The Jellyfish Algorithm

Ellie Grano

January 27, 2012

1 Background

1.1 Overview

The jellyfish algorithm was originally introduced in [Big09] and given its name in [BMPS09]. These papers describe certain subfactor planar algebras using generators and relations. The jellyfish algorithm is used to show that every closed diagram of the particular planar algebra is a multiple of the empty diagram. This is important for many reasons, including the possibility for an inner product. However the proofs that the zero box space is in fact isomorphic to $\mathbb{C}$ relies on known properties of the subfactor planar algebras and their principal graphs.

In [MPS08], a different algorithm was given to show every diagram was a multiple of the empty diagram for another subfactor planar algebra. They gave a combinatorial argument to show that the algorthim was well-defined so that the empty diagram was not zero.

Here we use the jellyfish algorithm of [Big09] and [BMPS09] on the D_{2n} planar algebra as it is presented in [Big09], mimicking the combinatorial approach in [MPS08].

First, we will state some necessary preliminary facts followed by the definition the D_{2n} planar algebra. Then we will define the jellyfish algorithm and give an example. The jellyfish algorithm is a function from closed diagrams to the complex numbers. There are many choices made in the algorithm, so we will first show the algorithm is well-defined. Then we will handle the less difficult task of showing the relations are satisfied so that the algorithm is well-defined on the planar algebra. Once we have done so, we'll have that the closed diagrams are isomorphic to the complex numbers, thus proving of the following theorem:

Theorem 1.1 *The planar algebra PA(S) is not the trivial planar algebra.*

1.2 Preliminary Definitions

Definition quantum numbers

Definition Temperley-Lieb at the special value q

Definition JW idempotents

Definition JW partial trace

Definition crossing in TL

Lemma 1.2 *The following crossing changes involving f^{4n-4} are equal in Temperley-Lieb at the special value q.*

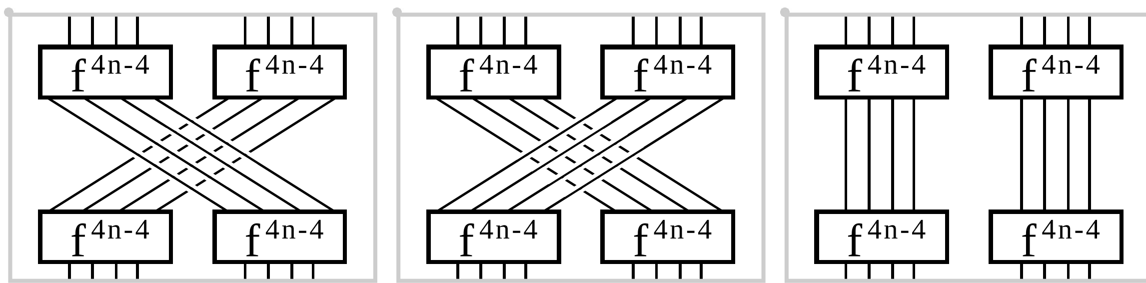


Figure 1: Lemma 1.2

This result is given in the D_{2n} paper. Need another reference though.

1.3 Define $PA(S)$

Definition pictures for the relations like S^2 (as in the ADE paper)

This will be the D_{2n} Planar Algebra defined by generators and relations.

Definition Let $PA(S)$ be the planar algebra generated by a single box S with $4n-4$ strands subject to the following relations:

1. A closed circle is equal to $[2]_q$ times the empty diagram.
2. Clockwise rotation of the S-box introduces an i.
3. Capping an S-box equals zero.
4. S^2 equals the Jones-Wenzl projection f^{2n-2}.

2 The Jellyfish Algorithm

2.1 Definition of the Jellyfish Algorithm

Definition This is a function from $PA(S)_0$ (i.e. any closed diagram) to the complex numbers.

