, it can be assumed that $K = \{1, \dots, k\}$. A different choice in K is simply a relabeling of the colors and does not affect the validity of the equation. We proceed by induction on n . For a fixed n , it suffices to prove [Equation 4.14](#) when $k = n$ and $K = \{1, \dots, n\}$ as the cases when $k < n$ follow from the induction hypothesis. [Lemma 4.5](#) is the base case.

Proof. Assume that the result holds for each $m \leq n$. That is, for any subset $M \subseteq \{1, \dots, n+1\}$ of cardinality m we assume $\text{Rel}([m], M)$ is true. We shall

prove that

$$\sum_{a=1}^{n+1} \prod_{i \neq a} (i \ a) = [n+1] \quad (4.15)$$

Define $K = \{1, \dots, n+1\}$, $K_1 = \{1, \dots, n\}$, and $K_{n+1} = \{2, \dots, n+1\}$. For $i = 2, \dots, n$ define $K_i = \{1, i, n+1\}$. Let $c_i = \prod_{a \neq i} (a \ i)$ be the i th term in [Equation 4.15](#).

First consider c_1 . Applying $\text{Rel}([n], K_1)$ to c_1 we find that

$$c_1 = (2 \ 1) \cdots (n \ 1) (n+1 \ 1) \quad (4.16)$$

$$= \left([n] - \sum_{i \in K_1 \setminus \{1\}} \prod_{j \neq i} (j \ i) \right) (n+1 \ 1) \quad (4.17)$$

Similarly, if we apply $\text{Rel}([n], K_{n+1})$ to c_{n+1} ,

$$c_{n+1} = (2 \ n+1) \cdots (n \ n+1) (1 \ n+1) \quad (4.18)$$

$$= \left([n] - \sum_{i \in K_{n+1} \setminus \{n+1\}} \prod_{j \neq i} (j \ i) \right) (1 \ n+1) \quad (4.19)$$

Next, let $i \in \{2, \dots, n\}$ and recall $\text{Rel}([3], K_i)$:

$$(i \ 1) (n+1 \ 1) + (1 \ i) (n+1 \ i) + (1 \ n+1) (i \ n+1) = [3]. \quad (4.20)$$

Then,

$$c_i = (1 \ i) (n+1 \ i) \prod_{a \notin K_i} (a \ i) \quad (4.21)$$

$$= ([3] - (i \ 1) (n+1 \ 1) - (1 \ n+1) (i \ n+1)) \left(\prod_{a \notin K_i} (a \ i) \right) \quad (4.22)$$

where [Equation 4.22](#) is obtained via [Equation 4.20](#).

As seen in Equation 4.17, c_1 contributes one term of degree 1 and $n-1$ terms of degree n . Likewise, Equation 4.19 shows that c_{n+1} contributes one term of degree 1 and $n-1$ terms of degree n . For each $i = 2, \dots, n$, Equation 4.22 shows that c_i contributes one term of degree $n-2$ and two terms of degree n .

First consider the degree one terms from c_1 and c_{n+1} . The sum of the degree one terms is

$$[n] (n+1 \ 1) + [n] (1 \ n+1) = [2][n]. \quad (4.23)$$

Let $i \in \{2, \dots, n\}$ and consider the degree $n-2$ term in c_i :

$$[3] \left(\prod_{a \notin K_i} (a \ i) \right).$$

By applying $\text{Rel}([n-1], \{2, \dots, n\})$, we see that the sum of the degree $n-2$ terms is

$$[3] \left(\sum_{i=2}^n \prod_{a \notin K_i} (a \ i) \right) = [3][n-1]. \quad (4.24)$$

The remaining terms in $\sum c_i$ are each of degree n . Fix $i \in \{2, \dots, n\}$. From c_1 , there is

$$- \left(\prod_{a \in K_1 \setminus \{i\}} (a \ i) \right) (n+1 \ 1) = - (n+1 \ 1) (1 \ i) \left(\prod_{a \notin K_i} (a \ i) \right) \quad (4.25)$$

and from c_{n+1} ,

$$- \left(\prod_{a \in K_{n+1} \setminus \{i\}} (a \ i) \right) (1 \ n+1) = - (1 \ n+1) (n+1 \ i) \left(\prod_{a \notin K_i} (a \ i) \right) \quad (4.26)$$

There are two terms of degree n in c_i :

$$- (i \ 1) (n+1 \ 1) \left(\prod_{a \notin K_i} (a \ i) \right) \quad (4.27)$$

and

$$-(1 \ n+1) (i \ n+1) \left(\prod_{a \notin K_i} (a \ i) \right) \quad (4.28)$$

The sum of Equations 4.25, 4.26, 4.27, and 4.28 is

$$-[2]^2 \prod_{a \notin K_i} (a \ i) \quad (4.29)$$

after applying $\text{Rel}([2], \{1, i\})$, $\text{Rel}([2], \{1, n+1\})$, and $\text{Rel}([2], \{i, n+1\})$. In total, there are $4(n-1)$ terms of degree n in $\sum c_i$. The sum of all degree n terms is

$$\sum_{i=2}^n \left(-[2]^2 \prod_{a \notin K_i} (a \ i) \right) = -[2]^2 [n-1] \quad (4.30)$$

using $\text{Rel}([n-1], \{2, \dots, n\})$. A quick counting argument verifies that all terms appearing in $\sum c_i$ have been accounted for. Therefore, Equations 4.23, 4.24, and 4.30 along with Lemmas 2.3 and 2.4 imply that

$$\sum_{i=1}^{n+1} c_i = [2][n] + [3][n-1] - [2]^2[n-1] = [n+1]$$

as desired. $\square$

4.3 From Webs to Cobwebs

We must show that the map defined in [1] is still well-defined, with the new set of cobweb relations. For clarity, we restate the map's definition before proving it is well-defined. The map will be defined in terms of a state sum.

4.3.1 State sums

Definition 4.8: A *state* σ on a web W is an assignment of subsets of $\mathcal{N} = \{1, \dots, n\}$ to each strand, satisfying the following:

- (i) A strand labeled k is assigned a set of cardinality k .
- (ii) The strands at a trivalent vertex are assigned sets K, L , and $K \sqcup L$.
- (iii) The strands at a bivalent vertex are assigned sets K and $\{1, \dots, n\} \setminus K$.

Implicitly, (ii) requires that K and L are disjoint.

Definition 4.9: The *boundary data* of a web W is an assignment of subsets of $\mathcal{N}$ to each strand in W that intersects the boundary. These assignments must satisfy the definition of a state on W .

Given a relation of webs, we call a set of boundary data *valid* if it assigns each boundary strand the same set in every diagram in the relation. A *state consistent with the chosen boundary data* is an assignment of subsets of $\mathcal{N}$ to all interior strands that satisfies the definition of a state. A state sum for a set of boundary data is taken over all states that are consistent with the choice in boundary data. If there are no such states, the state sum is zero.

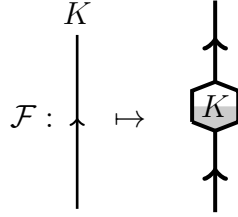
4.3.2 The Map

The map $\mathcal{F}$ sends each spider web W to a linear combination of cobwebs, determined by the terms in the state sum. Explicitly,

$$\mathcal{F} : W \mapsto \sum_{\text{states } \sigma} \text{Cob}(W, \sigma)$$

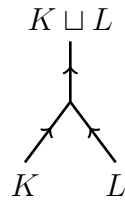
where $\text{Cob}(W, \sigma)$ is the cobweb for W associated to the state σ . It suffices to describe the action of $\mathcal{F}$ on strands and vertices in a spider web. To apply $\mathcal{F}$ to a web, we apply $\mathcal{F}$ locally to each strand/vertex and then connect the resulting strands in the cobweb. Following the discussion in [1], extending $\mathcal{F}$ to a general web is well-defined.

A strand assigned a set K is mapped to a collection of parallel strands in the cobweb, labeled by $c_{i,j}$ for every $i \in K, j \notin K$.

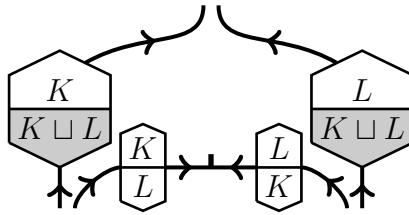


Recall that $\boxed{K}$ is shorthand for a collection of parallel strands each labeled by a root $c_{i,j}$ for $i \in K, j \notin K$. Since the strands are of distinct colors, they can be freely reordered.

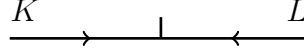
A trivalent vertex



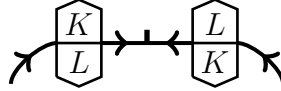
is mapped to a cobweb with the following strands:



A bivalent vertex



is mapped to a cobweb with the following strands:



4.4 Computing Ocneanu cells with Cobwebs

To prove that $\mathcal{F}$ is well-defined, we must show that it respects each of the relations in the spider. That is, for each spider relation we must show that the left and right sides of the equation get sent to the same linear combination of cobwebs.

4.4.1 Orientation reversal, tag operations, and the $I = H$ relation

The proofs in [1] for strands labeled n , reversing all orientations, tag operations, and the $I = H$ relation do not use the bubble bursting relation. Therefore, the proof that the map is well-defined with respect to each of these spider relations remains the same with our new cobweb relations.

