A new approach to the SL_n spider

Stephen Bigelow

August 30, 2018

Abstract

The SL_n spider gives a diagrammatic way to encode the representation category of the quantum group $U_q(\mathfrak{sl}_n)$. The aim of this paper is to define a new spider that contains the SL_n spider. The new spider is defined by generators and relations, according to fairly simple rules that start with combinatorial data coming from the root system of SL_n.

1 Introduction

A *spider* is an algebraic structure that can be used to encode the representation category of a quantum group. Kuperberg [6] gave generators and relations for spiders for rank two Lie algebras. Cautis, Kamnitzer, and Morrison [2] did the same for SL_n. The aim of this paper is to define a new spider that contains a copy of the SL_n spider.

The precise definition of the term "spider" is not all that important in this paper. We only need to define two specific spiders, and a map between them. We define a spider by defining its *webs*, and giving a list of relations these webs satisfy. A web will be some kind of graph drawn in a disk, with endpoints on the boundary of the disk. A relation will be a formal linear relation between webs. A relation can be applied "locally" inside a larger web, which remains unchanged outside the region where the relation is applied.

We will work over the field $\mathbb{Q}(q)$. The quantum integers are

$$[n] = q^{1-n} + q^{3-n} + \cdots + q^{n-3} + q^{n-1}.$$

We use $\mathbb{Q}(q)$ for the convenience of being able to divide by quantum integers, but it might be possible to prove our main result over the ring $\mathbb{Z}[q^{\pm 1}]$.

My motivation for the construction in this paper was to get a better understanding of spiders, which are currently presented by generators and relations that seem somewhat ad hoc. The relations for the new spider are simpler, and defined using the combinatorial data coming from the root system of SL_n. However, there is much that still remains mysterious.

I do not know how to prove that the new spider contains the SL_n spider without already knowing generators and relations for the SL_n spider. One

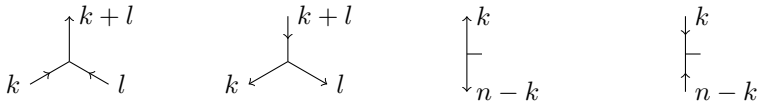


Figure 1: The four kinds of vertex in a web

approach might start by finding the Hopf algebra $U_q(\mathfrak{sl}_n)$ also encoded in the new spider, generalizing the case $n = 2$ that was done in [1].

I also do not know whether the new spider is a diagrammatic encoding of some other known approach to the SL_n spider. I think it has something to do with the graph planar algebra, as defined in [4], but for paths in the root lattice. With a more sophisticated definition, it should be possible to encode the Ocneanu cells for the $\mathcal{A}^{(n)}$ graph, as defined in [3].

A bonus feature of our approach is that it extends, with no extra effort, to webs with virtual crossings. These satisfy detour moves, but not virtual Reidemeister one. This therefore defines an invariant of "rotational virtual" knots and links, which is presumably the same as the one defined by Kauffman [5].

2 Webs

A *web* in the SL_n spider is a certain kind of graph drawn in the plane. Each strand is oriented and is labeled by an integer from $\{1, \ldots, n-1\}$. A web can have bivalent and trivalent vertices, of the forms shown in Figure 1. At a trivalent vertex, either strands labeled k and l fuse to a strand labeled $k+l$, or a strand labeled $k+l$ divides into strands labeled k and l. At a bivalent vertex, strands labeled k and $n-k$ either both enter or both exit the vertex, and a *tag* points to one of the two sides of the vertex, breaking its rotational symmetry.

The relations are shown in Figure 2. Here, and throughout the paper, we omit some of the labels on strands, as long as they can be deduced from context. We call the relations:

- switching a tag,
- canceling a pair of tags,
- $I = H$,
- bursting a digon,
- bursting a square.

