5-1-2008

Representations of the Temperley-Lieb Algebra

Anne Moore
Macalester College

Honors Paper

Macalester College

Spring 2008

Title: Representations of the Temperley-Lieb Algebra

Author: Anne Moore

SUBMISSION OF HONORS PROJECTS

Please read this document carefully before signing. If you have questions about any of these permissions, please contact Janet Sietmann in the Library.

Title of Honors Project: Representations of the Temperley-Lieb Algebra

Author's Name: (Last name, first name) Moore, Anne

The library provides access to your Honors Project in several ways:

- The library makes each Honors Project available to members of the Macalester College community and the general public on site during regular library hours.
- Using the latest technology, we make preservation copies of each Honors Project in both digital and microfilm formats.
- Every Honors Project is cataloged and recorded in CLICnet (library consortium OPAC) and in OCLC, the largest bibliographic database in the world.
- **To better serve the scholarly community,** a digital copy of your Honors Project will be made available via the Digital Commons @ Macalester (digitalcommons.macalester.edu).

The DigitalCommons@Macalester is our web based institutional repository for digital content produced by Macalester faculty, students, and staff. By placing your projects in the Digital Commons, all materials are searchable via Google Scholar and other search engines. Materials that are located in the Digital Commons are freely accessible to the world; however, your copyright protects against unauthorized use of the content. Although you have certain rights and privileges with your copyright, there are also responsibilities. Please review the following statements and identify that you have read them by signing below. Some departments may choose to protect the work of their Honors students because of continuing research. In these cases the project is still posted on the repository, but content can only be accessed by individuals who are located on campus.

The original signed copy of this form will be bound with the print copy of the Honors Project. The microfilm copy will also include a copy of this form. Notice that this form exists will be included in the Digital Commons version.

I **agree** to make my Honors Project available to the Macalester College community and to the larger scholarly community via the Digital Commons@Macalester or its successor technology.

Signed Anne E. Moore

OR

I **do not want** my Honors Project available to the larger scholarly community. I want my Honors Project available only in the library, **NOT** for interlibrary loan purposes, and **NOT** through the Macalester College Digital Commons or its successor technology.

Signed ____________________

NOTICE OF ORIGINAL WORK AND USE OF COPYRIGHT PROTECTED MATERIALS:

If your work includes images that are not original works by you, you must include permissions from original content provider or the images will not be included in the electronic copy. If your work includes discs with music, data sets, or other accompanying material that is not original work by you, the same copyright stipulations apply. If your work includes interviews, you must include a statement that you have the permission from the interviewees to make their interviews public.

BY SIGNING THIS FORM, I ACKNOWLEDGE THAT ALL WORK CONTAINED IN THIS PAPER IS ORIGINAL WORK BY ME OR INCLUDES APPROPRIATE CITATIONS AND/OR PERMISSIONS WHEN CITING OR INCLUDING EXCERPTS OF WORK(S) BY OTHERS.

All students must sign here.

Signature: Anne E. Moore

Date: May 6, 2008

Printed Name: Anne E. Moore

G:/libstaff/digitalcommons/dccopyrightpermissionrev.doc

Representations of the Temperley-Lieb Algebra

Anne Moore

Tom Halverson, First reader

Dan Flath, Second reader

Dick Molnar, Third reader

April, 2008

MACALESTER COLLEGE

Department of Mathematics

Abstract

This paper gives an introduction to Temperley-Lieb algebra that is easily accessible to undergraduates, presenting *TL* diagrams, the method for multiplying the diagrams, and the properties of the multiplication that it is necessary to preserve in a representation. The paper also gives a method for finding representations of the *TL* monoids (sets of diagrams classified by number of vertices) using Young tableaux, and shows that these representations are all of the irreducible representations. While ideas of Hecke algebra imply the fact that this method produces representations, this paper provides a direct proof, strictly within the field of representation theory. It also introduces some conclusions about the rank of a diagram and its action on the tableaux.

Contents

Acknowledgments

I wish to thank my adviser Professor Tom Halverson for all of his help at every stage of this project, including the formation of its author as a mathematician. He has been my advisor since I declared my major and taught three of my courses. Through this project, he shared his own research interests with me.

I also wish to thank Professor Dan Flath, who has always been more than willing to talk with me about my project, about life, about math, and about life as a mathematician. His down-to-earth answers, enthusiasm for math, and above all the confidence in me that he projects, make him an incredible mentor.

This capstone paper was produced using the Harvey Mudd College thesis template package in LaTex, created by Clare M. Connelly.

Chapter 1

Introduction

In 1971 Harold Temperley and Elliot Lieb published a thirty page article, *Relations between the 'Percolation' and 'Colouring' Problem and Other Graph-Theoretical Problems Associated with Regular Planar Lattices: Some Exact Results for the 'Percolation' Problem* [7], in which appeared what was later named the Temperley-Lieb algebra. As both were mathematical physicists, their interest lay in using the algebra for applied mathematical endeavors, such as finding results in statistical mechanics. The Temperley-Lieb algebra has enjoyed a wide range of uses in the applied sciences. Its appeal to mathematicians, particularly to those interested in algebra and combinatorics, arises from its nice generators and generator relations and its connection to the ubiquitous Catalan sequence.

In this paper, we present a method for finding matrix representations of the Temperley Lieb algebra using combinatorial objects called Young tableaux. This method was inspired by the representation theory of the Iwahori-Hecke algebras, which are quantum generalizations of the symmetric group. Like the symmetric group, the Iwahori-Hecke algebras have representations that are indexed by partitions, spanning modules whose bases are indexed by Young tableaux. There is also a homomorphism [3] from the Iwahori-Hecke algebras to the Temperley-Lieb algebras (Section 4.2, which we use to adapt the representation theory of the Iwahori-Hecke algebras to the representation theory of the Temperley-Lieb algebra. By looking at the "seminormal representations" of a Iwahori-Hecke algebra at the level of Temperley-Lieb algebra, we come up with our method for finding representations of Temperley-Lieb algebra.

After introducing our method, we prove its successfulness directly without using results about the Hecke algebras. We show that the resulting matrices are in fact representations of the Temperley-Lieb algebra, that the representations are distinct from each other, that they are irreducible, and that we can find every irreducible representation using this method.

It should be noted that these representations are isomorphic to representations found using different bases for the underlying vector spaces. The representations found from our method are the "seminormal" representations, a term dating back to when Alfred Young used Young tableaux to index the underlying vector spaces for symmetric group representations.

Chapter 2

Temperley-Lieb Diagrams

2.1 What the diagrams look like

Temperley-Lieb diagrams are graphs composed of two rows of n dots and the edges that connect each dot to exactly one other dot. TL_n refers specifically to those diagrams that have n dots in each row. The two rows are aligned so that one row is directly above the other. A matching can be between two dots in the same row or one dot from each row. In TL_n diagrams, edges cannot cross, nor can the fall below the bottom row of dots or rise above the top row. In other words, the edges cannot leave the rectangle defined by the two rows of dots. Figure 2.1 illustrates all of the diagrams in TL_3 and Figure 2.1 illustrates diagrams on six vertices that are not in TL_3.

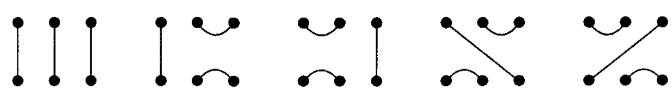


Figure 2.1: Diagrams of TL_3

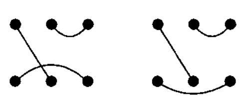


Figure 2.2: Not TL_3 diagrams

Figure 2.3: Diagram of TL_1

Figure 2.4: Diagrams of TL_2

2.2 The number of diagrams

For each TL_n, there are a finite number $|TL_n|$ of diagrams. The most natural way to find the number of diagrams in TL_n is recursively. Begin with the two rows of dots. There are n choices for which dot to connect to the first dot in the top row. To see this, consider labeling the dots in a clockwise direction.

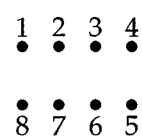
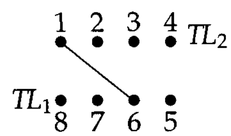


Figure 2.5: Labeled dots of TL_4

Odd numbered dots can only connect to even numbered dots and vice versa. This is due to that fact that each edge partitions that diagram into two parts. If an odd point connected to another odd point, then there would be an odd number of dots on each side of the edge. Pairing off the remaining dots would mean that at least one pairing would have to be between dots on both sides of the original edge, meaning that the edges would cross.

There are n choices for connecting point 1 to an *even* point, and this edge partitions the diagram into two parts. If there are k dots on one side of the edge, then there are $2n-2-k$ dots on the other side. The number of diagrams in TL_n with this edge is $|TL_{k/2}| * |TL_{(2n-2-k)/2}|$.



The labeled dots above show the skeleton for TL_4. To find $|TL_4|$ recursively, it is necessary to know the number of diagrams in TL_n where $n < 4$. There are five diagrams in TL_3, which are listed above. There are two diagrams in TL_2, one where dots that are in the same row are connected, and one where dots aligned vertically are connected. There is only one diagram in TL_1, connecting the only two dots. To make the recursive property work properly, set $|TL_0| = 1$.

$$\begin{aligned} |TL_4| &= |TL_0||TL_3| + |TL_1||TL_2| + |TL_2||TL_1| + |TL_0||TL_3| \\ &= 1 * 5 + 1 * 2 + 2 * 1 + 5 * 1 = 14 \end{aligned} \tag{2.1}$$

This recurrence relation is familiar from combinatorics as the *Catalan* numbers. Equation 2.2 gives a well known non-recursive way of finding the Catalan numbers. $|TL_n| = C_n$.

$$C_n = \frac{(2n)!}{(n+1)!n!} \tag{2.2}$$

Knowing the number of diagrams in TL_n becomes useful later on for demonstrating the irreducibility of the representations that are the main subject of this paper. For now, it gives us a way to check that we have found all of the diagrams in TL_n.

2.3 Multiplying the diagrams

Diagrams in TL_n can be multiplied by stacking the diagrams vertically and following the edges. The paths should begin at either the top row of the top diagram or the bottom row of the bottom diagram. The rows in the middle meet up so that when a path leads to the ith dot in one of the middle rows, the path continues from the ith dot in the other of the middle rows.