4.4.2 Bursting a digon

Consider the digon relation:

$$\begin{array}{c} \uparrow k \\ \circlearrowleft 1 \\ \uparrow \end{array} = [k] \begin{array}{c} \uparrow k \\ | \\ \uparrow \end{array} \quad (4.31)$$

We proceed by induction on $n \geq 3$. A stated web in the SL_n spider assigns each strand a subset of $\{1, \dots, n\}$. It suffices to prove that the digon relation holds for $k = n - 1$. When $k < n - 1$, we can view the diagram as a web in the SL_{n-1} spider and obtain a valid state (perhaps relabeling to ensure that the state is actually a state in the SL_{n-1} spider).

Let D_L and D_R denote the diagrams on the left and right sides of [Equation 4.31](#), respectively. The proof strategy is as follows:

1. Assign D_L and D_R an arbitrary set of (valid) boundary data.
2. For each state σ consistent with the chosen boundary data, compute $\mathrm{Cob}(D, \sigma)$. Then, $\mathcal{F}(D)$ is the sum over all such cobwebs:

$$\mathcal{F}(D) = \sum_{\text{states } \sigma} \mathrm{Cob}(D, \sigma)$$

3. Verify that $\mathcal{F}(D_L) = \mathcal{F}(D_R)$ by checking that the strands in the cobwebs agree.

To better understand the proof strategy, consider the digon relation in the SL_3 -spider. The following example gives explicit details of valid boundary data, states consistent with a given set of boundary data, and the resulting cobwebs. These computations are not specific to the digon relation and give insight for how to perform cobweb computations.

Example 4.10: Let $n = 3$. Strands in the SL_3 spider have labels $k \in \{1, 2\}$. When $k = 1$, the left side of the bubble has label 0 and thus can be deleted, making the relation trivially true.

Now let $k = 2$. Assign the top and bottom strands of both diagrams a set $K \subseteq \mathcal{N}$ of cardinality k . There are $\binom{3}{2} = 3$ possible sets of valid boundary data:

$$\begin{array}{cccccc}
 \{1, 2\} & & \{1, 2\} & & \{1, 3\} & & \{1, 3\} & & \{2, 3\} & & \{2, 3\} \\
 \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\
 \circlearrowleft & = [2] & \uparrow & \circlearrowleft & = [2] & \uparrow & \circlearrowleft & = [2] & \uparrow & \circlearrowleft & = [2] & \uparrow \\
 \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\
 \{1, 2\} & & \{1, 2\} & & \{1, 3\} & & \{1, 3\} & & \{2, 3\} & & \{2, 3\}
 \end{array}$$

For simplicity, set $K = \{1, 2\}$. Choosing a different set of boundary data does not change the proof. For instance, the proof when $K = \{2, 3\}$ is the exact same, after replacing each 1 with a 2, and each 2 with a 3.

A state on the web on the left side of the equation corresponds to a choice of $a \in K$. For the chosen state, assign $K \setminus \{a\}$ to the left side of the digon and $\{a\}$ to the right side. Then,

$$\begin{aligned}
& \begin{array}{c} \{1, 2\} \\ \uparrow \\ \text{cobweb} \\ \uparrow \\ \{1, 2\} \end{array} \xrightarrow{\mathcal{F}} \begin{array}{c} c_{1,3} \quad c_{2,3} \\ \uparrow \quad \uparrow \\ \text{cobweb} \\ \uparrow \quad \uparrow \\ c_{1,3} \quad c_{2,3} \end{array} + \begin{array}{c} c_{2,3} \quad c_{1,3} \\ \uparrow \quad \uparrow \\ \text{cobweb} \\ \uparrow \quad \uparrow \\ c_{2,3} \quad c_{1,3} \end{array} \\
&= \begin{array}{c} c_{1,3} \quad c_{2,3} \\ \uparrow \quad \uparrow \\ \text{cobweb} \end{array} + \begin{array}{c} c_{1,3} \quad c_{2,3} \\ \uparrow \quad \uparrow \\ \text{cobweb} \end{array} \\
&= \left(\begin{array}{c} c_{1,2} \\ \text{cobweb} \end{array} + \begin{array}{c} c_{1,2} \\ \text{cobweb} \end{array} \right) \begin{array}{c} c_{1,3} \quad c_{2,3} \\ \uparrow \quad \uparrow \end{array} \\
&= [2] \begin{array}{c} c_{1,3} \quad c_{2,3} \\ \uparrow \quad \uparrow \end{array}
\end{aligned}$$

where the last line is exactly $\mathcal{F}(D_R)$, as desired.

Lemma 4.11: $\mathcal{F}(D_L) = \mathcal{F}(D_R)$ for any choice in valid boundary data.

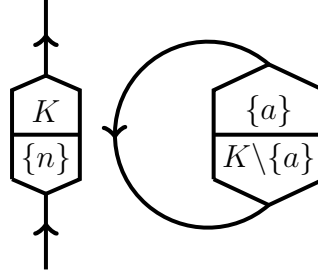
Proof. Fix $n \geq 3$ and consider the digon with $k = n - 1$. We may assume that the boundary data assigns the set $K = \{1, \dots, n - 1\}$ to all boundary strands. For this boundary data, a state corresponds to a choice of $a \in K$ which results in $n - 1$ cobwebs.

Consider the cobweb $\text{Cob}(D_L, a)$. For each $i \in K$ and $j \notin K$, there is a vertical strand labeled $c_{i,j}$ from the bottom to the top of the cobweb. For each $i \in K \setminus \{a\}$, there are strands labeled $c_{i,a}$ and $c_{a,i}$. It follows that

$$\text{Cob}(D_L, a) = \begin{array}{c} \begin{array}{|c|} \hline K \setminus \{a\} \\ \hline \{n\} \\ \hline \end{array} \end{array} \begin{array}{c} \begin{array}{|c|} \hline K \setminus \{a\} \\ \hline \{a\} \\ \hline \end{array} \end{array} \begin{array}{c} \begin{array}{|c|} \hline \{a\} \\ \hline K \setminus \{a\} \\ \hline \end{array} \end{array}$$

recalling that strands of distinct colors commute which allows us to group together the vertical strands.

Applying tag cancellation, each of the non-vertical strands becomes a positively-oriented bubble with label $c_{i,a}$. This leaves us with the diagram



which can be rewritten using the "bubble-shorthand" as

$$\text{Cob}(D_L, a) = \prod_{\substack{i \in K \\ i \neq a}} (i \ a) \left(\begin{array}{c} \begin{array}{|c|} \hline K \\ \hline \end{array} \end{array} \right)$$

Notice that the vertical strands are the same for each state a and thus can

$$\mathcal{F}(D_L) = \left(\sum_{a \in K} \prod_{\substack{i \in K \\ i \neq a}} (i \ a) \right) \left(\begin{array}{c} | \\ \text{\tiny $\blacktriangleright$} \\ \text{\scriptsize K} \\ \text{\tiny $\blacktriangleleft$} \\ | \end{array} \right) \quad (4.32)$$

$$= [k] \left(\begin{array}{c} \text{---} \uparrow \text{---} \\ | \\ \text{---} \downarrow \text{---} \end{array} \right) \quad (4.33)$$

Since there is only one state for D_R ,

$$\mathcal{F}(D_R) = \text{---} \begin{array}{c} \uparrow \\ \boxed{\text{K}} \\ \downarrow \end{array} \text{---}$$

4.4.3 Bursting a square

$$\begin{array}{c} \text{Diagram 1: A circle with four external lines. Top-left: incoming line labeled } k, \text{ top-right: outgoing line labeled } 1, \text{ bottom-left: incoming line labeled } k, \text{ bottom-right: outgoing line labeled } 1. \end{array} = \begin{array}{c} \text{Diagram 2: A vertical line with two vertices. Top vertex: incoming line labeled } k, \text{ outgoing line labeled } 1. \text{ Bottom vertex: incoming line labeled } k, \text{ outgoing line labeled } 1. \end{array} + [k-1] \begin{array}{c} \text{Diagram 3: A vertical line with two vertices. Top vertex: incoming line labeled } k, \text{ outgoing line labeled } k. \text{ Bottom vertex: incoming line labeled } k, \text{ outgoing line labeled } k. \end{array} \begin{array}{c} \text{Diagram 4: A vertical line with two vertices. Top vertex: incoming line labeled } 1, \text{ outgoing line labeled } 1. \text{ Bottom vertex: incoming line labeled } 1, \text{ outgoing line labeled } 1. \end{array} \quad (4.34)$$

Abbreviate the diagrams with D_1, D_2 , and D_3 so that Equation 4.34 becomes

$$D_1 = D_2 + [k - 1]D_3.$$

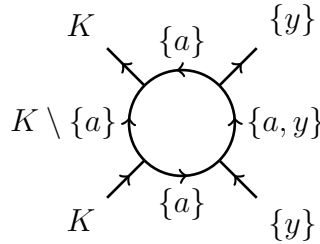
We wish to prove

Lemma 4.12: $\mathcal{F}(D_1) = \mathcal{F}(D_2 + [k - 1]D_3)$ for any choice in valid boundary data.