We allow strands to be labeled n, with the convention any such strand can be deleted. Any trivalent vertex that had a strand labeled n then becomes a

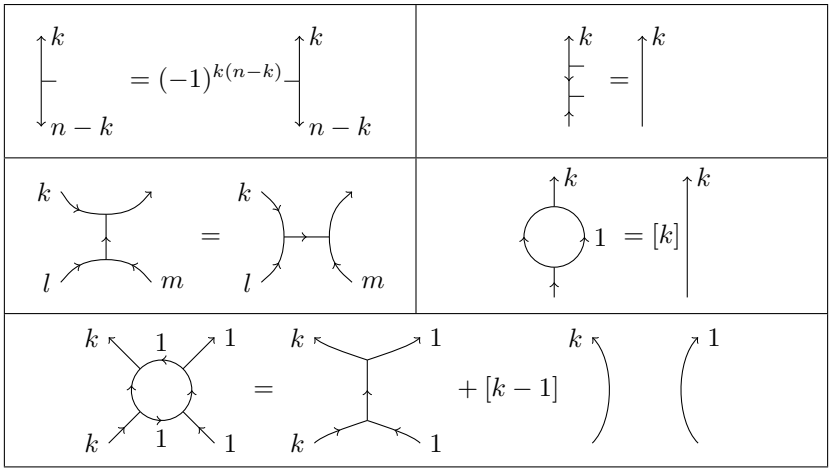


Figure 2: Web relations. Some labels are omitted, but can be deduced.

bivalent vertex with a tag on the side of the deleted strand. We also impose the relations obtained by reversing all of the arrows in any of the above relations.

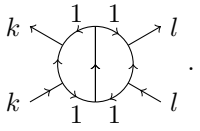
Note that we have used a convention for tags that is different from that of [2]. To recover the convention in [2], switch the side of the tag on every bivalent vertex whose strands are oriented inward. We have also used a smaller set of relations than in [2], so as to make our job easier.

Proposition 1. *The above set of relations is equivalent to the defining relations given in [2], but with a different convention for tags.*

Proof. Our relations appear in [2] as equations (2.3), (2.12), (2.6), (2.4) for $l = 1$, and (2.10) for $r = s = l = 1$. Conversely, the defining relations (2.3) through (2.10) in [2] can be deduced from the relations given here. We leave this as an exercise, with hints.

"Tag migration" (2.7) is a special case of the $I = H$ relation. The second version of "tag migration" (2.8) follows from the first, together with tag cancellation. "Square removal" (2.9) is proved by induction on s, where the case $s = 1$ follows from the $I = H$ relation and our digon bursting relation. "Removing a bigon" (2.4) is proved by induction on l, using the square removal relation. The other version of "removing a bigon" (2.5) follows from the first and tag cancellation.

For the "square switching" relation (2.10), note that we can burst either of the two squares in the web



.

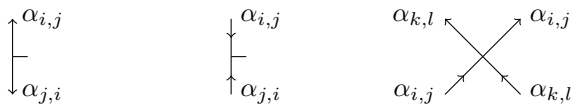


Figure 3: Bivalent vertices and virtual crossings in a cobweb.

Equating the two outcomes implies the case $r = s = 1$ of the square switching relation. The general case then follows by induction. □

3 Cobwebs

Biologists classify spider webs into types. If the diagrams in the previous section are supposed to resemble the usual "orb webs", then the diagrams introduced in this section more closely resemble "cobwebs", since they have no vertices except for bivalent vertices and virtual crossings.

Let $e_1, \ldots, e_n$ be the standard basis vectors in $\mathbb{R}^n$. The *roots* of the A_{n-1} root system are the vectors

$$\alpha_{i,j} = e_i - e_j$$

for $i, j \in \{1, \ldots, n\}$ with $i \neq j$. Call $\alpha_{i,j}$ a *positive root* if $i > j$.

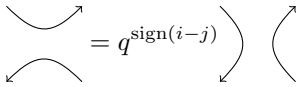
A *cobweb* is a collection of strands drawn in the plane. Each strand is oriented and is labeled by a root. A cobweb can have virtual crossings and bivalent vertices, as shown in Figure 3. A bivalent vertex has strands labeled $\alpha_{i,j}$ and $\alpha_{j,i}$, oriented both in or both out, and has a tag that points to one side of the vertex. A virtual crossing is a point where a pair of strands pass through each other.