As seen in the example of multiplication in TL_4, there are sometimes loops that are not part of a path beginning at either the top or the bottom. These loops do not become part of the product diagram, although there will be a way of counting the dropped loops in the TL_n algebra.

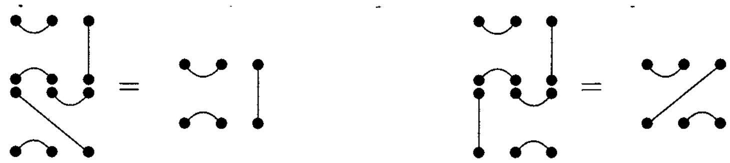


Figure 2.6: Multiplication in TL_3

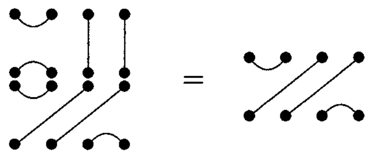


Figure 2.7: Multiplication in TL_4

Multiplication in TL_n is associative. Multiplying a string of diagrams is equivalent to stacking a sequence of diagrams and following paths beginning at the top row of the top diagram and the bottom row of the bottom diagram. However, multiplication in TL_n is generally not commutative.

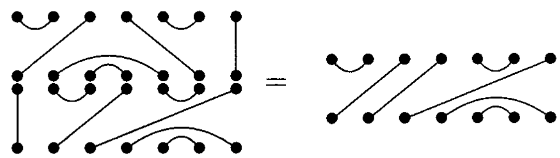


Figure 2.8: Multiplication in TL_7

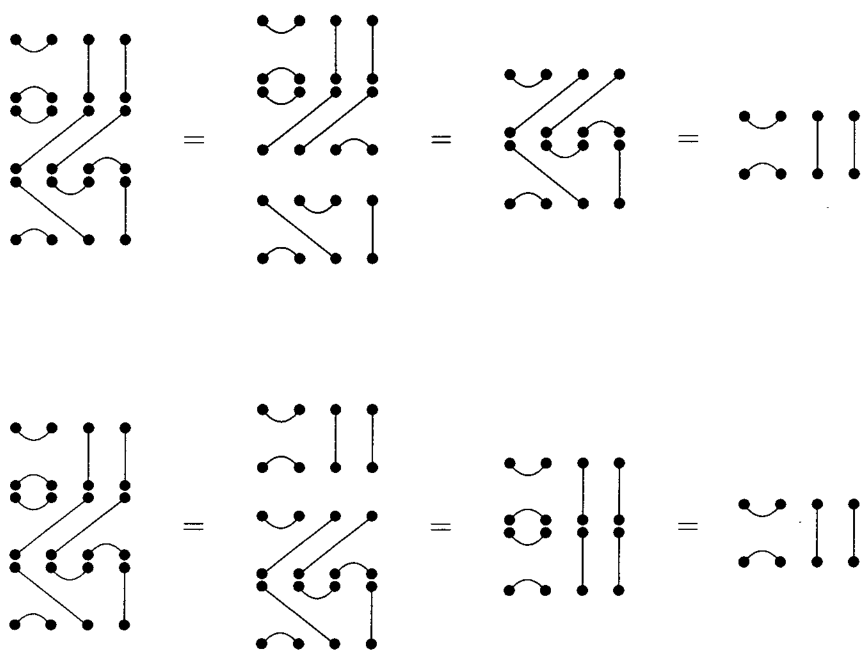


Figure 2.9: Multiplication of diagrams is associative

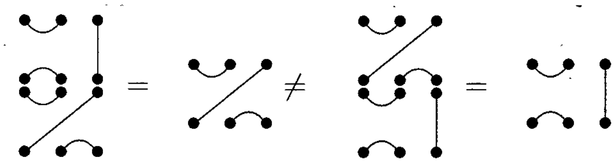


Figure 2.10: Multiplication of diagrams is not commutative

2.4 Identity, rank, and lack of inverses

There is an identity diagram in each TL_n. This diagram connects every dot in the top row with the dot directly under it in the bottom row. The

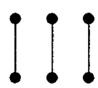


Figure 2.11: Identity Diagram in TL_3

identity diagram has the same properties as the identity element in any group algebra. If A is any diagram in TL_n and I is the identity diagram in TL_n, then $AI = IA = A$.

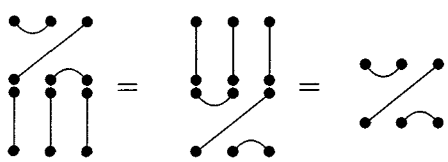


Figure 2.12: Multiplying by the identity diagram

The identity diagram is also significant because of its rank. The rank of a Temperley-Lieb diagram is the number of edges that connect the top row with the bottom row. The identity diagram is the only diagram in TL_n with rank n. When multiplying diagrams, the product diagram cannot have a rank that is any larger than the minimum rank of the multiplied diagrams. If the rank of a TL_n diagram is k, then that leaves $n - k$ dots in the top row that are connected to other dots in the top row, and equivalently leaves $n - k$ dots in the bottom row that are connected to other dots in the bottom row. Suppose this diagram is the first in a sequence of multiplied diagrams.

Then the connections among the $n - k$ remaining dots in the top row move directly into the product diagram. Likewise, if this diagram is the last in a sequence of multiplied diagrams, then the connections among the $n - k$ remaining dots in the bottom row move directly to the product diagram.

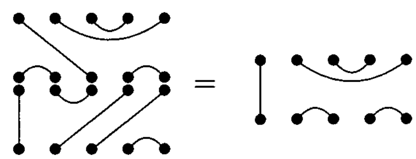


Figure 2.13: Multiplication and rank

The diagrams multiplied in figure 2.13 are of rank one and three, and the product diagram is of rank one. In order to better see the relationship between the rank of the multiplied diagrams and the rank of the product, consider just those edges that were moved directly from the multiplied diagrams to the product diagram.

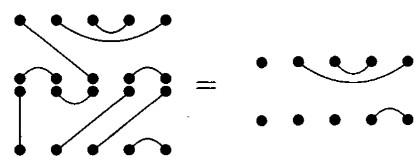


Figure 2.14: Direct edges only

The edges in the top row of the product in figure 2.14 were moved directly from the first diagram, and the edges in the bottom row of the product were moved directly from the second diagram. This leaves exactly one dot remaining in the top row which could connect to the bottom row.

The rank of the product diagram is not necessarily equal to the minimum rank of the multiplied diagrams. The rank of the product can be less than either rank of the multiplied diagrams. Figure 2.15 shows how this is possible.

The examples and arguments in this section have only shown that the rank cannot increase when two diagrams are multiplied, but the associative property makes it easy to extend this proof to cover multiplication of any number of diagrams. Multiplying the first two diagrams in a string yields a diagram of rank less than or equal to the minimum rank of the first two diagrams. Multiplying the result by the third diagram cannot yield a result

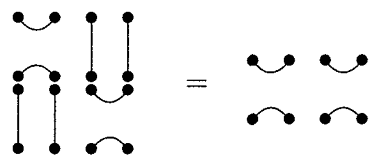


Figure 2.15: Rank decreases

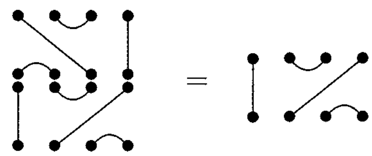


Figure 2.16: Rank stays the same

with a higher rank. As each diagram is multiplied in, the resulting rank stays constant or decreases.

The idea of rank provides a straightforward proof that no non-identity diagram has an inverse. Let A be a non-identity diagram in TL_n. Because it is not the identity, the rank of A is less than n. If A had an inverse A^{-1}, then $AA^{-1} = I$ is the identity diagram. However, the rank of AA^{-1} has be be less than or equal to the rank of A, so the rank of $AA^{-1} < n$. Thus $AA^{-1} \neq I$.

2.5 Generator diagrams

Aside from the identity diagram, each diagram in TL_n can be made by multiplying a string of *generator diagrams* of TL_n. There are $n-1$ generators in TL_n, labeled $e_1, e_2, \ldots, e_{n-1}$. The diagram e_i has an edge connecting the ith and $i+1$th dots in the top row and an edge connecting the ith and $i+1$th dots in the bottom row. All other edges connect dots in the top row to the dots directly under them in the bottom row.

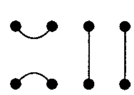
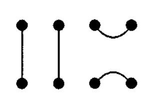
$e_1 =$  $e_2 =$  $e_3 =$ 

Figure 2.17: Generator Diagrams of TL_4

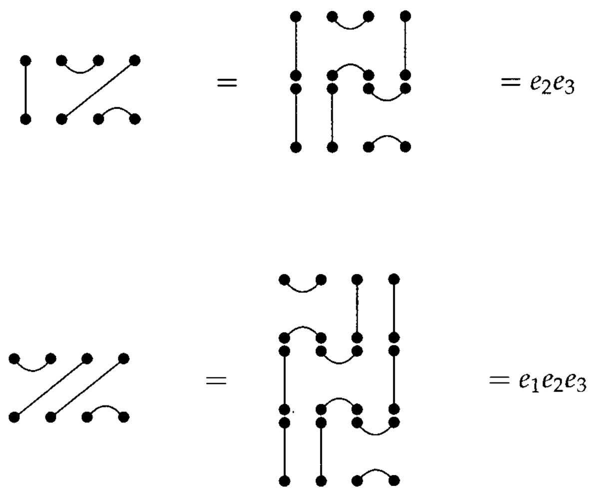


Figure 2.18: Generating diagrams

Figure 2.18 shows how two of the diagrams of TL_4 can be achieved by multiplying generator diagrams. In order to prove that all diagrams can be products of generator diagrams, we first show how to get a certain type of diagram by multiplying generator diagrams, and then provide an algorithm for multiplying diagrams of this type to get any given diagram.

The intermediate type of diagram has connections between adjacent dots in the top row if and only if the adjacent points directly below these two dots are also connected. Any dot that is not connected to an adjacent dot in the same row is connected to the dot directly above or below it.

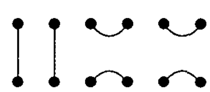


Figure 2.19: Intermediate-type diagram

The diagram in Figure 2.19 equals e_3e_5. In general, intermediate-type diagram equals $\prod_{i=1}^{n-1} e_i$ where the ith and $i+1$th dots in the same row are connected. Thus any diagram that can be made by multiplying intermediate-type diagrams can be made by multiplying generator diagrams. The following step is to show that any diagram can be made up of intermediate-type diagrams.