1. Part I: Put diagram in jellyfish form

   a) For each S-box in the diagram, draw an imaginary arc from it to the top of the diagram (transverse to everything in the diagram).

   b) Chose an ordering on the S-boxes.

   c) Following this ordering, drag each S-box one by one along its respective arc to the top of the diagram, passing under any other strands in the diagram.

   d) Evaluate the crossings and burst any bubbles as in Temperley-Lieb at $[2]_q$. Now we have a linear combination of diagrams in jellyfish form.

   e) For each diagram in jellyfish form follow Part II of the algorithm:

2. Part II: Evaluate

   a) If there is a cap on an S-box, evaluate the diagram as zero. Note: this includes the case if there is exactly one S-box.

   b) If there are two or more S-boxes, Lemma 2.1 will show that there is at least one pair of S-boxes connected by at least $2n-2$ strands.

      i. Choose such a pair of S-boxes and choose $2n-2$ consecutive strands connecting the S-boxes.

      ii. Replace the chosen pair of S-boxes with an f^{2n-2} following the $2n-2$ chosen strands.

      iii. Further, count clockwise from the marked point of one S-box the number of strands prior to (up to but not including) the chosen strands, say k strands, and counterclockwise from the marked point of the other S-box the number of strands prior to the $2n-2$ chosen strands, say l strands, and insert a factor of i raised to the $k-l$ power.

      iv. Evaluate the crossings and burst any bubbles as in Temperley-Lieb at $[2]_q$ and start Part II over again on each new diagram in jellyfish form.

   c) If there are no S-boxes in the diagram, evaluate the diagram in Temperley-Lieb at $[2]_q$.

Lemma 2.1 *Given a diagram in jellyfish form, there is at least one pair of S-boxes connected by at least $2n-2$ strands.*

Proof Line up the S-boxes in a row and draw one line connecting each to its neighbor. Each line has a number attached representing the multiplicity of strands between the S-boxes (anything from zero to $4n-4$). Then draw one line between non adjacent S-boxes that have multiplicity greater than zero. This produces an arc in the diagram. Since there are no crossings, we may only draw non-interesting arcs. Thus there must be an innermost arc. So there is an S-box connected to two or fewer S-boxes. Since no strands connect one S-box to itself, its $4n-4$ strands are divided in some way between the two S-boxes. Thus one of the S-boxes has at least $2n-2$ strands connecting it to the other S-box. Therefore we have a pair of S-boxes connected by at least $2n-2$ strands.

2.2 The Jellyfish Algorithm is Well-Defined for $PA(S)_0$

Theorem 2.2 *The jellyfish algorithm is well-defined for* $PA(S)_0$

In Section 2.2.1 we'll show Part II of the algorithm is well-defined for diagrams already in jellyfish form. Then in Section 2.2.2 we'll show Part I is well-defined for an arbitrary diagram so that the algorithm is well-defined for closed diagrams. To get well-definedness for $PA(S)_0$, in Section 2.2.3 we'll show the algorithm respects the relations presented in Definition 1.3.

2.2.1 Part II well-defined

Lemma 2.3 *Given a diagram in jellyfish form and two different applications of part II of the algorithm we get the same result.*

We'll show that each of the choices does not affect the evaluation result starting from the bottom up. So given a chosen pair of S-boxes and $2n-2$ chosen strands, the clockwise vs. counterclockwise choice does not affect the answer (Lemma 2.4). Then given a chosen pair of S-boxes, the choice of $2n-2$ consecutive strands does not change the result (Lemma 2.5). Then we will have for two applications of the algorithm with the same S-box pairings, the evaluation will be the same. Finally, the pairings of S-boxes can be changed and the answer will be the same (Lemma 2.7). Thus any difference in choices will result in the same answer.

Lemma 2.4 *Suppose we have a pair of S-boxes, say S-box 1 and S-box 2, and $2n-2$ chosen consecutive strands between the two. Then choosing to count from the marked point of S-box 1 clockwise, and 2 counterclockwise results in the same evaluation as choosing to count 2 clockwise and 1 counterclockwise. Here we assume the rest of the choices are the same.*

Proof First note that the last sentence makes sense because the choice of clockwise vs. counterclockwise affects only a coefficient. Now consider the coefficient for each choice. Suppose on S-box 1, there are k strands prior to the chosen strands clockwise from the marked point. Then counterclockwise from the marked point of S-box 1 the number of strands prior to the chosen strands will be $(4n-4)-(2n-2)-k = 2n-2-k$. Similarly, if S-box 2 has l strands counterclockwise from the marked point to the chosen strands, then we will have $2n-2-l$ for clockwise. Then choosing to count from the marked point of 1 clockwise, and 2 counterclockwise results in a coefficient of i^{k-l}. While choosing to count 2 clockwise, and 1 counterclockwise results in $i^{(2n-2-l)-(2n-2-k)} = i^{k-l}$. Thus the result is the same.