The relations are shown in Figure 4. These apply for any consistent way to fill in the omitted orientations and labels. We call them:

- switching a tag,
- canceling two tags,
- bursting a bubble,
- three detour moves: virtual Reidemeister moves two and three, and sliding a virtual crossing past a tag,
- smoothing a virtual crossing of two strands that have the same label.

We will also need the following "saddle relation".

Proposition 2. *The following is a consequence of the cobweb relations, where all strands have the same label* $\alpha_{i,j}$*.*



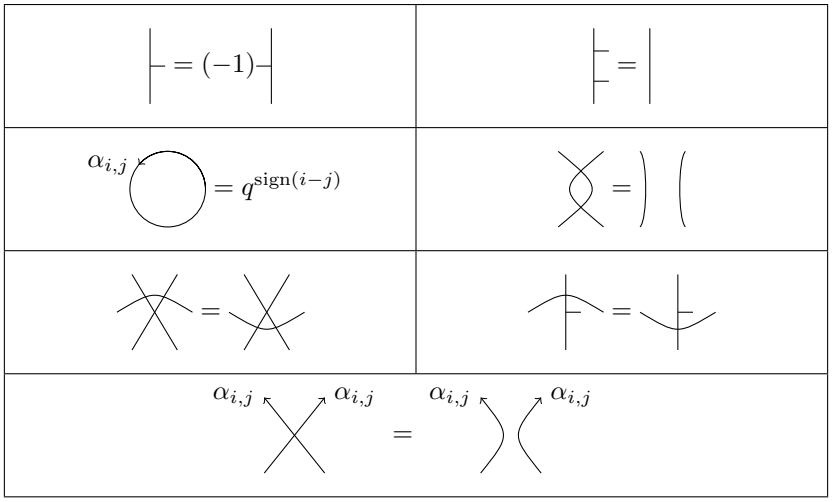


Figure 4: Cobweb relations. Unlabeled strands can have any consistent orientations and labels.

Proof. Starting with the cobweb on the left, use virtual Reidemeister two to introduce two virtual crossings, and then smooth both of these crossings, and burst the resulting bubble. □

Other important consequences of the cobweb relations include virtual Reidemeister one with a scalar, and bursting a negatively oriented bubble.

Here, all strands have the same label $\alpha_{i,j}$.

Whenever something is defined by generators and relations, it is worth checking that it is not trivial. The following proposition does this.

Proposition 3. *The space spanned by closed cobwebs is one-dimensional.*

Proof. It is easy to use the cobweb relations to reduce any closed cobweb to some monomial $\pm q^t$ times the empty cobweb. We will describe how to compute this monomial in a completely canonical way.

It is best here to think of each bivalent vertex as a place where a single strand changes its label and orientation. With this convention, a closed cobweb consists of a collection of immersed loops.

Give each loop the overall orientation that is the same as all segments labeled by a positive root, and the opposite of all segments labeled by a negative root. With respect to these orientations, let s be the number of tags that are on the left side of their loop, and let t be the sum of the turning numbers of all loops.

Our desired monomial is then $(-1)^s q^t$. Note that we would get the same answer if we counted tags on the right side of loops, since there must be an even number of tags in total.

Our calculation is easily shown to be invariant under all cobweb relations. This implies that the vector space of closed cobwebs modulo cobweb relations is precisely the vector space of scalar multiples of the empty diagram. □

4 The map

We will define a map that takes each web to a sum of cobwebs. We use a state sum. A *state* on a web is a way to assign a subset of $\{1, \ldots, n\}$ to each strand such that the following conditions are satisfied. If a strand is labeled by the integer k, then it must be assigned a set that has cardinality k. The strands at a trivalent vertex must be assigned sets K, L and $K \cup L$, where K and L are disjoint. The strands at a bivalent vertex must be assigned sets K and L such that K and L are disjoint, and $K \cup L = \{1, \ldots, n\}$.

For every state, we define a corresponding cobweb as follows. If a strand is labeled by a set K then replace it by a collection of parallel strands, with one strand labeled $\alpha_{i,j}$ for every $i \in K$ and $j \notin K$. The orientation of these parallel strands should be the same as that of the original strand. Because of virtual crossings, it does not matter in what order they are arranged.