Try to make the following diagram in figure 2.20.

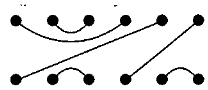
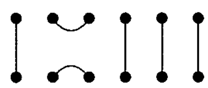
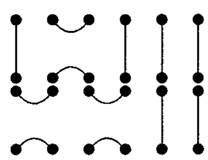


Figure 2.20: Make this figure

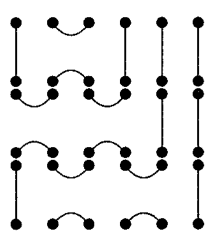
Begin by addressing only those edges exclusively in the top row. The edge from the second to the third dot is nestled in the edge from the first dot to the fourth dot. We cannot deal with the edge from the first dot to the fourth dot until we deal with the edge nestled inside it. Begin with e_2.



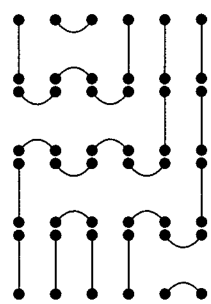
Now address the connection between the first and fourth dots. There is already a bridge between the the second and third, and that bridge has to be extended to the first and fourth.



Next address the edges connecting the top and bottom rows. There are places where the edge connecting the fifth dot in the top row and the first dot in the bottom row is above the edge connecting the sixth dot in the top row and the fourth dot in the bottom row. The top row has to be addressed first. A bridge has to be built between the fifth and first dots.



Finally, a bridge has to be made between the fourth and sixth dots. The bridge already exists between the fourth and fifth dots, so it just has to be extended.



The trick is to start with connections at the top and work downward.

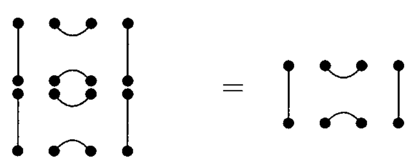
2.6 Generator relations

There are three particular relations that the generators have with each other.

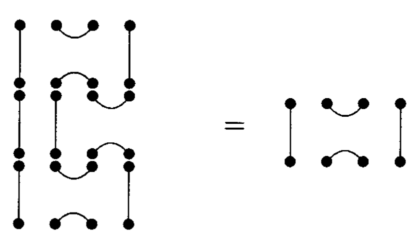
- $e_i e_i = e_i$ with one loop dropped
- $e_i e_{i+1} e_i = e_i$
- $e_i e_j = e_j e_i$ if $|i - j| \geq 2$

Each can be proved with an example, since extending the proof to other instances merely requires adding extra vertical edges as are in the identity diagram. These edges do not change anything about the outcome.

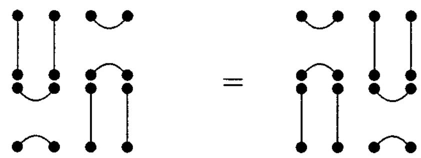
First Relation
$e_i e_i = e_i$ with one dropped loop



Second Relation
$e_i e_{i+1} e_i = e_i$



Third Relation
$e_ie_j = e_je_i$ when $|i-j| \geq 2$



Chapter 3

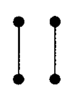
The $TL_n(x)$ Algebra

For $n \in \mathbb{Z}_{>0}$ and $x \in \mathbb{C}$ define the Temperley-Lieb algebra $TL_n(x)$ to be the vector space whose basis is given by the diagrams in TL_n. Thus

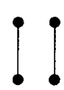
$$TL_n(x) = \mathbb{C}\text{-span}\,\{ \; d \mid d \in TL_n \; \} = \left\{ \sum_{d \in TL_n} \alpha_d d \mid \alpha_d \in \mathbb{C} \right\}. \tag{3.1}$$

The parameter $x \in \mathbb{C}$ needs to be chosen to avoid certain bad values for which the representations do not work. For example, you can see from the formulas that we derive in Section 5, that x cannot be a root of the polynomials $[d \pm 1]/[d]$ and $[d-1][d+1]/[d]$ defined there. It is know that these bad values occur when x is a root of unity (see for example [8]) and we will assume that our x is always chosen to avoid these situations.

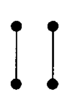
In the Temperley-Lieb algebra, the diagrams in TL_n are the basis for a vector space. For instance, the vector space of $TL_2(x)$ is

span{  ,  }

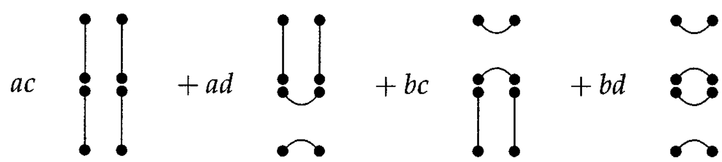
Let the following be one vector in the vector space

a  $+ b$ 

And let

c  $+ d$ 

be another vector in the vector space. To be an algebra, there has to be a defined multiplication of vectors. Multiplying the first of the above vectors by the second yields

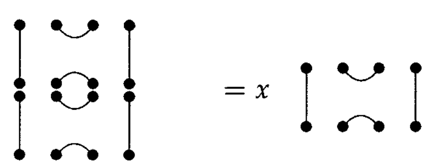


The first three of these terms are multiplied in exactly the same way as the diagrams in the TL_2 monoid, but the fourth term has a dropped loop. While these were simply dropped and forgotten in the monoid, the algebra keeps track of them.

3.1 Dropped loops in the algebra

The multiplication in Temperley-Lieb Algebra is practically the same as the multiplication of the diagrams, except that it counts the number of loops dropped when the diagrams are multiplied. To do this, it introduces the variable x, and every time a loop is dropped, the resulting product is multiplied by x. Thus, if two loops are dropped in the course of a multiplication, the resulting product diagram is multiplied by x^2. The exponent of the x tells the number of loops dropped.

The most important change brought about by the x is the product of e_ie_i. Whereas with the diagrams, the result was e_i again, having dropped one loop, the new relation reads $e_ie_i = xe_i$.



The other two relations, $e_ie_{i+1}e_i = e_i$, and $e_ie_j = e_je_i$ when $|i-j| \leq 2$, remain the same, as no loops were dropped out during these multiplications.

It is easy to see that exactly one loop was dropped out during the multiplication of e_i with itself, but the number of loops dropped in other multiplications can be more obscure. The first question is whether two loops can be dropped out at the same time. It seems fairly clear that the answer to this is yes, but it takes some experimentation to be sure. Figure 3.1 shows a test case.

It is possible to verify that both of these dropped loops appear as x's in the product by rewriting the expression in terms of generator diagrams.

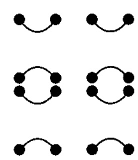
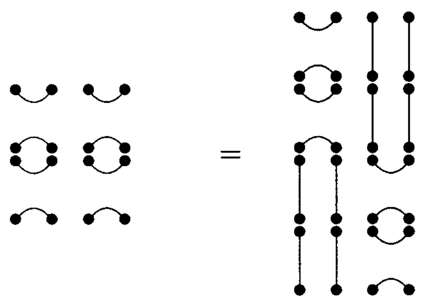


Figure 3.1: Two loops dropped

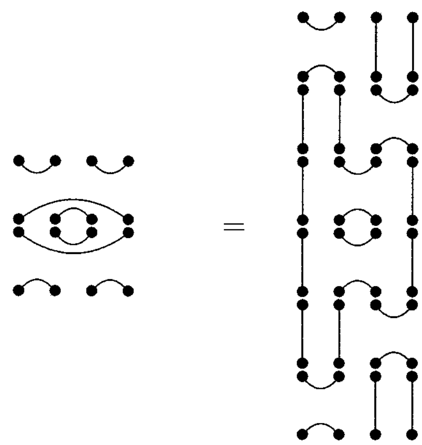


$$= (e_1e_1)(e_3e_3) = (xe_1)(xe_3) = x^2e_1e_3$$

A more complicated situation arises when one of the loops fits inside the other. Again, breaking this problem down to its generator components gives a definite answer about how many x's are in the product.

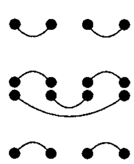
Using the generator relations, it is possible to simplify the equation in Figure 3.2, while at the same time picking out the x's. First, reorganize the expression by associativity into $e_1e_3(e_2e_2)e_3e_1$. We know from generator relation 1 that $e_2e_2 = xe_2$, and because x represents a constant, it can be pulled out front, giving $xe_1e_3e_2e_3e_1$. Generator relation 2 says that $e_ie_{i+1}e_i = e_i$, and similarly, $e_{i+1}e_ie_{i+1} = e_{i+1}$, the picture from relation 2 just being flipped horizontally. Thus, the expression further simplifies to $xe_1e_3e_1$. Then, because $|3-1| \geq 2$, this can be rewritten as $xe_1e_1e_3 = x^2e_1e_3$. Thus, even though the loops are nested, there are still two loops which are pulled out, and two x's in the product.

One case remains, where more than two edges make up the loop. To show that an x is pulled out in this case, find the generators and manipulate them algebraically.



$= (e_1e_3e_2)(e_2e_3e_1)$

Figure 3.2: Nested loops



$= (e_1e_3)(e_2e_3e_1) = e_1(e_3e_2e_3)e_1 = e_1e_3e_1 = e_1e_1e_3 = xe_1e_3$

Figure 3.3: Multi-edge loop

Chapter 4

Representations of the $TL_n(x)$ Algebra

4.1 Some facts about representations

The most familiar structure from abstract algebra is a *group*. A group is a set of elements that have a binary operation such that the set is closed under the operation, there is an identity element, and every element a has an inverse a^{-1} in the set such that $aa^{-1} = a^{-1}a = 1$. Call the operation $*$. Then, if a and b are both elements of the group, $a * b$ is an element of the group. The order of the elements is important, since $a * b$ may not equal $b * a$. If $a * b = b * a$ for all $a, b \in G$, where G is the group, then the group is called *Abelian*. Consider the set of diagrams in TL_n. There is a binary operation for these diagrams, namely the diagram multiplication just described. The set of diagrams in TL_n is similar to a group. However, every element in a group must have an inverse in the group, and we have already shown that none of the non-identity elements in TL_n have inverses. Therefore, the set of diagrams in TL_n is not a group but a *monoid*.