Lemma 2.5 *Given one chosen pair of S-boxes connected by at least $2n-2$ strands, different choices of $2n-2$ consecutive strands in the evaluation will yield the same result. Assume the rest of the choices of the algorithm are the same.*

Proof First let us convince ourselves that the remaining choices in the algorithm will be independent of our selection of strands. Consider two applications of the algorithm with different choices of strands for one pair of S-boxes, choice 1 and choice 2. Since there are no crossings in the diagram and no caps on the S-boxes, all strands connecting the pair must be consecutive. Thus, when the pair of S-boxes is replaced with a f^{2n-2}, these neighboring strands become partial traces of f^{2n-2} in the resulting diagram. The only difference is on which side the loops appear. These diagrams are equivalent in Temperly-Lieb, so once we reach the final stage of the algorithm we will have equivalence.

To see that the coefficient corresponding to the evaluation is independent of strand choice, we will also use that all strands connecting the pair are consecutive. Notice that counting clockwise the number of strands on one S-box corresponds to counting counterclockwise on the other S-box. Since we have Lemma 2.4, we can choice either S-box to count clockwise on. So if choice a is k strands more clockwise on S-box 1 than choice b, then choice a will be k strands more counterclockwise on S-box 2 than choice b. So if choice a yields coefficient i^{r-s}, then choice b yields coefficient $i^{(r+n)-(s+n)} = i^{r-s}$. , we get the same result for different choices of strands on a given pair of S-boxes.

Lemma 2.6 *Given the same pairings of S-boxes, clockwise rotation of the marked point of one S-box results in multiplication by i.*

Proof We use the same pairings of S-boxes by hypothesis. By the previous lemmas, we may make the same strand choices and clockwise vs. counterclockwise choices. Thus we can assure that there will be no difference in the diagrams, only in the coefficients. If there is an odd number of S-boxes, the diagram evaluates to zero, and the statement is true. Otherwise the S-box will be in a pair and we may choose to count it clockwise. Then if the original diagram has a coefficient of i^{k-l}, the other will be i^{k+1-l} since the marked point will be exactly one strand further clockwise from the chosen strands. Then we are done.

Lemma 2.7 *Two applications of the algorithm using different pairings of S-boxes give the same result.*

Proof Induct on the number of S-boxes in the diagram.

If there are fewer than four S-boxes then there is no choice of pairing of S-boxes. This is because if there is an odd number of S-boxes the diagram will evaluate to zero. If the diagram has no S-boxes it is in Temperley-Lieb, so there are no choices. If there are two S-boxes in the diagram, then this will be the only choice for a pair. Existence of strands between the two is given by Lemma 2.1. Then by Lemma 2.4 and Lemma 2.5 we have that any other choices in the algorithm will not change the result.

Suppose there are $k > 4$ S-boxes in the diagram and the evaluation of diagrams with fewer than k S-boxes is independent of the choice of pairings. For each application of the algorithm consider the first choice of pair of S-boxes. If both applications of the algorithm select the same pair of S-boxes as the first choice, then the evaluation results will be equal by the previous two lemmas and the inductive hypothesis. Then either the first choices are distinct or they share exactly one S-box in common.

Case 1: The first pair of S-boxes evaluated for one application of the algorithm are distinct from the first pair for the other. Say application a of the algorithm evaluates the S-boxes 1 and 2 first while b evaluates the S-boxes 3 and 4 first. Once we have evaluated 1 and 2 for a, we will have a linear combination of diagrams going through Part II of the algorithm again. In each diagram of the linear combination, the$S$-boxes 3 and 4 will still have at least $2n-2$ strands connecting each other. The only way replacing S-boxes 1 and 2 with a Jones-Wenzl idempotent could affect S-boxes 3 and 4 is by adding possible caps on these S-boxes. In this case the algorithm will always give zero. Thus by inductive hypothesis we are free to assume the second choice for a is S-boxes 3 and 4.