Consider a trivalent vertex in which strands enter from the left and right, and one strand exits to the top. Suppose the strands are assigned the sets K on the left, L on the right, and $K \cup L$ on the top. We replace this vertex with a cobweb that has the following strands:

- for every $i \in K$ and $j \notin K \cup L$ there is a strand labeled $\alpha_{i,j}$ going from the left to the top,
- for every $i \in L$ and $j \notin K \cup L$ there is a strand labeled $\alpha_{i,j}$ going from the right to the top, and
- for every $i \in K$ and $j \in L$ there are strands labeled $\alpha_{i,j}$ and $\alpha_{j,i}$ coming from the left and right to meet at a bivalent vertex with a tag pointing up.

For a trivalent vertex with one strand oriented in and two strands oriented out, simply reverse all of the arrows in the above.

Now consider a bivalent vertex with incoming strands entering from the left and right, and a tag pointing up. Suppose the strands are assigned the sets K on the left and L on the right. We map this vertex to a cobweb with the following strands. For every $i \in K$ and $j \in L$, there are strands labeled $\alpha_{i,j}$ and $\alpha_{j,i}$ coming from the left and right to meet at a bivalent vertex with a tag pointing up. For a bivalent vertex with two strands oriented out, simply reverse all of the arrows in the above.

Note that, for any vertex with any state, the two or three strands map to parallel sets of cobweb strands, and the vertex maps to a way to connect up the

endpoints from these parallel sets of strands. Thus, for any web with any state, the vertices and edges map to cobwebs that join up correctly to make a single complete cobweb.

We can at least partly explain where the definitions in this section come from. A strand labeled k corresponds to the kth exterior power of the defining representation of SL_n. The weight system for this representation is the set of vectors

$$\lambda_K = \sum_{i \in K} e_i$$

such that K has cardinality k. (Strictly speaking, we should project these vectors onto the hyperplane spanned by the roots, but this will not make any difference.) A strand assigned the weight λ_K is mapped to parallel strands, with one for each root whose inner product with λ_K is one. The rule for vertices is not so easily explained. It is, in some sense, the simplest rule possible, but the real justification is that it works, as we prove next.

5 The map is well-defined

Proposition 4. *The map defined in Section 4 is a well-defined injective map from the SL_n spider to the cobweb spider.*

Proof. Assuming the map is well-defined, it is clearly not the zero map, since it takes the empty web to the empty cobweb, which is non-zero by Proposition 3. Injectivity then follows from the fact that the SL_n spider has no non-zero proper ideals.

To prove that the map is well-defined, we check that it respects each of the web relations listed in Section 2. A set of *boundary data* on a relation is a way to assign a set to each of the strands on the boundary, where we assign the same set to every strand at any given position on the boundary. For all such boundary data, we replace each web by the sum of cobwebs over all states that are consistent with the boundary data. We then check that the left and right sides of the equation map to the same linear combination of cobwebs, up to the cobweb relations from Section 3. We must do this for every relation, and every choice of boundary data that extends to at least one state on at least one web in the relation.

We will repeatedly find ourselves computing the cobweb corresponding to a state on a web. It is easiest to do this one pair of roots $\pm\alpha_{i,j}$ at a time. We check cases based on where i and j lie. Cobweb strands with labels $\pm\alpha_{i,j}$ appear wherever a web strand was assigned a set that contains exactly one of i and j.

5.1 Strands labeled n

A strand that is labeled n must be assigned the set $\{1, \ldots, n\}$, and then mapped to an empty collection of parallel cobweb strands. All of our calculations will be compatible with the convention for strands labeled n.

5.2 Reversing all orientations

The cobweb relations are invariant under switching all orientations and mapping q to q^{-1}. The coefficients in the web relations are all invariant under mapping q to q^{-1}, so all of our proofs will also apply when all orientations are reversed.