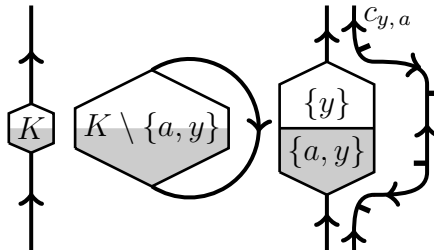
There are three cases to consider when assigning sets to each of the four boundary strands. We shall handle the proof of Lemma 4.12 for each case separately.

Case 1: Let K have cardinality k and let $y \in K$. Assign K to the top left and bottom left strands of each web and $\{y\}$ to the top right and bottom right strands.

Proof. A state on D_1 corresponds to a choice $a \in K$. With this state, assign $K \setminus \{a\}$ to the left side of the square, $\{a, y\}$ to the right side of the square, and $\{a\}$ to the top and bottom of the square:



Let $K' = K \setminus \{a, y\}$. The strands in the associated cobweb are:


$$\sum_{a \in K \setminus \{y\}} \prod_{i \in K \setminus \{a, y\}} (i \ a) = [k-1].$$

Then,

$$\mathcal{F}(D_1) = \sum_a \text{Cob}(D_1, a) = [k-1]$$

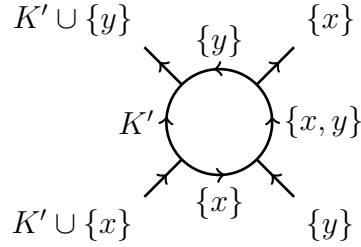
On the right side of the equation, the first term has no states that are consistent with the boundary data since $y \in K$ by assumption. The second term has one possible state – the resulting web agrees with $\mathcal{F}(D_1)$ as desired.

□

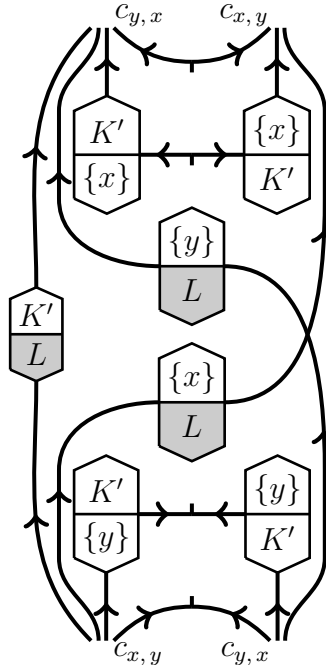
Case 2: Let K' have cardinality $k - 1$ and let $x, y \notin K'$. Assign $K' \cup \{x\}$

and $\{y\}$ to the bottom left and bottom right strands of each web. Assign $K' \cup \{y\}$ and $\{x\}$ to the top left and top right strands of each web.

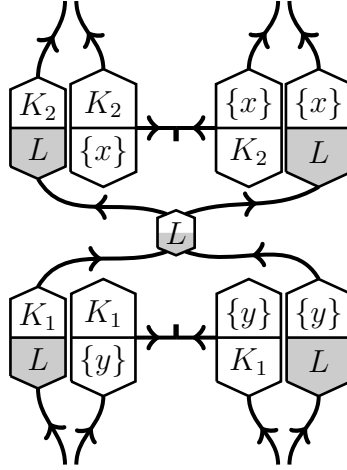
Proof. There is only one state on D_1 that is consistent with this boundary data. Assign K' to the left side of the square and $\{x, y\}$ to the right. Assign $\{x\}$ to the bottom and $\{y\}$ to the top of the square.



Let $K_1 = K' \cup \{x\}$, $K_2 = K' \cup \{y\}$, and $L = K' \sqcup \{x, y\}$. The strands are then:



There is only one consistent state for D_2 – this state assigns the set $K' \cup \{x, y\}$ to the vertical strand. The strands are as follows:



There are no consistent states for D_3 . Up to reordering strands of distinct colors, the cobwebs for D_1 and D_2 are equivalent as desired. $\square$

Case 3: Let K have cardinality k and $y \notin K$. Assign K to the top left and bottom left strands of all webs and $\{y\}$ to the top right and bottom right strands.

Proof. A state on D_1 corresponds to a choice of $a \in K$. For each state, assign $\{a\}$ to the top and bottom strands of the square, and assign $K \setminus \{a\}$ and $\{a, y\}$ to the left and right sides of the square.

Define $K_1 = K \setminus \{a\}$ and $K_2 = \{a, y\}$. Then, strands in $\text{Cob}(D_1, a)$ are as follows:

$\text{Cob}(D_1, a) =$

(4.35)

$=$

(4.36)

$$\begin{aligned}
 &= \text{Diagram (4.37)} \\
 &\quad \text{Diagram (4.37) shows a sequence of hexagonal cells and strands. The first cell is a circle with a clockwise arrow, containing a hexagon with top half $\{a\}$ and bottom half $K \setminus \{a\}$. This is followed by a vertical strand with an upward arrow. Then, a hexagon with top half $K \setminus \{a\}$ and bottom half K . This is followed by a vertical strand with an upward arrow. Then, a hexagon with top half $\{a\}$ and bottom half $K \cup \{y\}$. This is followed by a vertical strand with an upward arrow. Then, a hexagon with top half $\{y\}$ and bottom half $\{a, y\}$. This is followed by a vertical strand with an upward arrow. Two curved strands, labeled $c_{a,y}$ and $c_{y,a}$, connect the top and bottom of the first two hexagons. The strand $c_{a,y}$ goes from the top of the first hexagon to the top of the second, and $c_{y,a}$ goes from the bottom of the first hexagon to the bottom of the second. The strands $c_{a,y}$ and $c_{y,a}$ are highlighted in pink. The label (4.37) is on the right.
 \end{aligned}$$

$$\begin{aligned}
 &= \text{Diagram (4.38)} \\
 &\quad \text{Diagram (4.38) shows a sequence of hexagonal cells and strands. The first cell is a circle with a clockwise arrow, containing a hexagon with top half $\{a\}$ and bottom half $K \setminus \{a\}$. This is followed by a vertical strand with an upward arrow. Then, a hexagon with top half $K \setminus \{a\}$ and bottom half K . This is followed by a vertical strand with an upward arrow. Then, a hexagon with top half $\{a\}$ and bottom half $K \cup \{y\}$. This is followed by a vertical strand with an upward arrow. Then, a hexagon with top half $\{y\}$ and bottom half $\{a, y\}$. This is followed by a vertical strand with an upward arrow. Two curved strands, labeled $c_{a,y}$ and $c_{y,a}$, connect the top and bottom of the first two hexagons. The strand $c_{a,y}$ goes from the top of the first hexagon to the top of the second, and $c_{y,a}$ goes from the bottom of the first hexagon to the bottom of the second. The strands $c_{a,y}$ and $c_{y,a}$ are highlighted in pink. The label (4.38) is on the right.
 \end{aligned}$$

$$\begin{aligned}
 &= \text{Diagram (4.39)} \\
 &\quad \text{Diagram (4.39) shows a sequence of hexagonal cells and strands. The first cell is a circle with a clockwise arrow, containing a hexagon with top half $\{a\}$ and bottom half $K \setminus \{a\}$. This is followed by a vertical strand with an upward arrow. Then, a hexagon with top half K and bottom half $\{y\}$. This is followed by a vertical strand with an upward arrow. Then, a hexagon with top half $\{y\}$ and bottom half $\{a, y\}$. This is followed by a vertical strand with an upward arrow. A curved strand, labeled $c_{y,a}$, connects the top and bottom of the first hexagon. The strand $c_{y,a}$ is highlighted in pink. The label (4.39) is on the right.
 \end{aligned}$$

where (4.36) follows from tag cancellation, (4.37) and (4.39) follow from the fact that strands of distinct colors commute, and (4.38) follows from the pop

switch relation.

Observe that the vertical strands in (4.39) are independent from the choice in a . Therefore, the vertical strands can be factored out of the sum $\sum_a \text{Cob}(D_1, a)$:

$$\sum_{a \in K} \text{Cob}(D_1, a) = \begin{array}{c} \text{---} \nearrow \text{---} \\ \text{---} \nwarrow \text{---} \end{array} \begin{array}{c} \text{---} \nearrow \text{---} \\ \text{---} \nwarrow \text{---} \end{array} \sum_{a \in K} \left(\prod_{j \in K \setminus \{a\}} (j \ a) \right) (a \ y) \quad (4.40)$$

Let $a \in K$. The corresponding term in Equation 4.40 is a product of bubbles and can be written

$$\left(\prod_{j \in K \setminus \{a\}} (j \ a) \right) (a \ y) = [2] \left(\prod_{j \in K \setminus \{a\}} (j \ a) \right) - (y \ a) \left(\prod_{j \in K \setminus \{a\}} (j \ a) \right)$$

after applying $\text{Rel}([2], \{a, y\})$. Then,

$$\sum_{a \in K} \left(\prod_{j \in K \setminus \{a\}} (j \ a) \right) (a \ y) = [2][k] - [k+1] + \prod_{j \in K} (j \ y) \quad (4.41)$$