Like groups, monoids can be represented by matrices with elements from a field. For the sake of this paper, the field will functions of the variable x. A matrix representation is the assignment of matrices to every element in the group or monoid, such that if matrix A is assigned to element a and matrix B is assigned to element b, and $ab = c$, then $AB = C$ where C is the matrix assigned to c. It is possible for several elements to be assigned the same matrix. For example, the "trivial representation" sends all of the elements to 1. A correspondence that assigns a different matrix to each element is called a "faithful representation."

Consider the generator diagrams of TL_n. If these diagrams have matrices assigned to them, then the matrices are known for every diagram in the monoid. To find the matrix for diagram a, simply multiply the matrices corresponding to the generator diagrams in the same order as the diagrams are multiplied to get a. It is not possible to simply randomly assign matrices to the generator diagrams and find a representation. The generator matrices have to have the same relations as the generator diagrams. Let E_i be the matrix assigned to e_i. Then $E_iE_i = xE_i$, $E_iE_{i+1}E_i = E_i$, and $E_iE_j = E_jE_i$ when i and j are far enough apart.

Each representation has a corresponding vector space called a *module*. Suppose that $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_t$ are the basis vectors of this vector space. Each element in the monoid acts on the vector space, sending each vector to a different vector. For example, $a \cdot \mathbf{v}_1 = \mathbf{v}_3$, or $a \cdot (\mathbf{v}_2 - \mathbf{v}_1) = \mathbf{v}_4$. The action is linear, so $a \cdot \mathbf{v}_2$ would equal $\mathbf{v}_3 + \mathbf{v}_4$. Because the action is linear, the action of an element on any vector in the vector space is completely determined by how it acts on the basis vectors.

In order to find the representations from the modules, assign each of the basis vectors to the columns and rows of a $t \times t$ matrix. For example, $\mathbf{v}_i$ corresponds to column i and row i. Let $a \cdot \mathbf{v}_i = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \ldots + c_t\mathbf{v}_t$. Then the c's are the entries in the ith column. Suppose $a \cdot \mathbf{v}_1 = \mathbf{v}_3$. Then the first column in the matrix corresponding to a would be all zeros except for a one in the third row.

Multiplication of elements in the monoid is preserved by actions on vectors in the module.

$$ab \cdot \mathbf{v}_i = a \cdot (b \cdot \mathbf{v}_i) \tag{4.1}$$

This is a very useful property. For example, in order to check that $E_iE_j = E_jE_i$ for i and j far enough apart, we will be able to check that $e_i \cdot (e_j \cdot \mathbf{v}) = e_j \cdot (e_i \cdot \mathbf{v})$ for all $\mathbf{v}$ in the module.

4.2 Connection to the Hecke algebra

Our method for finding representations of the Temperley-Lieb algebra involves combinatorial objects called Young tableaux, which Alfred Young used to find representations of the symmetric group. The Iwahori-Hecke algebra $H_n(q)$, a quantum generalization (sometimes called a q-generalization) of the symmetric group, also has representations indexed by Young tableaux. The connection between $H_n(q)$ and $TL_n(x)$ was first noted in 1987 by V. F. R. Jones [4]. There is a surjective algebra homomorphism from $H_n(q)$ to $TL_n(x)$, which we use to "push" representations of $H_n(q)$ down to $TL_n(x)$.

The parameter x of the Temperley-Lieb algebra is related to the parameter q of the Hecke algebra by $x = q + q^{-1}$. We follow the presentation of this relation in [3].

The Hecke algebra $H_n(q)$ is defined by generators $1, T_i, T_2, \ldots, T_{n-1}$ and relations

$$\begin{aligned} T_i^2 &= (q - q^{-1})T_i + 1 \\ T_iT_j &= T_jT_i \qquad \text{for } |i-j| > 1 \\ T_iT_{i+1}T_i &= T_{i+1}T_iT_{i+1} \end{aligned} \tag{4.2}$$

The variable $q \in \mathbb{C}$ is "a fixed element in the base ring" $\mathbb{C}$.

The homomorphism from $H_n(q)$ to $TL_n(x)$ (found, for example, in the paper by Halverson, Mazzocco, and Ram [3]), sends T_i to $q - e_i$. We can check that the Temperley-Lieb relations hold under this identification:

$$\begin{aligned} e_i^2 &= (q - T_i)^2 = q^2 - 2qT_i + T_i^2 \\ &= q^2 - 2qT_i + (q - q^{-1})T_i + 1 \\ &= q^2 - 2q(q - e_i) + (q - q^{-1})(q - e_i) + 1 \\ &= q^2 - 2q^2 + 2qe_i + q^2 - qe_i - 1 + q^{-1}e_i + 1 \\ &= qe_i + q^{-1}e_i \\ &= (q + q^{-1})e_i \end{aligned}$$

Since we know that $e_i^2 = xe_i$, we can see from this that under this identification, we must have $q + q^{-1} = x$. This identity for q becomes important for finding representations of Temperley-Lieb algebra.

The representations of $H_n(q)$ are indexed by partitions of n. When passing to $TL_n(x)$, the representations indexed by partitions of more than two rows are in the kernel of the surjection $H_n(q) \rightarrow TL_n(x)$, but the representations on two rows survive (see [4] or [3]). The so called "seminormal representations" of the Hecke algebra are described in [3], and more about seminormal representations can be found in [6]. They give an action of the elements of $H_n(q)$ on a basis that is indexed by standard Young tableaux of partition shape. In what follows, we have translated this action to an action of the e_i on standard tableaux on two rows. We directly prove that this gives a representation of $TL_n(x)$, without referring back to the Hecke algebra, and we show that these representations form a complete set of pairwise non-isomorphic irreducible representations of $TL_n(x)$.

4.3 The Young tableaux vector space

The first step of finding representations for $TL_n(x)$ is to find an appropriate vector space to act on. The second step is the define an action of the TL_n

diagrams on on vectors in this vector space. Then it is just a matter of showing that the relations of $TL_n(x)$ are preserved in the representation. In this case, the TL_n diagrams act on a vector space where the basis vectors are Young tableaux.

4.3.1 Young diagrams

Young diagrams are left-justified rows of boxes where the number of boxes in each row is less than or equal to the number of boxes in the row above it. Figure 4.1 shows a Young diagram. Notice that each row begins in the same place and that no row has more boxes than the row above.

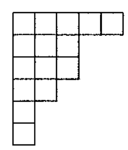


Figure 4.1: A Young diagram

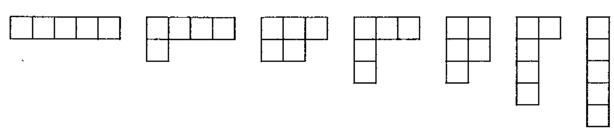
Young diagrams correspond to partitions of whole numbers. A partition of a number n is the sequence $(\lambda_1, \lambda_2, \ldots, \lambda_\ell)$ such that

$$\lambda_1 + \lambda_2 + \cdots + \lambda_\ell = n \quad \text{and} \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_\ell \tag{4.3}$$

For example the partitions of 5 are

$$(5), (4,1), (3,2), (3,1,1), (2,2,1), (2,1,1,1), (1,1,1,1,1)$$

and the Young diagrams of size five are



The Young diagram in Figure 4.1 corresponds to the partition $(5,3,3,2,1,1)$ of 15. The partition is also called the *shape* of the diagram. The length of a Young diagram is the number of rows, so the diagram in Figure 4.1 has length six. The Young diagrams that help in finding the representations for $TL_n(x)$ are those with length one or two. Figure 4.2 shows the Young diagrams of lengths one and two for n up to six.

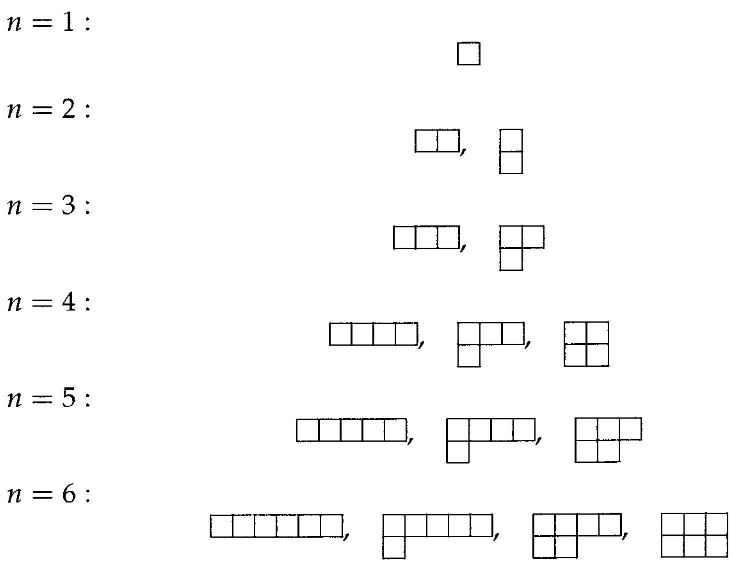


Figure 4.2: Diagrams of length ≤ 2

4.3.2 Young tableaux

In Young tableaux, the integers 1 through n are placed in the boxes. Each integer is placed exactly once. For a Young tableau to be *standard*, the numbers have to be placed so that the numbers are in ascending order from left to right and from top to bottom. In other words, no integer can be to the right of or underneath a larger integer. The diagram in Figure 4.3 is standard. The diagram in Figure 4.4 is not standard because the 2 is underneath the 4, contradicting the rule that the numbers go in ascending order from top to bottom.

1	2	5
3	4	

Figure 4.3: Standard

Diagrams in Figures 4.3 and 4.4 are of size $n = 5$ and shape (3,2).

There are a finite number of standard tableaux that can fit into a diagram of a given shape. There is a formula for finding this number making use of *hook lengths* of the boxes.

Definition: The *hook length* of a box in a Young diagram is the number

1	4	5
3	2	

Figure 4.4: Not standard

of boxes under the given box, added to the number of boxes to the right of the given box, plus one.

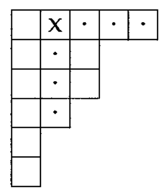


Figure 4.5: Box x has hook length 7

The number of tableaux is $n!/(\prod h)$ where the h's are the hook lengths for each box.