5.3 Tag operations

The tag switching and tag canceling relations for webs follow immediately from the same relations for cobwebs. Note that a strand labeled k is mapped to $k(n-k)$ parallel cobweb strands, so a tag switch in the web corresponds to $k(n-k)$ tag switches in the cobweb.

5.4 The $I = H$ relation

Assign the sets K, L, and M to the strands labeled k, l and m, and $K \cup L \cup M$ to the strands at the top-right. Here, K, L, and M are disjoint and have cardinalities k, l, and m. We must then assign $L \cup M$ to the vertical strand in the I-shaped web, and $K \cup M$ to the horizontal strand in the H-shaped web. Each side of the equation is thus mapped to a single cobweb.

Let $\alpha_{i,j}$ be a root. We want to show that the strands labeled $\alpha_{i,j}$ and $\alpha_{j,i}$ are the same for the I- and H-shaped webs. To be excessively systematic, we could check sixteen cases, based on whether i lies in K, L, M, or none of them, and similarly for j. For example, if $i \in K$ and $j \in L$ then there are strands labeled $\alpha_{i,j}$ and $\alpha_{j,i}$ coming from the top-left and bottom-left, and meeting at a bivalent vertex with a tag to the right. If $i \in K$ and $j \notin K \cup L \cup M$ then there is a strand labeled $\alpha_{i,j}$ going from the top-left to the top-right. The remaining cases are similar.

5.5 Bursting a digon

Assign the set K to the strands on the top and bottom of the webs. A state on the web on the left side of the equation corresponds to a choice of $a \in K$. Namely, assign $K \setminus \{a\}$ and $\{a\}$ to the left and right sides of the digon.

The corresponding cobweb has the following strands. For every $i \in K$ and $j \notin K$, there is a strand labeled $\alpha_{i,j}$ going vertically from the bottom of the cobweb to the top. For every $j \in K \setminus \{a\}$, there are strands labeled $\alpha_{j,a}$ and $\alpha_{a,j}$ going up the left and right side of the digon, and meeting at two bivalent vertices. We can cancel the tags to get a positively oriented bubble labeled $\alpha_{a,j}$. Bursting these bubbles over all $j \in K \setminus \{a\}$ gives q to the power of the exponent:

$$|\{j \in K \mid j < a\}| - |\{j \in K \mid j > a\}|.$$

Now take the sum over all $a \in K$. The vertical strands of the cobwebs are the same in all terms, and the powers of q sum to $[k]$. This is the same as the right side of the digon relation.

5.6 Bursting a square

Assign sets to each of the four strands on the boundary of the webs. There are three cases to consider.

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

Let $K' = K \setminus \{y\}$. A state on the web on the left side of the equation corresponds to a choice of $a \in K'$. Namely, assign $\{a\}$ to the horizontal strands, $\{a, y\}$ to the right side of the square, and $K \setminus \{a\}$ to the left.

By checking cases, we can find all strands in the cobweb corresponding to the above state. All strands that start at the bottom of the cobweb go vertically to the top, after canceling two pairs of tags in the case of $\alpha_{y,a}$. For all $j \in K' \setminus \{a\}$, there is a closed loop made by strands labeled $\alpha_{a,j}$ and $\alpha_{j,a}$ meeting at two bivalent vertices. We can cancel the tags to get a positively oriented bubble labeled $\alpha_{a,j}$. Bursting all of these bubbles gives q to the power of the exponent

$$|\{j \in K' \mid j < a\}| - |\{j \in K' \mid j > a\}|.$$

Now take the sum over all $a \in K'$. The vertical strands are the same for every term, and the powers of q add up to $[k-1]$.

On the right side of the equation, the first term has no state that is consistent with the boundary data, and the second has one. We get the same result as for the left side of the equation.

Case 2: Let K' have cardinality $k-1$, and $x, y \notin K'$. Assign $K' \cup \{x\}$ and $\{y\}$ to the bottom-left and bottom-right strands of all webs. Assign $K' \cup \{y\}$ and $\{x\}$ to the top-left and top-right strands of all webs.