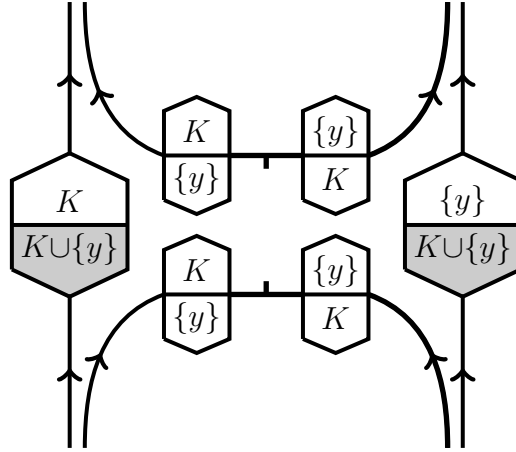
$$= [k-1] + \prod_{j \in K} (j \ y) \quad (4.42)$$

where (4.41) follows from $\text{Rel}([k], K)$ and $\text{Rel}([k+1], K \cup \{y\})$. Therefore,

$$\sum_{a \in K} \text{Cob}(D_1, a) = \left([k-1] + \prod_{j \in K} (j \ y) \right) \begin{array}{c} \text{---} \nearrow \text{---} \\ \text{---} \nwarrow \text{---} \end{array} \begin{array}{c} \text{---} \nearrow \text{---} \\ \text{---} \nwarrow \text{---} \end{array} \quad (4.43)$$

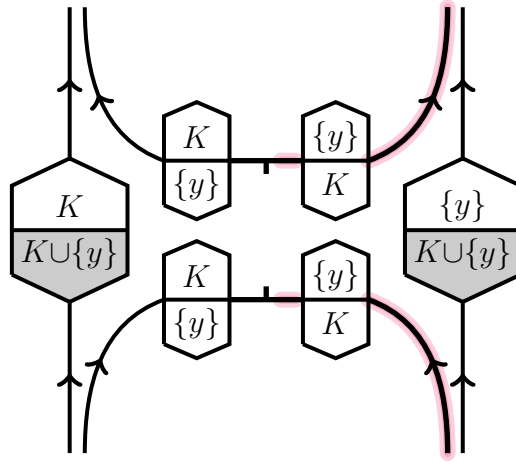
Next we compute the cobweb for D_2 . There is only one state consistent with the given boundary data – it assigns the middle strand the set $K \cup \{y\}$. Then,

$$\text{Cob}(D_2, K \cup \{y\}) =$$



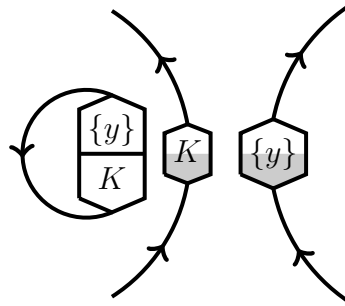
$$(4.44)$$

$$=$$



$$(4.45)$$

$$=$$



$$(4.46)$$

$$= \prod_{j \in K} (j \ y) \quad \begin{array}{c} \text{Diagram: Two hexagons side-by-side. The left hexagon is labeled } K \text{ and the right hexagon is labeled } \{y\}. \text{ Each hexagon has two strands passing through it, one from the top and one from the bottom, with arrows indicating a crossing.} \end{array} \quad (4.47)$$

where (4.46) is obtained from (4.45) by applying the pop switch relation and tag cancellation to the highlighted portion of the strands.

There is only one state on D_3 consistent with the given boundary data. The cobweb is

$$\begin{array}{c} \text{Diagram: Two hexagons side-by-side. The left hexagon is labeled } K \text{ and the right hexagon is labeled } \{y\}. \text{ Each hexagon has two strands passing through it, one from the top and one from the bottom, with arrows indicating a crossing.} \end{array} \quad (4.48)$$

Therefore,

$$\mathcal{F}(D_2 + [k-1]D_3) = \left(\prod_{j \in K} (j \ y) + [k-1] \right) \quad \begin{array}{c} \text{Diagram: Two hexagons side-by-side. The left hexagon is labeled } K \text{ and the right hexagon is labeled } \{y\}. \text{ Each hexagon has two strands passing through it, one from the top and one from the bottom, with arrows indicating a crossing.} \end{array} \quad (4.49)$$

which clearly agrees with $\mathcal{F}(D_1)$. □

Chapter 5

Evaluating Closed Webs

The main theorem in [8] implies that any closed web diagram in the SL_n spider is equivalent to a scalar multiple of the empty diagram. Here “scalar” means an element of the Laurent ring $\mathbb{Z}[q, q^{-1}]$. If a web W is equivalent to p times the empty diagram, it’s convenient to say that W evaluates to p , or W is p . Without loss of generality, we may assume that W is connected. If not, the connected components of W could be reduced independently and the whole web would be the product of the components.

Throughout this section, assume that W is a closed web. W can be reduced by successively applying the spider relations. The main result of this chapter is an algorithm that reduces W , the details of which are discussed in [Section 5.6](#).

Conjecture 5.1: *Any closed web W in the SL_n spider simplifies to $g \in \mathbb{Z}[q, q^{-1}]$ where g is a unimodal symmetric polynomial.*

The main theorem in [8] proves that W simplifies to some $g \in \mathbb{Z}[q, q^{-1}]$. To prove [Conjecture 5.1](#), it suffices to prove that g is a unimodal symmetric polynomial. The evaluation algorithm provides the computational framework

necessary to make further progress on [Conjecture 5.1](#).

5.1 Unimodal symmetric polynomials

Recall that to prove [Conjecture 5.1](#) it suffices to prove that the resulting polynomial in $\mathbb{Z}[q, q^{-1}]$ is unimodal and symmetric (defined below). The purpose of this section is to prove that quantum integers, factorials, and binomials are all unimodal symmetric polynomials. In [Section 5.2](#), these results are used to prove [Conjecture 5.1](#) on webs with a small number of vertices. In addition, the results are used to describe a collection of web relations that preserve the unimodal and symmetric properties of a web. That is, the relations can be applied one at a time where at each step, the coefficients are always unimodal and symmetric.

Definition 5.2: A sequence $a_0, a_1, \dots, a_n$ of real numbers is said to be *unimodal* if for some $0 \leq j \leq n$ we have $a_0 \leq a_1 \leq \dots \leq a_j$ and $a_j \geq a_{j+1} \geq \dots \geq a_n$.

Definition 5.3: The sequence $a_0, a_1, \dots, a_n$ is *symmetric* if $a_i = a_{n-i}$ for $0 \leq i \leq n$.

Let $\mathcal{P} \subseteq \mathbb{Z}[q, q^{-1}]$ denote the collection of polynomials that are unimodal and symmetric with respect to the degree of q .

Example 5.4: Consider the quantum integer $[n] = q^{n-1} + q^{n-3} + \dots + q^{-n+3} + q^{-n+1}$. The sequence of coefficients is both unimodal and symmetric. Therefore, $[n] \in \mathcal{P}$.

Definition 5.5: The q -analog, or q -number, of an integer n is

$$\{n\}_q = \frac{1 - q^n}{1 - q} = 1 + q + q^2 + \cdots + q^{n-1}.$$

The q -factorial is

$$\{n\}_q! = \{n\}_q \{n-1\}_q \cdots \{1\}_q$$

and the q -binomial is

$$\left\{ \begin{matrix} n \\ k \end{matrix} \right\}_q = \frac{\{n\}_q!}{\{k\}_q! \{n-k\}_q!}.$$

When q is clear from context, the subscript may be omitted. It is often advantageous to rewrite a quantum integer in terms of a q -number, motivating the following lemmas.

Lemma 5.6:

$$[n]_q = q^{-n+1} \{n\}_{q^2}$$

Proof. Observe that

$$\{n\}_{q^2} = 1 + q^2 + q^4 + \cdots + q^{2(n-1)}.$$

Then,

$$q^{-n+1} \{n\}_{q^2} = q^{-n+1} + q^{-n+3} + \cdots + q^{n-1} = [n]_q$$

as desired. □

Lemma 5.7:

$$[n]_q! = q^{-n(n-1)/2} \{n\}_{q^2}!$$

Lemma 5.8:

$$\left[\begin{matrix} n \\ k \end{matrix} \right]_q = q^{k^2 - nk} \left\{ \begin{matrix} n \\ k \end{matrix} \right\}_{q^2}$$

Proposition 5.9 ([22]): *Let $A(q), B(q) \in \mathcal{P}$ have nonnegative coefficients. Then, $A(q)B(q) \in \mathcal{P}$.*

It follows directly from Proposition 5.9 that the product of quantum integers is a symmetric unimodal polynomial in q .

Theorem 5.10 ([22]): *The q -binomial coefficient $\left\{ \begin{smallmatrix} n \\ k \end{smallmatrix} \right\}$ is symmetric and unimodal.*

Then, $\left[\begin{smallmatrix} n \\ k \end{smallmatrix} \right]$ is symmetric and unimodal as both q^{k^2-nk} and $\left\{ \begin{smallmatrix} n \\ k \end{smallmatrix} \right\}_{q^2}$ are symmetric unimodal polynomials.

5.2 Simplifying small webs

Before describing the evaluation algorithm, we discuss various simplification moves that reduce the number of vertices present in a web. Throughout this section, let W be a closed connected web in the SL_n spider with v vertices.

Lemma 5.11: *W has an even number of vertices.*

Proof. Replace each bivalent vertex in W with the corresponding trivalent vertex so that every vertex in W is degree three. Viewing W as a graph with V vertices and E edges,

$$2E = \sum_{v_i} \deg(v_i) = 3V.$$

proving that V is even. □

In fact, the number of trivalent and the number of bivalent vertices are both even. If T and B are the number of trivalent and bivalent respectively, then the Fundamental Theorem of Graph Theory gives

$$2E = 2B + 3T$$

implying that T is even. Since $V = B + T$ is even, B must be even as well. Using tag migration and tag cancellation, remove each pair of bivalent vertices until only the trivalent vertices remain. We may therefore assume that W has only trivalent vertices.