4	3	1
2	1	

Figure 4.6: Hook lengths of boxes in (3,2) diagram

Figure 4.6 is not a tableau. The numbers in the boxes are each box's hook length. Using the formula and Figure 4.6, the number of Young tableaux of shape (3,2) equals $5!/(4 \cdot 3 \cdot 1 \cdot 2 \cdot 1) = 5!/24 = 5$.

4.3.3 Vector spaces

A set of tableaux of size n of the same shape index a vector space which will be shown to be the module of a representation for $TL_n(x)$. For example, each of the tableaux of shape (3,1) represents a basis vector of a three-dimensional vector space, since there are three tableaux of shape (3,1). Every vector in this space is of the form

$$\mathbf{v} = a \begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline 4 \\ \cline{1-1}\end{array} + b \begin{array}{|c|c|c|}\hline 1 & 2 & 4 \\ \hline 3 \\ \cline{1-1}\end{array} + c \begin{array}{|c|c|c|}\hline 1 & 3 & 4 \\ \hline 2 \\ \cline{1-1}\end{array}$$

where $a, b,$ and c are constants.

$$\begin{array}{|c|c|c|}\hline 1&2&3\\\hline 4&5\\\cline{1-2}\end{array},\quad \begin{array}{|c|c|c|}\hline 1&2&4\\\hline 3&5\\\cline{1-2}\end{array},\quad \begin{array}{|c|c|c|}\hline 1&2&5\\\hline 3&4\\\cline{1-2}\end{array},\quad \begin{array}{|c|c|c|}\hline 1&3&4\\\hline 2&5\\\cline{1-2}\end{array},\quad \begin{array}{|c|c|c|}\hline 1&3&5\\\hline 2&4\\\cline{1-2}\end{array}.$$

Figure 4.7: The five tableaux of shape (3,2)

Let m be an element of $TL_n(x)$. Then

$$m\cdot\mathbf{v} = am\cdot\left(\begin{array}{|c|c|c|}\hline 1&2&3\\\hline 4\\\cline{1-1}\end{array}\right) + bm\cdot\left(\begin{array}{|c|c|c|}\hline 1&2&4\\\hline 3\\\cline{1-1}\end{array}\right) + cm\cdot\left(\begin{array}{|c|c|c|}\hline 1&3&4\\\hline 2\\\cline{1-1}\end{array}\right)$$

Therefore, the action of m on any vector in the vector space is known by how m acts on the basis vectors.

Any diagram in TL_n is a product of generator diagrams. Suppose $m = e_ie_j$. Then $m\cdot\mathbf{v} = e_i(e_j\cdot\mathbf{v})$. Therefore, the action of any diagram on a vector is known by how the generator diagrams act on the vector. In conclusion, the action of any diagram on any vector on the Young tableaux vector space of a given shape is known by how the generator diagrams act on the Young tableaux of that shape. Each monoid $TL_n(x)$ will have a different representation for every shape of size n.

4.4 The action

A good clue for how e_i acts on a tableau is found by looking at how elements from the symmetric group act on the tableau. A simple transposition $s_i = (i, i+1) \in S_n$ is the permutation that switches i and $i+1$. The element s_i acts on a tableau t by switching the placements of i and $i+1$. For example,

$$s_4\left(\begin{array}{|c|c|c|}\hline 1&2&4\\\hline 3&5\\\cline{1-2}\end{array}\right) = \begin{array}{|c|c|c|}\hline 1&2&5\\\hline 3&4\\\cline{1-2}\end{array}$$

$$s_4\left(\begin{array}{|c|c|c|}\hline 1&2&3\\\hline 4&5\\\cline{1-2}\end{array}\right) = \begin{array}{|c|c|c|}\hline 1&2&3\\\hline 5&4\\\cline{1-2}\end{array}$$

Notice that sometimes $s_i(t)$ is no longer standard, as seen in the second example above.

The action of e_i on t sends t to a linear combination of t and $s_i(t)$ when $s_i(t)$ is standard, and to a multiple of t when $s_i(t)$ is not standard.

$$e_i\cdot t = \begin{cases} Ct + C's_i(t), & \text{if } s_i(t) \text{ is standard} \\ Ct, & \text{if } s_i(t) \text{ is not standard} \end{cases}$$

4.4.1 $[d]$

Finding the values of C and C' involves the polynomial $[d]$ of x. Further information about $[d]$ can be found in [3]. Finding the value of $[d]$ involves the variable q from the Hecke algebra.

$$x = q + q^{-1} \tag{4.4}$$

$$[d] = \frac{q^d - q^{-d}}{q - q^{-1}} = q^{d-1} + q^{d-3} + \cdots + q^{-(d-1)} \tag{4.5}$$

The following are some values of $[d]$.

$$\begin{aligned}
[0] &= \frac{q^0 - q^0}{q - q^{-1}} = 0 \\
[1] &= \frac{q^1 - q^{-1}}{q^1 - q^{-1}} = 1 \\
[2] &= q^{2-1} + q^{-(2-1)} = q^1 + q^{-1} = x
\end{aligned}$$

The value of [3] takes slightly more work to find.

$$[3] = q^{3-1} + q3 - 3 + q-(3-1) = q^2 + q^0 + q^{-2} = q^2 + 1 + q^{-2}$$

Because there are q's with powers of 2 and -2, compare [3] with x^2.

$$x^2 = (q + q^{-1})^2 = q^2 + 2 + q^{-2}$$

Therefore, $[3] = x^2 - 1$.

Some further values are

$$\begin{aligned}
[4] &= x^3 - 2x \\
[5] &= x^4 - 3x^2 + 1 \\
[6] &= x^4 - 4x^2 + 3 \\
[7] &= x^6 - 5x^4 + 6x^2 - 1
\end{aligned}$$

4.4.2 Finding the Constants C and C'

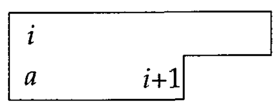
When acting on a tableau with e_i, let d be the number of steps from i to $i+1$ in the tableau. Suppose e_3 were acting on the following tableau.

$$e_3 \cdot \begin{array}{|c|c|c|} \hline 1 & 2 & 3 \\ \hline 4 & 5 \\ \cline{1-2} \end{array}$$

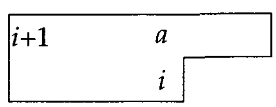
To get from 3 to 4, step from the 3 to the 2, from the 2 to the 1, and then from the 1 to the four. Alternatively, step from the 3 to the 2, from the 2 to the 5, and from the 5 to the four. Either way, it takes three steps to get from 3 to 4: a combination of two steps to the left and one step down. Call this the "southwestern" direction. If the placements of 3 and 4 were switched,

$$e_3 \cdot \begin{array}{|c|c|c|}\hline 1 & 2 & 4 \\ \hline 3 & 5 \\ \cline{1-2} \end{array}$$

then there would still be three steps from 3 to 4, but they would go in the "northeastern" direction. The direction from i to $i+1$ in a standard tableau will always be either to the southwestern or northeastern direction. Paths that are strictly east are still said to go in the northeastern direction, and likewise paths that are strictly south are said to go in the southwestern direction. The direction from i to $i+1$ in a standard tableau will always be either to the southwestern or northeastern direction. Imagine a case where the steps went in a southeastern direction.



For the tableau to be standard, the value of a would have to be greater than i but less than $i+1$. Since the entries can only be integers, there is no possible value of a. Likewise, it would be impossible for the steps from i to $i+1$ to go in a northwestern direction.



The value of a would have to be greater than $i+1$ and less than i.

$$C = \begin{cases} \frac{[d-1]}{[d]} & \text{if } i+1 \text{ is } d \text{ steps N/E from } i \\ \frac{[d+1]}{[d]} & \text{if } i+1 \text{ is } d \text{ steps S/W from } i \end{cases} \tag{4.6}$$

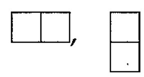
$$C' = \frac{\sqrt{[d-1][d+1]}}{[d]} \tag{4.7}$$

$$e_i \cdot t = \begin{cases} Ct + C's_i(t), & \text{if } s_i(t) \text{ is standard} \\ Ct, & \text{if } s_i(t) \text{ is not standard} \end{cases}$$

4.5 Sample Representations

4.5.1 Representations of $TL_2(x)$

There are two Young diagrams of size two, and so there are two representations that can be found using the above method.



There is one tableau of shape (2).

$$\boxed{1}\boxed{2}$$

There is only one generator, e_1, in $TL_2(x)$. The only action to look at, therefore, is

$$e_1 \cdot \boxed{1}\boxed{2}$$

The step from 1 to 2 is in the northeastern direction, so

$$C = \frac{[1-1]}{[1]} = \frac{[0]}{[1]} = 0$$

Therefore, e_1 maps to (0). It can be seen from the above equation that e_i acting on any tableau where $i+1$ is immediately to the right of i will send that tableau to zero.

There is also only one tableau of shape (1,1).

$$\begin{array}{c}\boxed{1}\\ \boxed{2}\end{array}$$

In this case, the step from 1 to 2 is in the southwestern direction, so

$$C = \frac{[1+1]}{[1]} = \frac{[2]}{[1]} = x$$

It is always the case that $C = x$ when i is directly above $i+1$. In this representation, e_1 maps to (x).

The placements on 1 and 2 could not be switched in either tableau, so there is no C' value.

4.5.2 Representations for $TL_3(x)$

There are two Young diagrams of size three.

$$\begin{array}{|c|c|c|}\hline \ & \ & \ \\ \hline \end{array}, \quad \begin{array}{|c|c|}\hline \ & \ \\ \hline \ \\ \cline{1-1}\end{array}$$

There is only one tableau of the shape (3).

$$\begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline \end{array}$$

Because $i+1$ is immediately to the right of i for all i, both e_1 and e_2 map to (0).

Things get more interesting for tableaux of shape (2,1). There are two tableaux of this shape.

$$\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 \\ \cline{1-1}\end{array} \quad \begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 \\ \cline{1-1}\end{array}$$

Because 2 is directly to the right of 1, e_1 sends the first tableau to zero, the first column in the matrix mapped to by e_1 is $\begin{pmatrix} 0 & \\ 0 & \end{pmatrix}$. The 2 is directly under the 1 in the second tableau, so $e_1 \cdot \begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 \\ \cline{1-1}\end{array} = x\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 \\ \cline{1-1}\end{array}$. Therefore, the second column is $\begin{pmatrix} & 0 \\ & x \end{pmatrix}$. This indicates that the second tableau is sent to the linear combination of zero times the first tableau and x times the second tableau. Thus, e_1 is mapped to $\begin{pmatrix} 0 & 0 \\ 0 & x \end{pmatrix}$.