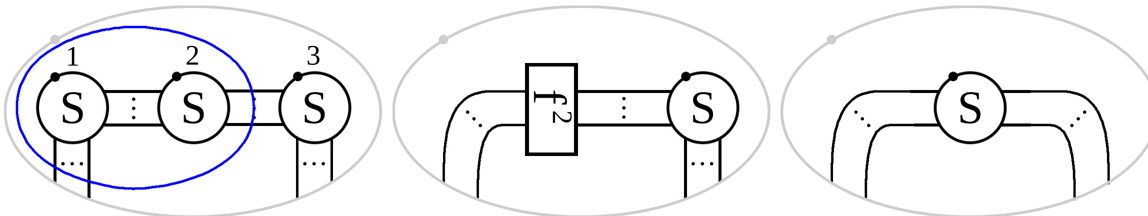
Similarly for choice b, the result is the same as though we choose to evaluate 1 and2 after we first evaluate 3 and4. Both of the diagrams will be the same linear combinations of the same diagrams at this point. By the inductive hypothesis the following choices of S-box pairs will not affect the result of the evaluation.

Case 2: The first pair of S-boxes evaluated for the two applications of the algorithm share exactly one S-box in common. Say application a evaluates the S-boxes 1 and 2 first while b evaluates the S-boxes 2 and 3 first. Then necessarily half of the strands connected to S-box 2 will connect to 1 and the other half to 3. So application a and b must be as in Figure 2(a) and Figure 2(d), with possibly rotation for the marked point. Suppose that the marked point of S-box 1 is r strands clockwise from the marked point in our diagram, S-box 2 s strands, and S-box 2 t strands.

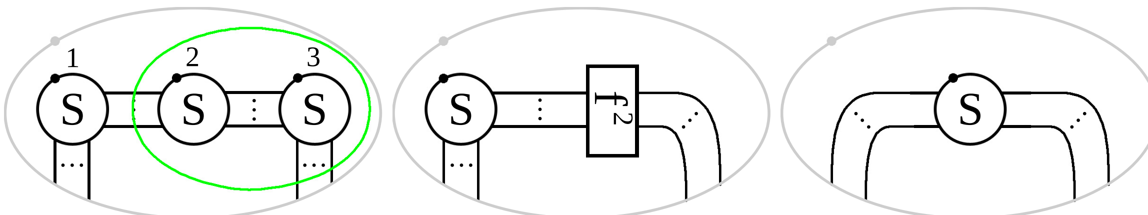
Consider application a of the algorithm. We first evaluate S-boxes 1 and 2. By Lemma 2.6, this will be equal to evaluating Figure 2(b) with an additional coefficient of i^{r+s}. But f^{2n-2} is the identity plus a linear combination of diagrams with caps. Each diagram with a cap on the remaining S-box 3 will evaluate to zero in the algorithm. So evaluating Figure 2(b) is equivalent to evaluating the single S-box 3. Then by Lemma 2.6 this will evaluate the same as Figure 2(f) with an additional coefficient of i^t. In summary, application a evaluates the same as Figure 2(f) multiplied by i^{r+s+t}.

Similarly, the evaluation of b will be equivalent to that of Figure 2(e) times i^{s+t}. Then evaluating Figure 2(e) is equivalent to evaluating the single S-box 1. Then by

Lemma 2.6 this will evaluate the same as Figure 2(f) with an additional coefficient of i^r. Thus application a evaluates the same as Figure 2(f) multiplied by i^{r+s+t}. Then by the inductive hypothesis the remaining pairings do not affect the outcome and the proof is complete.



(a) Application a (b) Replace 1 and 2 with f^{2n-2} (c) Equivalent Evaluation



(d) Application b (e) Replace 2 and 3 with f^{2n-2} (f) Equivalent Evaluation

Figure 2: Case 2 of Lemma 2.7

2.2.2 Part I

To finish the proof of Theorem 2.2, Lemma 2.12 will show changing the order for which the S-boxes are dragged along their arcs doesn't change the output of the evaluation. Then Lemma 2.14 will show that the arcs themselves can be chosen arbitrarily and that we get the same output. Thus we will have well-definedness.