If W has a vertex of the form

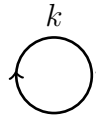


then the leftmost edge has label zero and can therefore be deleted. The remaining loop can be reduced via [Lemma 5.12](#).

There are at most three edges connecting any pair of vertices in W .

Lemma 5.12: *A closed web without vertices is in $\mathcal{P}$.*

Proof. If W has $v = 0$ vertices, then W is either the empty diagram or is a bubble



In the case that W is a bubble, Equation 3.9 implies

$$W = \begin{bmatrix} n \\ k \end{bmatrix}_q.$$

In either case, Theorem 5.10 gives the desired result. $\square$

Lemma 5.13: *The theta web is in $\mathcal{P}$.*

Proof. Let v_1, v_2 be vertices in W connected by three edges. Since W is connected and every vertex is degree three, it follows that

$$W = \begin{array}{c} \begin{array}{c} \text{ } \end{array} \\ \begin{array}{c} \text{ } \end{array} \\ \begin{array}{c} \text{ } \end{array} \end{array}$$

After bursting the top digon via Equation 3.2, we have

$$W = \begin{bmatrix} k+l \\ l \end{bmatrix} \begin{array}{c} \text{ } \end{array}$$

By Lemma 5.12 and Theorem 5.10, W is a symmetric unimodal polynomial. $\square$

Remark 5.14. Lemma 5.12 and Lemma 5.13 prove Conjecture 5.1 for webs with at most two vertices.

5.3 Removing small faces

A *small face* in a web is a face that is bounded by two or three edges. This section describes how the small faces in a web can be iteratively removed.

The resulting web has strictly fewer vertices than the original web and its coefficient is in $\mathcal{P}$. It will be convenient to understand a web in terms of its “flow”. Every vertex in W is assigned a flow, either positive or negative, depending on whether the vertex is a split or a merge. Recall that every vertex in W is degree three as the bivalent vertices occur in pairs, which can then be removed via tag migration and tag cancellation.

Definition 5.15: The *flow* of a vertex v_i in W is

$$f(v_i) = \begin{cases} 1 & v_i \text{ is a merge} \\ -1 & v_i \text{ is a split} \end{cases}.$$

If W' is a web contained in W , then the flow in W' is $f(W') = \sum_{v_i \in W'} f(v_i)$.

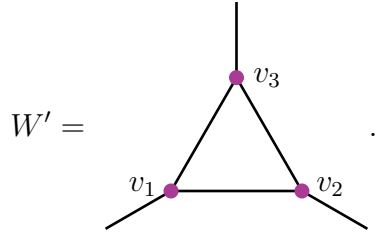
The *global flow* of W is $f(W) = \sum f(v_i)$.

Lemma 5.16: Any 2-face in W can be removed at the expense of a factor of an element of $\mathcal{P}$.

Proof. Apply Equation 3.2 or Equation 3.3 to W to remove the 2-face, multiplying the resulting web by some quantum binomial. By Lemma ??, the result holds. $\square$

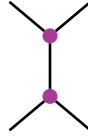
Lemma 5.17: The flow on any 3-face W' in W is $f(W') = \pm 1$.

Proof. Let $W' \subseteq W$ be the web



If $f(W') = 3$, then all three vertices are merge type. By inspection, this is impossible. Likewise, $f(W') \neq -3$. Therefore, $f(W') = \pm 1$. Without loss of generality, suppose that $f(v_1) = f(v_2) = 1$ and $f(v_3) = -1$. $\square$

Definition 5.18: Let $W' \subseteq W$ be a web of the form

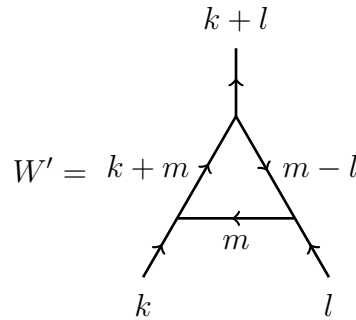


If $f(W') = \pm 2$, then W is $I=H$ compatible.

Remark 5.19. The terminology “ $I=H$ compatible” is meant to emphasize that Equation 3.4 can be applied to the web. Whenever a web contains a pair of adjacent vertices with the same flow, the $I=H$ relation can be applied. This property is used in the following lemma, to remove any 3-face in a web.

Lemma 5.20: Any 3-face in W can be replaced with a trivalent vertex, multiplied by an element of $\mathcal{P}$.

Proof. Let $W' \subseteq W$ be a 3-face. Without loss of generality, assume $f(W') = 1$. Up to mirror-reflections, arrow reversal, and rotation, there is only one valid orientation for W' :



Notice that W' is $I=H$ compatible. Therefore,

$$\begin{array}{c} k+l \\ \uparrow \\ \begin{array}{ccc} k+m & & m-l \\ \swarrow & \text{---} & \searrow \\ k & m & l \end{array} \end{array} = \begin{array}{c} k+l \\ \uparrow \\ \begin{array}{ccc} k+m & & m-l \\ \swarrow & \text{---} & \searrow \\ k & k+l & l \end{array} \end{array} = \begin{bmatrix} n-k-l \\ m-l \end{bmatrix}_q \begin{array}{c} k+l \\ \uparrow \\ \begin{array}{cc} k & l \end{array} \end{array}$$

where the last equality uses [Equation 3.3](#). Since any quantum binomial is a symmetric unimodal polynomial, the proof is complete. $\square$

Let W' be the web obtained by removing all small faces in W . The coefficient of W' is a product of quantum binomials and is therefore in $\mathcal{P}$. Therefore, we have proven [Conjecture 5.1](#) for any W where all faces are degree two or three. Without loss of generality, let W be a closed connected web with no bivalent vertices and every face of degree at least four. If there were a face of degree two or three, it could be removed via [Lemma 5.16](#) or [Lemma 5.20](#), reducing the total number of vertices in W by two while preserving the symmetric and unimodal properties.

5.4 A collection of valid moves

Working towards a proof of [Conjecture 5.1](#), we describe a collection of “valid moves” in the SL_n spider that preserve the unimodal and symmetric properties. Precisely, a valid move is a relation in the SL_n spider that replaces existing strands and vertices without introducing any new coefficients or terms,

or replaces part of a web and multiplies by a polynomial in $\mathcal{P}$. Further, a valid move cannot increase the number of vertices present in a web. This requirement ensures that any algorithm defined on valid moves is getting closer to the empty diagram.

Proposition 5.21: *The following moves preserve the symmetric unimodal property:*

- (a) *Switching a tag*
- (b) *Cancelling tags*
- (c) *Popping a bubble*
- (d) *Removing a digon*
- (e) *Removing a 3-face*
- (f) $I=H$

Proof. Each of (a), (b), or (f) clearly meet the criterion of a valid move. (c) is a valid move via Lemma 5.12, (d) is a valid move via Lemma 5.16, and (e) is a valid move via Lemma 5.20. $\square$

5.5 Morsifying W

The goal of this section is to rewrite a closed web so that each horizontal slice contains a single elementary web, possibly with identity morphisms on either side.

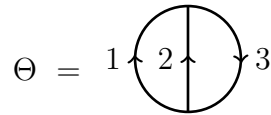
We follow the notation introduced in [10]. Let W be a web. A strand labeled k is now labeled by the corresponding minuscule weight, ω_k . Since the fundamental weights of SL_n are all minuscule, this does not change any of the web properties. Pick an isotopy representative for W such that the height function is a Morse function and the boundary of W is at the top level. There is a sequence of horizontal lines $\ell_0, \dots, \ell_m$ that slices W into its elementary pieces. That is, between each ℓ_{k-1}, ℓ_k there is a single cup, cap, or vertex, possibly tensored with an identity morphism. Let W_k be the portion of W between ℓ_{k-1} and ℓ_k . Each W_i is a morphism and $W = W_m \circ \dots \circ W_1$.



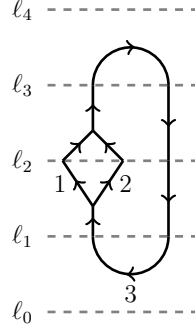
Figure 5.1: Elementary web pieces (cup, cap, merge, and split).

For each horizontal line ℓ_k , let $\lambda^{(k)}$ be the vector labels of the edges that intersect ℓ_k . If an edge with label ω_k is pointing downwards, replace its label with $\omega_k^* = \omega_{n-k}$. Observe that if W is a closed web, then W_1 contains a cup and W_m contains a cap. In particular, $\lambda^{(0)}, \lambda^{(m)} = \emptyset$.

Example 5.22: Consider the closed “theta” web in the SL_4 spider:



Recall that webs that are equivalent under planar isotopy are equal in the spider. Therefore, Θ is equivalent to this web:



The vector labels for each of the horizontal lines are $\vec{\lambda}^{(0)} = \emptyset$, $\vec{\lambda}^{(1)} = \{\omega_3, \omega_1\}$, $\vec{\lambda}^{(2)} = \{\omega_1, \omega_2, \omega_1\}$, $\vec{\lambda}^{(3)} = \{\omega_3, \omega_1\}$, and $\vec{\lambda}^{(4)} = \emptyset$. Then, $\Theta = W_4 \circ W_3 \circ W_2 \circ W_1$ where W_1 is a cup, W_2 is a split, W_3 is a merge, and W_4 is a cap. Technically, W_2 and W_3 are tensor products of an elementary web type with an identity morphism, but this does not affect the computations discussed in the following sections. The key idea is to ensure that each horizontal slice contains exactly one elementary web, possibly tensored with identity morphisms on either side.