Consider the web on the left side of the equation. There is only one state consistent with the boundary data. Namely, assign $\{y\}$ and $\{x\}$ to the top and bottom sides of the square, and assign K' and $\{x, y\}$ to the left and right sides. By checking cases, the corresponding cobweb has the following strands:

- for every $j \in K' \cup \{y\}$, there is a cup formed by strands labeled $\alpha_{x,j}$ and $\alpha_{j,x}$ that meet at a bivalent vertex with a downward tag,
- for every $j \in K' \cup \{x\}$, there is a cap formed by strands labeled $\alpha_{y,j}$ and $\alpha_{j,y}$ that meet at a bivalent vertex with an upward tag, and
- every other strand goes from the bottom of the cobweb to the top.

On the right side of the equation, the first term has one state that is consistent with the boundary data, and the second has none. The cobweb corresponding to the first term is the same as the cobweb corresponding to the left side of the equation.

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.

Consider the web on the left side of the equation. A state corresponds to a choice of $a \in K$. Namely, assign $\{a\}$ to the top and bottom sides of the square,

and assign $K \setminus \{a\}$ and $\{a, y\}$ to the left and right sides. After canceling tags, the corresponding cobweb has the following strands:

- for $j \in K \setminus \{a\}$, there is a positively oriented bubble labeled $\alpha_{a,j}$,
- the strands labeled $\alpha_{a,y}$ and $\alpha_{y,a}$ form a cup and a cap with two bivalent vertices, and
- every other strand goes vertically from the bottom of the cobweb to the top.

Use Proposition 2 to convert the cup and cap to $q^{\text{sign}(y-a)}$ times vertical strands with a canceling pair of tags. This, and bursting all bubbles, gives us the coefficient q to the power of the exponent

$$|\{j \in K \mid j < a\}| - |\{j \in K \mid j > a\}| + \text{sign}(y - a).$$

Take the sum these powers of q over all $a \in K$. If it were not for the $\text{sign}(y-a)$ term in the exponent of q, we would get $[n]$. The $\text{sign}(y-a)$ changes this to $[n-1]$ but with one term repeated. Specifically, we get

$$q^m + [n-1], \text{ where } m = |\{j \in K \mid j < y\}| - |\{j \in K \mid j > y\}|.$$

Now consider the first term on the right side of the equation. This has one state consistent with the boundary data. In the cobweb corresponding to this state, all strands go vertically from the bottom to the top with the following exceptions. For all $j \in K$, the strands labeled $\alpha_{j,y}$ and $\alpha_{y,j}$ form a cup and a cap with two bivalent vertices. Use Proposition 2 to convert each such cup and cap to $q^{\text{sign}(y-j)}$ times vertical strands with a canceling pair of tags. These coefficients combine to give q^m, with the same m as above.

Finally, consider the second term on the right side of the equation. This has one state that is consistent with the boundary data. The corresponding cobweb has all vertical strands.

In the end, all cobwebs have the same vertical strands, and the coefficients add up to $q^m + [n-1]$ on either side of the equation. This completes the proof. □

References

[1] Stephen J. Bigelow. A diagrammatic definition of $U_q(\mathfrak{sl}_2)$. *J. Knot Theory Ramifications*, 23(6):1450036, 9, 2014.

[2] Sabin Cautis, Joel Kamnitzer, and Scott Morrison. Webs and quantum skew Howe duality. *Math. Ann.*, 360(1-2):351–390, 2014.

[3] David E. Evans and Mathew Pugh. Ocneanu cells and Boltzmann weights for the SU(3) $\mathcal{ADE}$ graphs. *Münster J. Math.*, 2:95–142, 2009.

[4] Vaughan F. R. Jones. The planar algebra of a bipartite graph. In *Knots in Hellas '98 (Delphi)*, volume 24 of *Ser. Knots Everything*, pages 94–117. World Sci. Publ., River Edge, NJ, 2000.

[5] Louis H. Kauffman. Rotational virtual knots and quantum link invariants. *J. Knot Theory Ramifications*, 24(13):1541008, 46, 2015.

[6] Greg Kuperberg. Spiders for rank 2 Lie algebras. *Comm. Math. Phys.*, 180(1):109–151, 1996.