Lemma 2.8 *If we change a diagram by equivalent elements of Temperley-Lieb in the following situations, the evaluation result is the same.*

1. *After the S-boxes have traveled along their S-box arcs to the top of the diagram (after part I(c) of the algorithm).*
2. *Before the algorithm starts given that the portion of the diagram we are changing avoids the S-boxes and the S-box arcs and all the choices in the algorithm remain the same.*

Proof 1. Suppose we have a diagram whose S-boxes have already traveled along their S-box arcs to the top of the diagram. Now replace the given portion of the diagram that is in Temperley-Lieb (no S-boxes) with the equivalent diagram in Temperley-Lieb. The next step of the algorithm is to evaluate all the crossings and burst all the bubbles as in Temperley-Lieb. Thus the diagrams will be the exact same linear combination of diagrams up to planar isotopy of the strands. So we are free to make the same choices for the rest of the algorithm. Once in Temperley-Lieb, the evaluation will be the same.

2. Here, we have a diagram with S-box arcs chosen and an ordering given on these arcs. Then we replace a portion of the diagram avoiding the S-boxes and the S-box arcs. Since the switch avoided all S-boxes and S-box arcs, we are able to make the same choices once the switch is made. Once we move the S-boxes along the arcs, each arc becomes $4n-4$ strands. Then since the difference in Temperley-Lieb avoided the arcs, part 1 shows that the equivalence holds.

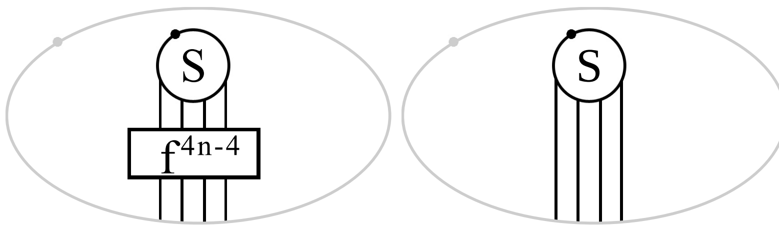
Lemma 2.9 *A diagram with a cap on an S-box will always evaluate to zero.*

Proof The arc portion of the algorithm might pull strands over or under the cap, but none through the cap. So using the Lemma 2.8, the cap can be dragged under or over the strands one by one, and moved so that it is again a cap on the S-box. The algorithm then gives us zero.

Lemma 2.10 *Given a diagram with arcs for each S-box, replacing one arc with an f^{4n-4} rather than just $4n-4$ strands will yield the same evaluation result.*

Proof First consider replacing the arc with a f^{4n-4} just under the S-box as in Figure 7. Recall that each Jones-Wenzl idempotent is the identity plus a linear combination of diagrams with caps. Each diagram with a cap on the S-box will evaluate to zero by Lemma 2.9. Thus the evaluation will be the same with replaced by f^{4n-4} as by the identity.

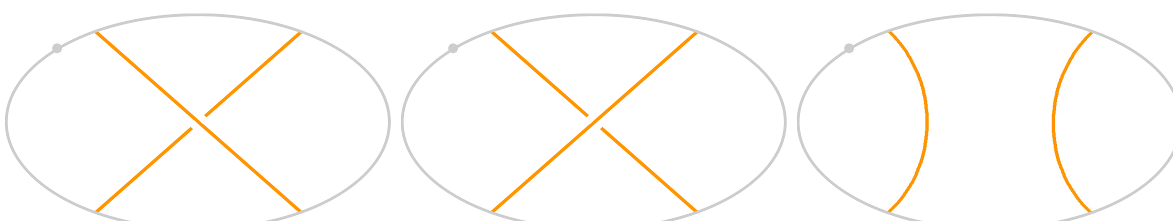
Then by Lemma **??** we can move the f^{4n-4} along the strands of the arc. Thus we could have placed the f^{4n-4} anywhere on the arc without changing the result of the algorithm.



(a) (b)

Figure 3: Lemma 2.10

Lemma 2.11 *Suppose we have a diagram with chosen S-box arcs. Then we may interchange the diagrams of arcs in Figure 4(a), Figure 4(b), and Figure 4(c) without affecting the result. Here the lines represent the S-box arcs and the crossings indicate how the $4n-4$ strands will cross.*



(a) (b) (c)

Figure 4: Lemma 2.11

Proof By Lemma 2.10, we may add in an f^{4n-4} into the diagram along an arc. Then we may add in four f^{4n-4}'s in each of the three diagrams as in Figure 1(a), Figure 1(b), and Figure 1(c). By Lemma 1.2, all the diagrams in Figure 1 are equal in Temperley-Lieb. So by Lemma 2.8 each diagram will evaluate the same. Thus, changing the crossing will evaluate the same.