5.6 The Algorithm

Definition 5.23: Let $\mathcal{T}_n(2)$ be the collection of tuples containing the empty tuple and tuples of length at most 2 with entries in $\{1, \dots, n\}$. Define a strict partial order $\prec$ on $\mathcal{T}_n(2)$ as follows:

- (a) $() \prec (n)$
- (b) $(k + l) \prec (k, l)$

Definition 5.24: Let $\mathcal{C}$ be the (strict) monoidal category defined as follows:

The set of objects in $\mathcal{C}$ is $\mathcal{T}_n = \bigcup_{k=1}^{\infty} \mathcal{T}_n(k)$ where $\mathcal{T}_n(k)$ contains all tuples of length at most k . The monoidal product on objects is given by concatenation: if $S = (s_1, \dots, s_k)$ and $T = (t_1, \dots, t_m)$, then the monoidal product is

$$S \otimes T = (s_1, \dots, s_k, t_1, \dots, t_m)$$

The unit object is the empty tuple $()$. Notice that $() \otimes S = S = S \otimes ()$ for any object S .

Suppose that $k, l \in \{1, \dots, n-1\}$ and $k+l \leq n$. The generating morphisms in $\mathcal{C}$ are:

1. Split: $\mathcal{S}_{k,l} : (k+l) \rightarrow (k, l)$
2. Merge: $\mathcal{M}_{k,l} : (k, l) \rightarrow (k+l)$
3. Insertion: $\mathcal{I} : () \rightarrow (n)$
4. Deletion: $\mathcal{D} : (n) \rightarrow ()$

along with the identity morphisms $\iota_j : (j) \rightarrow (j)$. The monoidal product of two morphisms is obtained by applying each morphism to the corresponding entries of the resulting object. For example,

$$\mathcal{I} \otimes \mathcal{S}_{2,1} : (3) = () \otimes (3) \mapsto (n) \otimes (2, 1) = (n, 2, 1)$$

Remark 5.25. Let $S = (s_1, \dots, s_k, \dots, s_m)$ be an object in $\mathcal{C}$ and suppose that f is a generating morphism. Then, f can be applied to the k th position of S via the morphism

$$(\iota_{s_1} \otimes \dots \otimes \iota_{s_k}) \otimes f \otimes (\iota_{s_{k+1}} \otimes \dots \otimes \iota_{s_m}).$$

This notation quickly becomes cumbersome and thus we write $f(S)$ when the position at which f is applied is clear from context. If the generating morphisms are applied at position j of $T = (t_1, \dots, t_m)$, this means: The split morphism applied to position j of T yields

$$\mathcal{S}_{k,l}^{(j)} : T \mapsto (t_1, \dots, t_{j-1}, k, l, t_{j+1}, \dots, t_m)$$

The merge morphism applied to position j of T yields

$$\mathcal{M}_{k,l}^{(j)} : T \mapsto (t_1, \dots, t_{j-1}, t_j + t_{j+1}, t_{j+2}, \dots, t_m)$$

assuming that $t_j = k$ and $t_{j+1} = l$. The insertion and deletion morphisms applied to position j of T are

$$\mathcal{I}^{(j)} : T \mapsto (t_1, \dots, t_{j-1}, n, t_j, \dots, t_m)$$

and

$$\mathcal{D}^{(j)} : T \mapsto (t_1, \dots, t_{j-1}, t_{j+1}, \dots, t_m)$$

respectively (assuming that $t_j = n$ for deletion).

For instance, if $\mathcal{I}$ is applied to the 4th position of $T = (1, 3, 5, 2, 1)$, we write $\mathcal{I}^{(4)}(T)$ in place of $(\iota_{(1,3,5)} \otimes \mathcal{I} \otimes \iota_{(2,1)})(T)$. Then,

$$\mathcal{I}^{(4)}(T) = (\iota_{(1,3,5)} \otimes \mathcal{I} \otimes \iota_{(2,1)})(T) = (1, 3, 5) \otimes (n) \otimes (2, 1) = (1, 3, 5, n, 2, 1).$$

Definition 5.26: Let $\mathcal{C}$ be as defined previously. Define a partial order $\preceq$ on the objects in $\mathcal{C}$: $R \preceq T$ if and only if there exists a morphism from R to T that is generated only by the split, insertion, and identity morphisms.

Example 5.27: Let $n = 4$. Set $R = (1, 2, 4, 1)$ and $T = (3, 1)$. Observe that

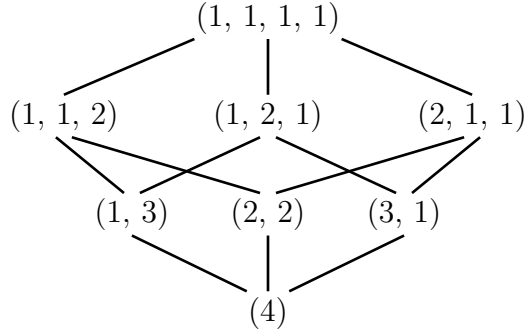
$$(\mathcal{S}_{1,2} \otimes \mathcal{I} \otimes \iota_{(1)})(T) = \mathcal{S}_{1,2}((3)) \otimes \mathcal{I}(()) \otimes \iota((1)) \quad (5.1)$$

$$= (1, 2, 4, 1) \quad (5.2)$$

$$= R \quad (5.3)$$

and thus $T \preceq R$.

Example 5.28: Let $n = 4$. Consider the collection of objects $R \in \mathcal{C}$ where $(4) \preceq R$ via a morphism built only from split and identity morphisms. The Hasse diagram for these objects is



Before describing the evaluation algorithm, we define a functor $\mathcal{G}$ from $\mathcal{S}p_n$ to $\mathcal{C}$. We may assume that every web in $\mathcal{S}p_n$ is written in Morse position. In $\mathcal{S}p_n$, a strand pointing downward with label k corresponds to the object $\{k^-\}$. The functor Γ_n described in [8] sends $\{k^-\}$ to $(\bigwedge^k \mathbb{C}_q^n)^*$. In the representation category, $(\bigwedge^k \mathbb{C}_q^n)^* \cong \bigwedge^{n-k} \mathbb{C}_q^n$ meaning that $\{k^-\}$ can be “replaced” with the object $\{n - k\}$ in $\mathcal{S}p_n$ assuming that all strands in the resulting web connect properly.

On objects, $\mathcal{G}$ sends $(k_1^{\epsilon_1}, \dots, k_m^{\epsilon_m})$ to $(t_1, \dots, t_m)$ where

$$t_i = \begin{cases} k_i & \epsilon_i > 0 \\ n - k_i & \epsilon_i < 0 \end{cases}$$

On morphisms, $\mathcal{G}$ is generated by the following actions on elementary webs:

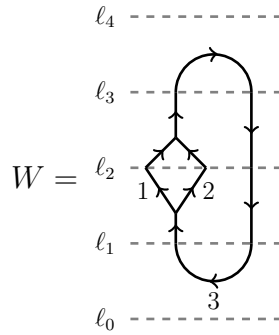
$$\begin{array}{c} k \quad l \\ \swarrow \quad \searrow \\ \uparrow \\ k+l \end{array} \mapsto \mathcal{S}_{k,l} : (k+l) \rightarrow (k, l) \quad (5.4)$$

$$\begin{array}{c} k+l \\ \uparrow \\ \swarrow \quad \searrow \\ k \quad l \end{array} \mapsto \mathcal{M}_{k,l} : (k, l) \rightarrow (k+l) \quad (5.5)$$

$$\begin{array}{c} \frown \\ k \end{array} \mapsto \mathcal{S}_{k,n-k} \circ I : () \rightarrow (k, n-k) \quad (5.6)$$

$$\begin{array}{c} k \\ \smile \end{array} \mapsto D \circ \mathcal{M}_{k,n-k} : (k, n-k) \rightarrow () \quad (5.7)$$

Example 5.29: Consider the following web in the SL_4 spider:



Each ℓ_i is an object in $\mathcal{S}p_4$:

- ℓ_0 is $()$
- ℓ_1 is $(3^+, 3^-)$

- ℓ_2 is $(1^+, 2^+, 3^-)$
- ℓ_3 is $(3^+, 3^-)$
- ℓ_4 is $()$

The corresponding objects in $\mathcal{C}$ are:

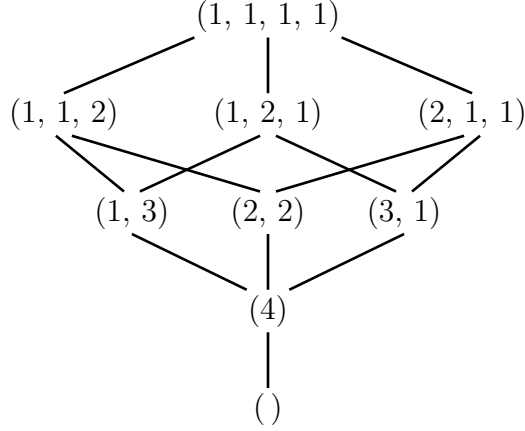
- $\mathcal{G}(\ell_0) = ()$
- $\mathcal{G}(\ell_1) = (3, 1)$
- $\mathcal{G}(\ell_2) = (1, 2, 1)$
- $\mathcal{G}(\ell_3) = (3, 1)$
- $\mathcal{G}(\ell_4) = ()$