The matrix for e_2 introduces a non-zero C'. For the first time, switching the i and $i+1$ yields a standard tableau.

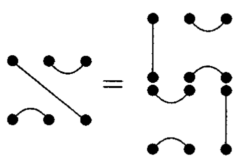
$$\begin{aligned}
e_2 \cdot \begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 \\ \cline{1-1}\end{array} &= C\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 \\ \cline{1-1}\end{array} + C'\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 \\ \cline{1-1}\end{array} \\
&= \frac{[2+1]}{[2]}\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 \\ \cline{1-1}\end{array} + \frac{\sqrt{[2-1]\,[2+1]}}{[2]}\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 \\ \cline{1-1}\end{array} \\
&= \frac{x^2-1}{x}\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 \\ \cline{1-1}\end{array} + \frac{\sqrt{x^2-1}}{x}\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 \\ \cline{1-1}\end{array}
\end{aligned}$$

$$\begin{aligned}
e_2 \cdot \begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 \\ \cline{1-1}\end{array} &= C'\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 \\ \cline{1-1}\end{array} + C\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 \\ \cline{1-1}\end{array} \\
&= \frac{\sqrt{[2-1]\,[2+1]}}{[2]}\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 \\ \cline{1-1}\end{array} + \frac{[2-1]}{[2]}\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 \\ \cline{1-1}\end{array} \\
&= \frac{\sqrt{x^2-1}}{x}\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 \\ \cline{1-1}\end{array} + \frac{1}{x}\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 \\ \cline{1-1}\end{array}
\end{aligned}$$

$$e_2 \longrightarrow \begin{pmatrix} \frac{x^2-1}{x} & \frac{\sqrt{x^2-1}}{x} \\ \frac{\sqrt{x^2-1}}{x} & \frac{1}{x} \end{pmatrix}$$

In the context of this paper, two tableaux that have i and $i+1$ switched but for which everything else is the same shall be referred to as "complements under i." The two tableaux of shape (2,1) are complements under 2. Because complements have the same number of steps from i to $i+1$, the C' values are the same for each.

Now that the matrices in the (2,1) representation have been found for the generators of $TL_3(x)$, it is possible to find matrices for other non-generator diagrams. For example,



$$= e_2e_1 \longrightarrow \begin{pmatrix} \frac{1}{x} & \frac{\sqrt{x^2-1}}{x} \\ \frac{\sqrt{x^2-1}}{x} & \frac{x^2-1}{x} \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & x \end{pmatrix} = \begin{pmatrix} 0 & \sqrt{x^2-1} \\ 0 & x^2-1 \end{pmatrix}$$

Chapter 5

Verifying the relations

Simply assigning an action over a Young diagram vector space does not make the resulting matrices a representation. In order to prove that the resulting matrices are a representation, it is necessary to show that the three generator relations hold. Again the three relations are

$$e_ie_i = xe_i$$
$$e_ie_{i+1}e_i = e_i$$
$$e_ie_j = e_je_i \text{ when } |i-j| \geq 2$$

5.1 The Squared Generator Relation

5.1.1 Rearranging the basis

Notice in the previous examples that the matrices depend on the order of the tableau. If tableau t is the first tableau in the sequence, then $e_i \cdot t$ will be in the first column and the coefficient C of t for $e_i \cdot t$ will be in the first row.

When e_i acts on tableau t, the result is a linear combination of t and $s_i(t)$. By arranging the tableaux in the matrix for e_i such that t is in row and column j and $s_i(t)$ is in row and column $j+1$, the result is an isolated 2×2 block along the diagonal of the matrix. By arranging the tableaux so that all complements under i are adjacent, the resulting matrix becomes block diagonalized into 2×2 blocks and 1×1 blocks. The 1×1 blocks occur when the tableau has no complement under i. An important property of a block diagonalized matrix is that squaring it by itself is the same as squaring each block individually.

The five tableaux of shape (3,2) are as follows.

$$t_1 = \begin{array}{|c|c|c|}\hline 1&2&3\\\hline 4&5\\\cline{1-2}\end{array},\quad t_2 = \begin{array}{|c|c|c|}\hline 1&2&4\\\hline 3&5\\\cline{1-2}\end{array},\quad t_3 = \begin{array}{|c|c|c|}\hline 1&2&5\\\hline 3&4\\\cline{1-2}\end{array},\quad t_4 = \begin{array}{|c|c|c|}\hline 1&3&4\\\hline 2&5\\\cline{1-2}\end{array},\quad t_5 = \begin{array}{|c|c|c|}\hline 1&3&5\\\hline 2&4\\\cline{1-2}\end{array}.$$

When making the matrix for e_2, we notice that t_1 has no compliment under 2, t_2 and t_4 are compliments under 2, and t_3 and t_5 are compliments under 2. If we order the basis t_4, t_2, t_5, t_3, t_1, then we get the following matrix:

$$e_2 \longrightarrow \begin{pmatrix} \frac{1}{x} & \frac{\sqrt{x^2-1}}{x} & 0 & 0 & 0 \\ \frac{\sqrt{x^2-1}}{x} & \frac{x^2-1}{x} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{x} & \frac{\sqrt{x^2-1}}{x} & 0 \\ 0 & 0 & \frac{\sqrt{x^2-1}}{x} & \frac{x^2-1}{x} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Another thing to notice about this arrangement of the basis vectors is that in t_4, the steps from 2 to 3 are in the northeast direction, while in t_2, the steps are in the southwest direction. Likewise, in t_5, the steps are northeast, and in t_3, the steps are southwest. We can always arrange the basis so that complementary vectors are in this order. This is important, as it allows us to generalize the 2×2 block.

5.1.2 The general 2×2 block

The goal of this section is to show that the 2×2 block multiplied by itself under matrix multiplication yields the same 2×2 block scaled by x. The first tableau, t, is the one where the steps from i to $i+1$ are in the northeast direction, so acting on it with e_i yields

$$\frac{[d-1]}{[d]}t + \frac{\sqrt{[d-1]\,[d+1]}}{[d]}s_i(t)$$

so the first column of the matrix is

$$\begin{pmatrix} \frac{[d-1]}{[d]} \\ \frac{\sqrt{[d-1][d+1]}}{[d]} \end{pmatrix}$$

The steps from i to $i+1$ in second tableau, $s_i(t)$, go in the southwestern direction, so acting on it with e_i yields

$$\frac{\sqrt{[d-1]\,[d+1]}}{[d]}t + \frac{[d+1]}{[d]}$$

Thus the 2×2 block looks like

$$\begin{pmatrix} \frac{[d-1]}{[d]} & \frac{\sqrt{[d-1][d+1]}}{[d]} \\ \frac{\sqrt{[d-1][d+1]}}{[d]} & \frac{[d+1]}{[d]} \end{pmatrix}$$

To make this matrix easier to work with, we substitute variables in for $[d]$'s.

$$a = [d-1], \quad b = [d], \quad c = [d+1].$$

The matrix becomes

$$\begin{pmatrix} \frac{a}{b} & \frac{\sqrt{ac}}{b} \\ \frac{\sqrt{ac}}{b} & \frac{c}{b} \end{pmatrix}.$$

When this is squared, it becomes

$$\begin{pmatrix} \frac{a^2+ac}{b^2} & \frac{(a+c)\sqrt{ac}}{b^2} \\ \frac{(a+c)\sqrt{ac}}{b^2} & \frac{ac+c^2}{b^2} \end{pmatrix} = \frac{a+c}{b} \begin{pmatrix} \frac{a}{b} & \frac{\sqrt{ac}}{b} \\ \frac{\sqrt{ac}}{b} & \frac{c}{b} \end{pmatrix}.$$

It remains to prove that $(a+c)/b = x$.

$$\begin{aligned} \frac{a+c}{b} &= \frac{[d-1]+[d+1]}{[d]} \\ &= \frac{q^{d+1}-q^{-d-1}+q^{d-1}-q^{-d+1}}{q^d-q^{-d}} \\ &= \frac{q^{2d+2}-1+q^{2d}-q^2}{q^{2d+1}-q} \\ &= \frac{(q^{2d}-1)(q^2+1)}{q(q^{2d}-1)} \\ &= q+\frac{1}{q} \\ &= x \end{aligned}$$

An equivalent, rather attractive statement, is that

$$\frac{[d-1]+[d+1]}{[d]} = [2]$$

5.1.3 The 1×1 block

The 1×1 blocks occur when a tableau doesn't have a complement, so in order to understand the 1×1 block, it is necessary to know when a tableau does or does not have a complement. Consider each of the following possibilities:

1. i and $i+1$ are in the same row.

   The 1×1 block is a single number, so there is no similar internal structure that says why it should square to x times itself. However, the 1×1 block only occurs when a tableau does not have a complement. It is therefore necessary to determine when a tableau does or does not have a complement.

2. i and $i+1$ are in the same column.

   It must be the case that i is directly above $i+1$, and switching them yields a non-standard tableau, so there would be no complement under i.

3. i and $i+1$ are not in the same row or column.

   There are several different ways the i and $i+1$ can be arranged.

   -
a_1	a_2	a_3	$i+1$
i			

     The a's in the case must be less than $i+1$, and so must be less than i (since they cannot equal i). Therefore, when i and $i+1$ are switched, the result is standard. The number of a's is not specific.

   -
a_1	a_2	a_3	i
$i+1$			

     This is the complement of the one above, so we know it has a complement.

   -
i	
a	$i+1$

     No tableau like this could be standard, because a would have to be greater than i but less than $i+1$. Likewise,

$i+1$	a
	i

  would be impossible, because a would have to be less than i and greater than $i+1$.

The only standard tableaux that do not have complements are those with i and $i+1$ adjacent in a row or in a column. The d value for the number of steps between i and $i+1$ is 1. The 1×1 blocks in the e_1 matrix are the C values for these tableaux.

$$C = \begin{cases} \frac{[1-1]}{[1]} = 0 & \text{if } i \text{ and } i+1 \text{ are in the same row} \\ \frac{[1+1]}{[1]} = x & \text{if } i \text{ and } i+1 \text{ are in the same column} \end{cases}$$

Therefore, it is also true for the 1×1 blocks that squaring them is the same as multiplying by x.