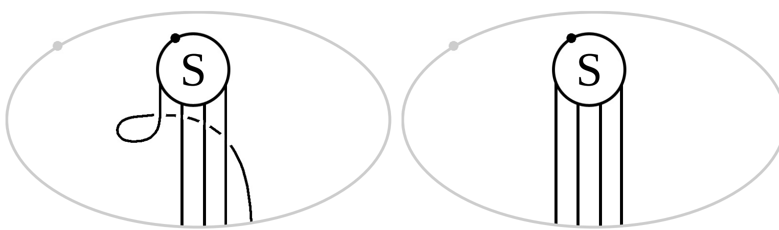
Lemma 2.12 *Suppose we have a diagram and have chosen one arc for each S-box. Then choosing two different orderings on these arcs yields the same result.*

Proof It is enough to show that changing the orders of two consecutive S-boxes yields the same result. Then by permutation we can change the order of any of them.

Consider the two S-boxes whose order we would like to swap. Where ever the two arcs cross, simply change the crossing as Lemma 2.11. This is exactly the diagram with the order of the two arcs swapped. So they evaluate the same.

Lemma 2.13 *Using the Reidemeister moves on the arcs does not affect the result of the evaluation.*

Proof Notice here, each arc represents what will be $4n-4$ strands, and since we already have Lemma 2.8 we have all but Reidemeister 1. First consider a single twist of one strand as in Figure 5(a). Here, the three strands represent $4n-5$ strands. Once we evaluate the crossings, we will end up with $q^{2n-1} = i$ times Figure 5(b) plus diagrams with caps on the S-box. By Lemma 2.9 these will all evaluate to zero. So Figure 5(a) will evaluate to i times Figure 5(b).



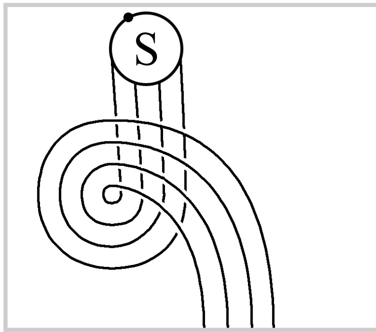
(a) (b)

Figure 5: One twist evaluates the same as i times no twist.

Now a positive twist in an S-box arc near the S-box will result in $4n-4$ strands tucking under $4n-4$ strands as in Figure 6(a). Then by Lemma 2.8, this is the same as Figure 6(b). Then we have $i^{4n-4} = 1$ times Figure 6(c). So the positive twist in the arc near the S-box evaluates the same as no twist. Since we have the Reidemeister 2 and 3, any positive twist may be moved next to the S-box. Note that left hand $R1$ follows from right hand. Thus we have all the Reidemeister moves.

Lemma 2.14 *Two applications of the algorithm using different arcs for the S -boxes give the same answers.*

Proof It is enough to show that changing the arc of one S-box does not change the result of the algorithm. By Lemma 2.12, we may assume this S-box is last to travel along its S-box arc. The arc represents what will be $4n-4$ strands under every passing strand in the diagram. By Lemma 2.13 we have all three Reidemeister moves for the

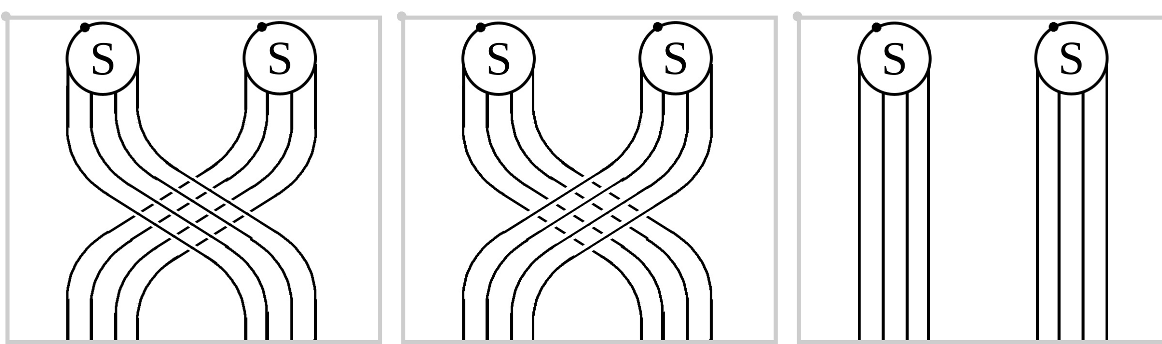


(a) (b) (c)

Figure 6: Reidemeister 1 for the S-box arc

arcs so long as they don't cross an S-box. Since the S-box is the last to go, we may move the arc by Lemma 2.8 so long as we don't change the position of the S-box at the top of the diagram.