The slice between ℓ_{i-1} and ℓ_i contains exactly one elementary web. Let W_i be the slice between ℓ_{i-1} and ℓ_i . Then, $\mathcal{G}$ acts on the slices W_i as follows:

- $\mathcal{G}(W_1) = \mathcal{S}_{3,1} \circ \mathcal{I} : () \rightarrow (3, 1)$
- $\mathcal{G}(W_2) = \mathcal{S}_{1,2} \otimes \iota_{(1)} : (3, 1) \rightarrow (1, 2, 1)$
- $\mathcal{G}(W_3) = \mathcal{M}_{1,2} \otimes \iota_{(1)} : (1, 2, 1) \rightarrow (3, 1)$
- $\mathcal{G}(W_4) = D \circ \mathcal{M}_{3,1} : (3, 1) \rightarrow ()$

Upon inspection, $\mathcal{G}(W_4 \circ W_3 \circ W_2 \circ W_1) = \mathcal{G}(W_4) \circ \mathcal{G}(W_3) \circ \mathcal{G}(W_2) \circ \mathcal{G}(W_1)$.

Remark 5.30. Notice that each $\mathcal{G}(\ell_i)$ in the previous example appears as a node in the Hasse diagram



Furthermore, the nodes corresponding to $\mathcal{G}(\ell_{i-1})$ and $\mathcal{G}(\ell_i)$ have an edge connecting them in the Hasse diagram. If $\mathcal{G}(\ell_{i-1}) \preceq \mathcal{G}(\ell_i)$ then $\mathcal{G}(W_i)$ is a split or insertion. If $\mathcal{G}(\ell_i) \preceq \mathcal{G}(\ell_{i-1})$, then $\mathcal{G}(W_i)$ is a merge or deletion. The composition W is therefore a walk in the Hasse diagram, starting and ending at $()$.

We are now ready to describe the algorithm that reduces any closed web in the SL_n spider to a polynomial in $\mathbb{Z}[q, q^{-1}]$. Assume that W is a closed, connected web written in Morse position. Suppose that W is not connected. Then [Algorithm 1](#) is applied to each connected component separately. The value of W is obtained by multiplying the output of [Algorithm 1](#) on each of the connected components.

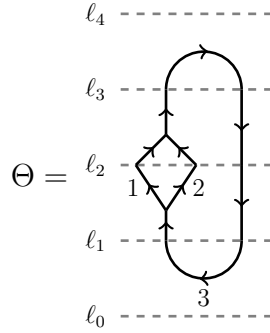
Let $\ell_0, \dots, \ell_m$ denote the horizontal lines that slice W into a composition $W_m \circ \dots \circ W_1$ of webs that contain exactly one split, merge, cup, or cap. Since W is closed, $\ell_0 = () = \ell_m$. Each W_i is a morphism from ℓ_{i-1} to ℓ_i in $\mathcal{S}p_n$. With respect to the partial order defined in [Definition 5.26](#), either $\mathcal{G}(\ell_{i-1}) \preceq \mathcal{G}(\ell_i)$ or

$\mathcal{G}(\ell_i) \preceq \mathcal{G}(\ell_{i-1})$. It is now clear that each slice in W falls into exactly one of the cases described in [Algorithm 1](#).

[Algorithm 1](#) takes $(\ell_0, \dots, \ell_m)$ as input and initializes $x = x_1 \wedge \dots \wedge x_n$. The algorithm iterates over the slices W_i . During each iteration of the loop, W_i is matched to one of the generating morphism types (split, merge, insertion, or deletion) and applied to x . The result is stored in x before returning to the beginning of the loop.

5.6.1 An example computation

Recall the morsified Θ web



and the corresponding morphism in $\mathcal{C}$ obtained via the functor $\mathcal{G}$. In the representation category, Θ is a morphism from $\mathbb{C}_q$ to $\mathbb{C}_q$. As $U_q(\mathfrak{sl}_4)$ representations, $\mathbb{C}_q \cong \wedge^4(V)$ which is generated by $x_{\{1,2,3,4\}} = x_1 \wedge x_2 \wedge x_3 \wedge x_4$. For brevity, we write x_{1234} instead of $x_{\{1,2,3,4\}}$. We apply [Algorithm 1](#) to prove

$$\Theta \equiv [4]_q [3]_q = q^{-5} + 2q^{-3} + 3q^{-1} + 3q + 2q^3 + q^5.$$

Algorithm 1: Evaluation of a closed web

Input: Sequence $(\ell_0, \dots, \ell_m)$ of label vectors in a closed web W **Output:** Image of $x_1 \wedge \dots \wedge x_n$ under W $x \leftarrow x_1 \wedge \dots \wedge x_n$ **for** i **in** $\text{range}(m)$: $j \leftarrow$ index where $\mathcal{G}(\ell_i)$ and $\mathcal{G}(\ell_{i+1})$ first differ **if** $\mathcal{G}(\ell_i) \preceq \mathcal{G}(\ell_{i+1})$: **if** $(k+l)$ replaced with (k, l) : $x \leftarrow \mathcal{S}^{(j)}(x)$ */* split morphism at position j */* **else:** $x \leftarrow \mathcal{I}^{(j)}(x)$ */* insertion morphism at position j */* **else:** **if** (k, l) replaced with $(k+l)$: $x \leftarrow \mathcal{M}^{(j)}(x)$ */* merge morphism at position j */* **else:** $x \leftarrow \mathcal{D}^{(j)}(x)$ */* deletion morphism at position j */***return** x

Equivalently, we must show

$$\Theta(x_{1234}) = (q^{-5} + 2q^{-3} + 3q^{-1} + 3q + 2q^3 + q^5)x_{1234}$$

using the isomorphism $\mathbb{C}_q \cong \wedge^4(V)$. We shall step through each slice of Θ to illustrate how the algorithm works.

The first slice is a cup, $W_1 = S_{3,1} : () \rightarrow (3, 1)$. Applying W_1 to x_{1234} yields

$$\begin{aligned} W_1(x_{1234}) = & -(x_{123} \otimes x_4) + q^{-1}(x_{124} \otimes x_3) \\ & - q^{-2}(x_{134} \otimes x_2) + q^{-3}(x_{234} \otimes x_1) \end{aligned} \quad (5.8)$$

The next slice is $W_2 = S_{1,2} \otimes \text{id} : (3, 1) \rightarrow (1, 2, 1)$. Since the generating morphisms are $\mathbb{C}_q$ -linear, we can apply them to [Equation 5.8](#) term-by-term:

$$\begin{aligned} W_2(-x_{123} \otimes x_4) = & -x_1 \otimes x_{23} \otimes x_4 + q^{-1}x_2 \otimes x_{13} \otimes x_4 \\ & - q^{-2}x_3 \otimes x_{12} \otimes x_4 \end{aligned} \quad (5.9)$$

$$\begin{aligned} W_2(q^{-1}x_{124} \otimes x_3) = & q^{-1}x_1 \otimes x_{24} \otimes x_3 - q^{-2}x_2 \otimes x_{14} \otimes x_3 \\ & + q^{-3}x_4 \otimes x_{12} \otimes x_3 \end{aligned} \quad (5.10)$$

$$\begin{aligned} W_2(-q^{-2}x_{134} \otimes x_2) = & -q^{-2}x_1 \otimes x_{34} \otimes x_2 + q^{-3}x_3 \otimes x_{14} \otimes x_2 \\ & - q^{-4}x_4 \otimes x_{13} \otimes x_2 \end{aligned} \quad (5.11)$$

$$\begin{aligned} W_2(q^{-3}x_{234} \otimes x_1) = & q^{-3}x_2 \otimes x_{34} \otimes x_1 - q^{-4}x_3 \otimes x_{24} \otimes x_1 \\ & + q^{-5}x_4 \otimes x_{23} \otimes x_1 \end{aligned} \quad (5.12)$$

Then,

$$W_3 \circ W_2 (-x_{123} \otimes x_4) = (-q^2 - 1 - q^{-2}) x_{123} \otimes x_4 \quad (5.13)$$

$$W_3 \circ W_2 (q^{-1} x_{124} \otimes x_3) = (q + q^{-1} + q^{-3}) x_{124} \otimes x_3 \quad (5.14)$$

$$W_3 \circ W_2 (-q^{-2} x_{134} \otimes x_2) = (-1 - q^{-2} - q^{-4}) x_{134} \otimes x_2 \quad (5.15)$$

$$W_3 \circ W_2 (q^{-3} x_{234} \otimes x_1) = (q^{-1} + q^{-3} + q^{-5}) x_{234} \otimes x_1 \quad (5.16)$$

Now,

$$W_4(x_{123} \otimes x_4) = -q^3 x_{1234} \quad (5.17)$$

$$W_4(x_{124} \otimes x_3) = q^2 x_{1234} \quad (5.18)$$

$$W_4(x_{134} \otimes x_2) = -q x_{1234} \quad (5.19)$$

$$W_4(x_{234} \otimes x_1) = x_{1234} \quad (5.20)$$

Combining these results, we see that W sends x_{1234} to $(q^{-5} + 2q^{-3} + 3q^{-1} + 3q + 2q^3 + q^5)x_{1234}$. Therefore, W equals $q^{-5} + 2q^{-3} + 3q^{-1} + 3q + 2q^3 + q^5$, a unimodal symmetric polynomial.