If squaring each block of a block-diagonalized matrix gives x times the block, then squaring the whole matrix gives x times the matrix. Also, while there might not be any arrangement of the basis vectors that block-diagonalizes every e_i matrix, each e_i matrix has a similar matrix that is block diagonalized. Because the similar matrix squared is x times the similar matrix, it is true that the matrix (no matter what order the basis vectors) will square to x times itself.

5.2 The $e_ie_j = e_je_i$ property

Another property of the generators of TL algebra is

$$e_ie_j = e_je_i \quad \text{when} \quad |i-j| \geq 2$$

Consider why the property worked for TL_n diagram. There was no real interaction between the two diagrams because one or the other diagram was always the identity at any given point. A similar thing happens when e_i and e_j act on tableaux. The action of e_i on the tableau involves switching i and $i+1$ while the action of e_j on the tableau involves switching j and $j+1$. Because i and j are further than one apart, no two of these numbers are the same. The switching, at least, is independent of whether e_i or e_j acts on the tableau first.

We must show that, because i and $i+1$ are distinct from j and $j+1$, that

$$e_i \cdot (e_j \cdot v_t) = e_j \cdot (e_i \cdot v_t).$$

Let

$$\begin{aligned} v_t &= \text{the original tableau} \\ v_s &= \text{the original tableau with } i \text{ and } i+1 \text{ switched} \\ v_r &= \text{the original tableau with } j \text{ and } j+1 \text{ switched} \\ v_q &= \text{the original tableau with } j \text{ and } j+1 \text{ switched and } i \text{ and } i+1 \text{ switched.} \end{aligned}$$

For example, we could have

$$v_t = \begin{array}{|c|c|c|c|c|c|}\hline 1&2&3&4&7&10\\ \hline 5&6&8&9\\ \cline{1-4}\end{array}$$

$$i=7:\ v_s = \begin{array}{|c|c|c|c|c|c|}\hline 1&2&3&4&8&10\\ \hline 5&6&7&9\\ \cline{1-4}\end{array}$$

$$j=9:\ v_r = \begin{array}{|c|c|c|c|c|c|}\hline 1&2&3&4&7&9\\ \hline 5&6&8&10\\ \cline{1-4}\end{array}$$

$$v_q = \begin{array}{|c|c|c|c|c|c|}\hline 1&2&3&4&8&9\\ \hline 5&6&7&10\\ \cline{1-4}\end{array}$$

By definition, $\mathbf{v}_t$ and v_s are complements under i, $\mathbf{v}_r$ and $\mathbf{v}_q$ are complements under i, $\mathbf{v}_t$ and $\mathbf{v}_q$ are complements under j, and $\mathbf{v}_s$ and $\mathbf{v}_r$ are complements under j.

Begin with the following assignments for the group action of e_i and e_j on v_t.

$$e_i v_t = A v_t + A' v_s \tag{5.1}$$
$$e_j v_t = B v_t + B' v_r \tag{5.2}$$

Because the distance and direction between i and $i+1$ is the same in tableau $\mathbf{v}_r$ as $\mathbf{v}_t$,

$$e_i \mathbf{v}_r = A\mathbf{v}_r + A'\mathbf{v}_q.$$

Likewise, because the distance and direction between j and $j+1$ is the same in tableau $\mathbf{v}_s$ as $\mathbf{v}_t$,

$$e_j \mathbf{v}_s = B\mathbf{v}_s + B'\mathbf{v}_q.$$

$$\begin{aligned} e_j \cdot (e_i v_t) &= e_j \cdot (A v_t + A' v_s) \\ &= A(e_j v_t) + A'(e_j v_s) \\ &= A(B v_t + B' v_r) + A'(B v_s + B' v_q) \\ &= AB v_t + AB' v_r + A'B v_s + A'B' v_q \end{aligned}$$

$$\begin{aligned} e_i \cdot (e_j v_t) &= e_i(Bv_t + B'v_r) \\ &= B(e_i v_t) + B'(e_i v_r) \\ &= B(Av_t + A'v_s) + B'(Av_r + A'v_q) \\ &= ABv_t + A'Bv_s + AB'v_r + A'B'v_q \end{aligned}$$

Showing that the actions of $e_i e_j$ and $e_j e_i$ are the same on the basis vectors of the tableaux vector space is sufficient for showing that the corresponding representation matrices are the same.

5.3 The $e_i e_{i+1} e_i = e_1$ property

This property says that $e_i e_{i+1} e_i \cdot t = e_i \cdot t$. The proof of this is different for different arrangements of i, $i+1$, and $i+2$. For ease in notation, let

$$\begin{aligned} a &= i \\ b &= i+1 \\ c &= i+2 \end{aligned}$$

For the case where a and b are in the same row, $e_i \cdot t = 0$. Then

$$e_i e_{i+1} e_i \cdot v_t = e_i e_{i+1} \cdot (e_i \cdot t) = e_i e_{i+1} \cdot (0) = 0$$

In all other cases, a or b must go directly in front of c on some row, and the remaining a or b must go on a different row. In order to be standard, no j in the bottom row can be further to the right than the $j-1$ or $j+1$ in the top row (see section 5.1.3). Therefore, there is a total of four cases to examine.

At this point, it is necessary to put the action of generator diagrams on tableaux as a function of direction and number of steps d from i to $i+1$.

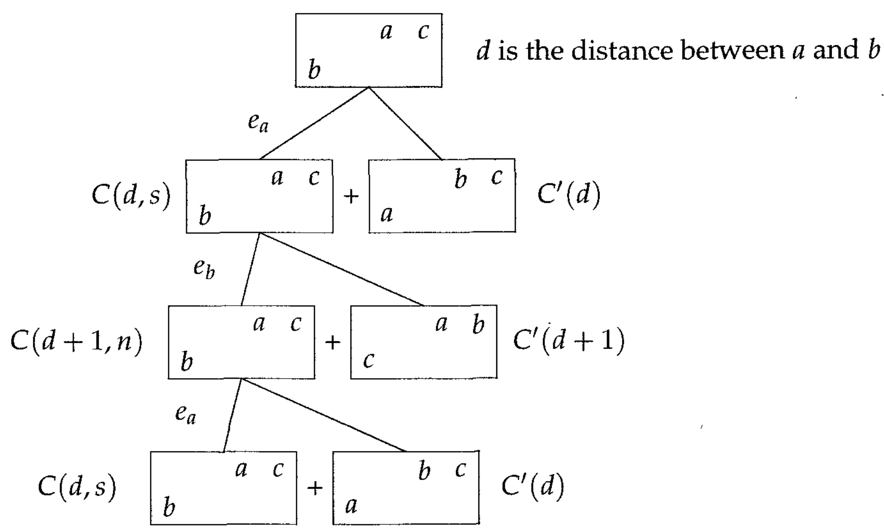
$$\begin{aligned} C(d,n) &= \frac{[d-1]}{[d]} \\ C(d,s) &= \frac{[d+1]}{[d]} \\ C'(d) &= \frac{\sqrt{[d-1]\,[d+1]}}{[d]} \end{aligned}$$

In the tree below, each block contains the letters a, b, and c. The spacing is arbitrary, but because a, b, and c are consecutive, they must be adjacent whenever they appear on the same row. Therefore, letting d be the distance between a and b in the first block means that the distance between b and c is $d+1$. The edges connecting the first and second row of blocks and the third and fourth row of blocks are labeled e_a, while the edges between the second and third row are labeled e_b. Each block connects downward to the linear combination of that arises when that generator acts on the that block. A block does not connect downwards if acting on it with the generator yields zero. In the final row, the two blocks are the tableaux that are involved in $e_a e_b e_a$, but the coefficient of that each tableau is the product of all of the coefficients along to path leading to it from the top block. Thus the diagram below says that, when t is a tableau as in the top block,

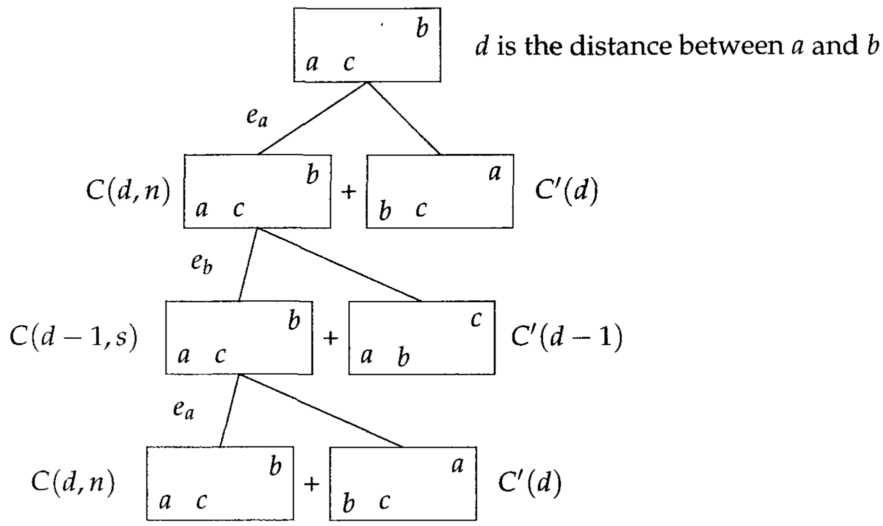
$$\begin{aligned} e_a e_b e_a \cdot t &= C(d,s)C(d+1,n)C(d,s)t + C(d,s)C(d+1,n)C'(d)s_a(t) \\ &= C(d,s)C(d+1,n)(C(d,s)+C'(d)) \\ &= C(d,s)C(d+1,n)(e_a \cdot t) \end{aligned}$$

To show that $e_a e_b e_a \cdot t = e_a \cdot t$, show that $C(d,s)C(d+1,n)$=1.

$$C(d,s)C(d+1,n) = \frac{[d+1]}{[d]} \cdot \frac{[d]}{[d+1]} = 1$$

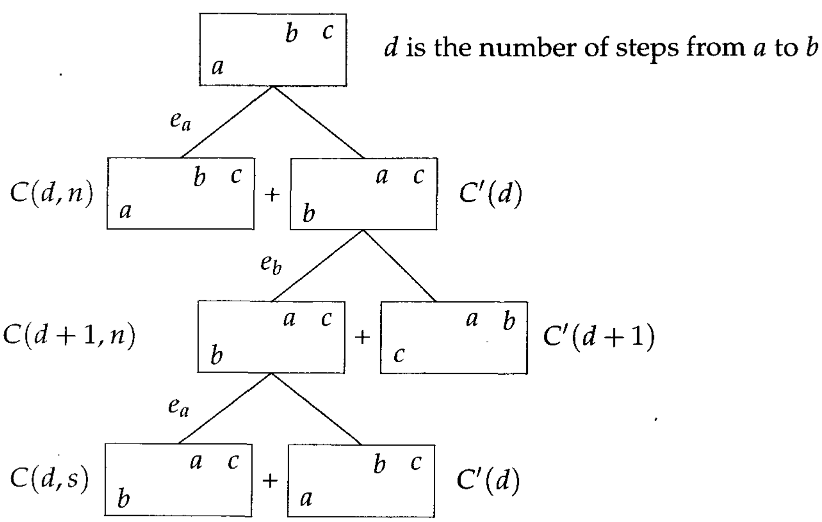


When i and $i+2$ are together in the bottom row and $i+1$ is in the top row, we get a similar chart.