To see that we can also move the S-box positions at the top of the diagram, we again need Lemma 2.11. Consider the diagrams in the lemma directly under two S-boxes at the top of the diagram. This lemma allows us to swap the positions of two S-boxes along the top of the diagram by replacing Figure 7(c) with either Figure 7(a) or Figure 7(b). Choosing the correct crossing preserves the order of the S-box arcs. Thus making the replacement is the same as moving the arc to a different position. Then we have that the evaluation will be the same.



(a) (b) (c)

Figure 7: Switching the S-boxes does not change the evaluation result.

2.2.3 Algorithm Respects the Relations

We show that changing a closed diagram by one of the relations given in Definition 1.3 does not change the result of the evaluation algorithm.

Relation 1 Since the algorithm is well-defined we may choose to run the algorithm so that the S-box arcs avoid the chosen circle. This choice works for the diagram with the circle and for the one with the circle removed and replaced with $[2]_q$ times the empty diagram. Then when we remove each S-box the diagram will only differ by the circle removed and the $[2]_q$ times the empty diagram. This is equal in Temperley-Lieb, so when we complete the algorithm, the results will be the same.

Relation 2 Consider running the algorithm on both diagrams. For each, make the same choices for Part I of the algorithm. Once we get to Part II of the algorithm make the same choices as well. The S-box in question will either be a pair of S-boxes removed or it will be the last S-box remaining. If the later is the case, we get zero for both diagrams. Otherwise, we have two diagrams with our S-box connected to another S-box by at least $2n-2$ strands. The difference is that in one diagram the S-box has been twisted clockwise and a factor of i introduced.

Now chose in both diagrams the same $2n-2$ strands (ignoring for now the marked point). Thus once we replace the S^2 with an f^{2n-2} the diagrams will be the same, and the coefficients are all that need to be checked. Then count clockwise on our S-box and counterclockwise on the other. In the first diagram, we introduce a factor of i^{k-l}, while in the twisted diagram we introduce a factor of $i^{(k-1)-l} = i^{-1}(i^{k-l})$. Thus since this diagram had an additional factor of i, they cancel and both diagrams are i^{k-l} times the same diagram. Thus completing the algorithm will yield the same results.

Relation 3 This is Lemma 2.9.

Relation 4 Consider the diagram with the S^2 as opposed to the f^{2n-2}. Choose two arcs that attach to the marked points of these two S-boxes. Note that for the S^2, the marked points are on the same side of the $2n-2$ strands connecting the two. Draw the arcs next to each other, not crossing and not enclosing another S-box. Have the arcs start from the marked point of each S-box, traveling to the top of the diagram while avoiding the $2n-2$ connecting the S-boxes together.

For the remaining S-boxes in the diagram, draw arcs that avoid our two special S-boxes, the strands connecting them, and their two S-box arcs. This is possible since all can be enclosed by a region separate from the other S-boxes leaving many paths to the top of the diagram.

Once all the S-boxes are at the top of the diagram, we may move the $2n-2$ strands connecting the two special S-boxes to the top of the diagram following the paths that the two S-boxes took. The S-box arcs are now $4n-4$ strands under every strand that the arcs crossed. Thus we may use Lemma 2.8 to drag the $2n-2$ connecting strands to the top of the diagram with no crossings between the S-boxes.

Then we may choose to evaluate this pair first, using the $2n-2$ strands. This yields an f^{2n-2} in place of the S^2. Now the diagrams are equivalent, so the relation holds.

References

[Big09] Stephen Bigelow, *Skein Theory for the ADE Planar Algebras*. 2009.

[BMPS09] Bigelow, Morrison, Peters and Snyder, *Constructing the extended Haagerup planar algebra*. 2009.

[MPS08] Morrison, Peters and Snyder, *Skein theory for the D_{2n} planar algebras*. 2008.