Chapter 6

Future directions

6.1 Extending the algorithm

At present, the implementation of [Algorithm 1](#) is a proof of concept and has not been optimized. Therefore,

Problem 6.1: Analyze the time complexity of [Algorithm 1](#) .

It would be interesting to compare this algorithm to an analogous version defined via cobwebs. [Remark 5.30](#) discusses a graph theoretic version of [Algorithm 1](#) . A closed web becomes a closed walk in a graph. If a web contains a digon, then the corresponding walk contains a segment of the form $A \rightarrow B \rightarrow A$. This begs the question, can the other spider relations be described via graph theory?

6.2 Big-data approach

The Kazhdan-Lusztig polynomials for the type A Coxeter group are studied in [16]. The authors take a “big-data” approach to develop various conjectures about the structure of Kazhdan-Lusztig polynomials.

Problem 6.2: Use techniques akin to those in [16] to develop conjectures about the value of a closed web.

6.3 A geometric perspective

The methods discussed thus far are algebraic and make little (if any) use of the underlying geometry. Recall that the fundamental representations of $\mathfrak{sl}_n$ are given by

$$V \cong \mathbb{C}^n, \Lambda^2(V), \dots, \Lambda^{n-1}(V).$$

These representations are Grassmannian varieties.

Definition 6.3: Let V be an n -dimensional vector space over $\mathbb{C}$. A Grassmannian $\mathrm{Gr}(k, V)$ is a differentiable manifold that parametrizes the set of k -dimensional linear subspaces of V . When V is clear from context, we write $\mathrm{Gr}(k, n)$.

In Section 5.5, a web is sliced into horizontal strips, each of which contains exactly one elementary web. Between each horizontal strip is a vector of labels. An entry k in the vector means that the slice contains a copy of the fundamental representation $\Lambda^k(\mathbb{C}^n)$. From the geometric point of view, the entry k in the

vector of labels corresponds to the Grassmannian $\text{Gr}(k, n)$. The Geometric Satake correspondence establishes a link between the representation category of $\mathfrak{sl}_n$ and the affine Grassmannian. This is an example of where the representation theory of $\mathfrak{sl}_n$ and $U_q(\mathfrak{sl}_n)$ differ – the original version of geometric Satake is incompatible with the quantum deformation used to define $U_q(\mathfrak{sl}_n)$.

An alternate version of geometric Satake for $U_q(\mathfrak{sl}_n)$ is conjectured in [10, Conjecture 6.3], and proved when $n = 2$ and $n = 3$. However, the machinery does not work for a general quantum parameter q . Cautis and Kamnitzer define a quantum K -theoretic version of geometric Satake for $U_q(\mathfrak{sl}_n)$ in a series of papers [3, 4, 5, 6, 7].

Problem 6.4: Make the connection between Algorithm 1 and the quantum K -theoretic version of geometric Satake for $U_q(\mathfrak{sl}_n)$ explicit.

Bibliography

- [1] Stephen Bigelow. A new approach to the SL_n spider, 2018. URL <https://arxiv.org/abs/1808.10575>.
- [2] Ken A. Brown and Ken R. Goodearl. *Lectures on algebraic quantum groups*. Advanced Courses in Mathematics. CRM Barcelona. Birkhäuser Verlag, Basel, 2002. ISBN 3-7643-6714-8. doi: 10.1007/978-3-0348-8205-7. URL <https://doi-org.proxy.library.ucsb.edu/10.1007/978-3-0348-8205-7>.
- [3] Sabin Cautis. Clasp technology to knot homology via the affine grassmannian, 2014. URL <https://arxiv.org/abs/1207.2074>.
- [4] Sabin Cautis and Joel Kamnitzer. Knot homology via derived categories of coherent sheaves i, $\mathfrak{sl}(2)$ case, 2007. URL <https://arxiv.org/abs/math/0701194>.
- [5] Sabin Cautis and Joel Kamnitzer. Knot homology via derived categories of coherent sheaves ii, $\mathfrak{sl}_m$ case. *Inventiones mathematicae*, 174(1):165–

- 232, July 2008. ISSN 1432-1297. doi: 10.1007/s00222-008-0138-6. URL <http://dx.doi.org/10.1007/s00222-008-0138-6>.
- [6] Sabin Cautis and Joel Kamnitzer. Knot homology via derived categories of coherent sheaves iv, coloured links, 2014. URL <https://arxiv.org/abs/1410.7156>.
- [7] Sabin Cautis and Joel Kamnitzer. Quantum k-theoretic geometric satake: the case. *Compositio Mathematica*, 154(2):275–327, October 2017. ISSN 1570-5846. doi: 10.1112/s0010437x17007564. URL <http://dx.doi.org/10.1112/S0010437X17007564>.
- [8] Sabin Cautis, Joel Kamnitzer, and Scott Morrison. Webs and quantum skew Howe duality. *Math. Ann.*, 360(1-2):351–390, 2014. ISSN 0025-5831,1432-1807. doi: 10.1007/s00208-013-0984-4. URL <https://doi.org/10.1007/s00208-013-0984-4>.
- [9] David E. Evans and Mathew Pugh. Ocneanu cells and Boltzmann weights for the $SU(3)$ ADE graphs. *Münster J. Math.*, 2:95–142, 2009. ISSN 1867-5778,1867-5786.
- [10] Bruce Fontaine, Joel Kamnitzer, and Greg Kuperberg. Buildings, spiders, and geometric Satake. *Compos. Math.*, 149(11):1871–1912, 2013. ISSN 0010-437X,1570-5846. doi: 10.1112/S0010437X13007136. URL <https://doi.org/10.1112/S0010437X13007136>.
- [11] William Fulton and Joe Harris. *Representation theory*, volume 129 of

- Graduate Texts in Mathematics*. Springer-Verlag, New York, 1991. ISBN 0-387-97527-6; 0-387-97495-4. doi: 10.1007/978-1-4612-0979-9. URL <https://doi-org.proxy.library.ucsb.edu/10.1007/978-1-4612-0979-9>. A first course, Readings in Mathematics.
- [12] Ellie Grano and Stephen Bigelow. The pop-switch planar algebra and the Jones-Wenzl idempotents. *J. Knot Theory Ramifications*, 24(6):1550032, 14, 2015. ISSN 0218-2165,1793-6527. doi: 10.1142/S0218216515500327. URL <https://doi.org/10.1142/S0218216515500327>.
- [13] Brian C. Hall. *Lie groups, Lie algebras, and representations*, volume 222 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2003. ISBN 0-387-40122-9. doi: 10.1007/978-0-387-21554-9. URL <https://doi-org.proxy.library.ucsb.edu/10.1007/978-0-387-21554-9>. An elementary introduction.
- [14] James E. Humphreys. *Introduction to Lie algebras and representation theory*, volume 9 of *Graduate Texts in Mathematics*. Springer-Verlag, New York-Berlin, 1978. ISBN 0-387-90053-5. Second printing, revised.
- [15] Greg Kuperberg. Spiders for rank 2 Lie algebras. *Comm. Math. Phys.*, 180(1):109–151, 1996. ISSN 0010-3616,1432-0916. URL <http://projecteuclid.org/euclid.cmp/1104287237>.
- [16] Abel Lacabanne, Daniel Tubbenhauer, and Pedro Vaz. Big data approach to kazhdan-lusztig polynomials. *Journal of Experimental Mathematics*, 2

- (1):21–62, March 2026. ISSN 3069-1125. doi: 10.56994/jxm.002.001.002.
URL <http://dx.doi.org/10.56994/JXM.002.001.002>.
- [17] Scott Morrison. A diagrammatic category for the representation theory of $U_q(sl_n)$, 2007. URL <https://arxiv.org/abs/0704.1503>.
- [18] Adrian Ocneanu. The classification of subgroups of quantum $SU(N)$. Lecture, January 2000. URL <https://cel.hal.science/cel-00374414>.
- [19] Paul A. Pearce and Jorgen Rasmussen. Ocneanu algebra of seams: Critical unitary e_6 rsos lattice model, 2025. URL <https://arxiv.org/abs/2409.06236>.
- [20] V. B. Petkova and J.-B. Zuber. The many faces of Ocneanu cells. *Nuclear Phys. B*, 603(3):449–496, 2001. ISSN 0550-3213,1873-1562. doi: 10.1016/S0550-3213(01)00096-7. URL [https://doi.org/10.1016/S0550-3213\(01\)00096-7](https://doi.org/10.1016/S0550-3213(01)00096-7).
- [21] Ph. Roche. Ocneanu cell calculus and integrable lattice models. *Comm. Math. Phys.*, 127(2):395–424, 1990. ISSN 0010-3616,1432-0916. URL <http://projecteuclid.org/euclid.cmp/1104180144>.
- [22] Richard P. Stanley. Log-concave and unimodal sequences in algebra, combinatorics, and geometry. *Annals of the New York Academy of Sciences*, 576(1):500–535, 1989. ISSN 0077-8923.