Again, it is a fact that $C(d,n) * C(d-1,s) = \frac{[d-1]}{[d]} \cdot \frac{[d]}{[d-1]} = 1$, so the relation also works when i and $i+2$ are on the bottom row.

The last case that remains is the one where $i+1$ and $i+2$ are in the same row. We begin with the chart showing b and c on the top row.



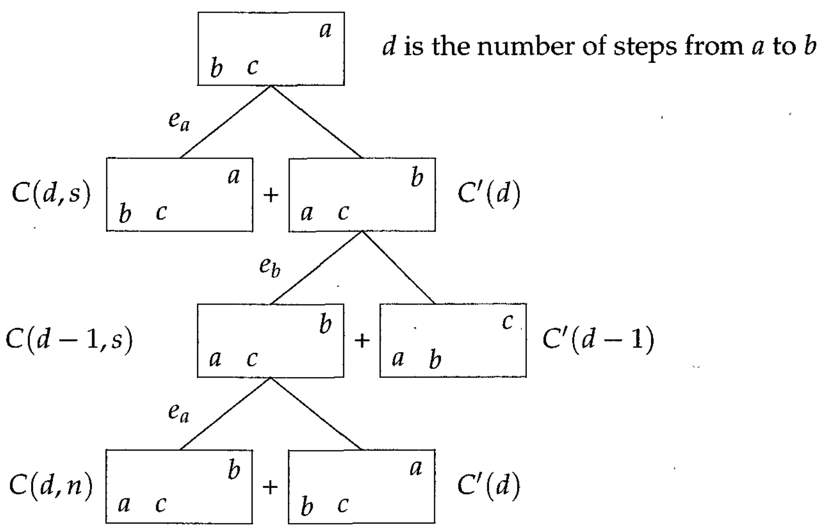
Remarkably,

$$C'(d) * C(d+1,n) * C'(d) = \frac{\sqrt{[d-1]\,[d+1]}}{[d]} \frac{[d]}{[d+1]} \frac{\sqrt{[d-1]\,[d+1]}}{[d]} = \frac{[d-1]}{[d]} = C(d,n) \tag{5.3}$$

and

$$C'(d) * C(d+1,n) * C(d,s) = \frac{\sqrt{[d-1]\,[d+1]}}{[d]} \frac{[d]}{[d+1]} \frac{[d+1]}{[d]} = \frac{\sqrt{[d-1]\,[d+1]}}{[d]} = C'(d) \tag{5.4}$$

When $i+1$ and $i+2$ are in the bottom row, the chart is as follows.



Again, the following equations show that the relation $e_i e_{i+1} e_i = e_i$ works on these tableaux. It is a fact that

$$C'(d) * C(d-1,s) * C'(d) = \frac{\sqrt{[d-1]\,[d+1]}}{[d]} \frac{[d]}{[d-1]} \frac{\sqrt{[d-1]\,[d+1]}}{[d]} = \frac{[d+1]}{[d]} = C(d,s) \tag{5.5}$$

and

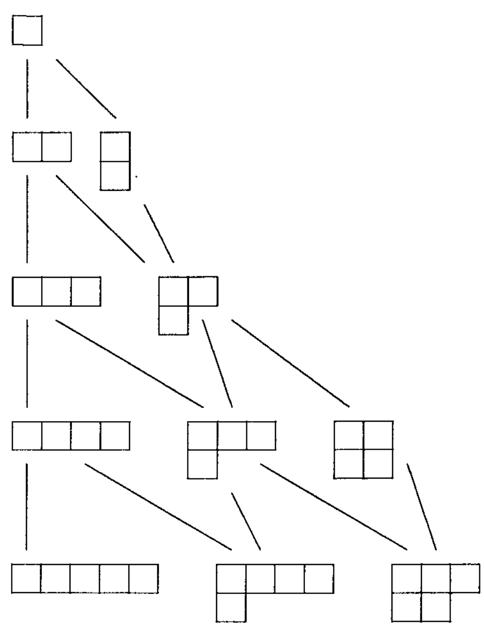
$$C'(d) * C(d-1,s) * C(d,n) = \frac{\sqrt{[d-1]\,[d+1]}}{[d]} \frac{[d]}{[d-1]} \frac{[d-1]}{[d]} = \frac{\sqrt{[d-1]\,[d+1]}}{[d]} = C'(d) \tag{5.6}$$

This completes the proof that the matrices described are a representation for the *TL* monoid.

Chapter 6

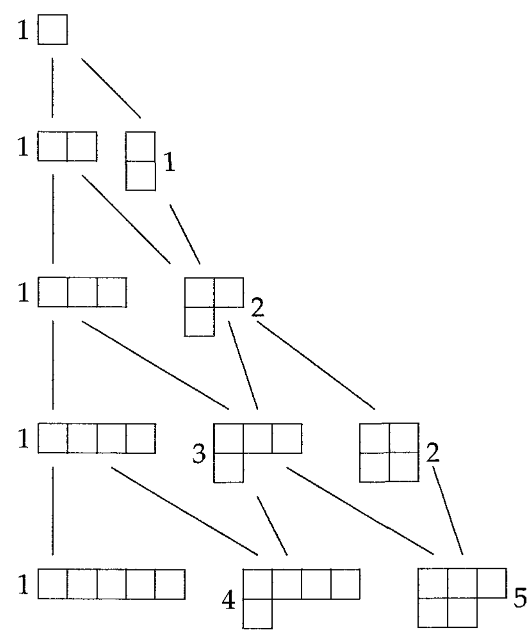
Irreducibility

Each Young diagram of size n has diagrams of size $n-1$ inside it. The following tree shows this relation, limited to the Young diagrams of one or two rows.



This is similar to the way in which TL_n has TL_{n-1} inside it. Simply placing another vertical edge at the end of a TL_{n-1} diagram makes it a TL_n diagram. This concept is familiar from the generators: e_2 is a generator of TL_3, TL_4, TL_5, etc.

We will prove that each representation created using our method is irreducible using induction. First, however, we will show a good indication that the representations are irreducible. As is known from character theory, summing the squares of the dimensions of all irreducible representations gives the order of the group (or monoid). The dimensions of our representations are the number of tableaux that can be made in a certain shape. The following is the same tree as before, and the number next to each Young diagram is the number of tableaux that can be made in that shape.



$$
\begin{aligned}
1^2 &= 1\\
1^2+1^2 &= 2\\
1^2+2^2 &= 5\\
1^2+3^2+2^2 &= 14\\
1^2+4^2+5^2 &= 42
\end{aligned}
$$

These numbers should be recognizable as the first Catalan numbers and orders of monoids TL_1 through TL_5. It would therefore seem reasonable to imagine that these representations are irreducible.

We know that the representation for $n = 1$ corresponding to the single box is an irreducible representation because it is one-dimensional. That is our base case. We then assume that the representations for $n-1$ are irreducible and prove that the representations for n are irreducible. The standard tableaux corresponding to a particular Young diagram can be divided depending on where n shows up in the tableau. For example, with the $(3,2)$ shape,

$$\begin{array}{ccc}1&2&3\\4&5&\end{array},\quad \begin{array}{ccc}1&2&4\\3&5&\end{array},\quad \begin{array}{ccc}1&2&5\\3&4&\end{array},\quad \begin{array}{ccc}1&3&4\\2&5&\end{array},\quad \begin{array}{ccc}1&3&5\\2&4&\end{array}.$$

the vector space over all of these tableaux is the direct sum of the vector space where 5 is in the top row and the vector space where 5 is on the bottom row.

$$\left\{\begin{array}{ccc}1&3&5\\2&4&\end{array},\ \begin{array}{ccc}1&2&5\\3&4&\end{array}\right\} \oplus \left\{\begin{array}{ccc}1&2&3\\4&5&\end{array},\ \begin{array}{ccc}1&2&4\\3&5&\end{array},\ \begin{array}{ccc}1&3&4\\2&5&\end{array}\right\}$$

These correspond to vector spaces for tableaux of size 4, which we have assumed to be irreducible.

$$\left\{\begin{array}{cc}1&3\\2&4\end{array},\ \begin{array}{cc}1&2\\3&4\end{array}\right\} \oplus \left\{\begin{array}{ccc}1&2&3\\4&&\end{array},\ \begin{array}{ccc}1&2&4\\3&&\end{array},\ \begin{array}{ccc}1&3&4\\2&&\end{array}\right\}$$

Because we have assumed that these are irreducible, the above is the only way that the vector space could possibly be reduced to two modules. However, TL_5 has something TL_4 does not have: the generator e_4.

$$e_4 \cdot \begin{array}{ccc}1&2&5\\3&4&\end{array} = C\begin{array}{ccc}1&2&5\\3&4&\end{array} + C'\begin{array}{ccc}1&2&4\\3&5&\end{array}$$

This is true in general. Because we assume the irreducibility of $n-1$, the only possible submodules of a representation of n in TL_n are determined by the placement of n, but e_{n-1} acts on tableau in such a way as to yield a linear combination of tableaux with different placements of n. Therefore, the placement of n does not define a submodule, so the representation has no submodule and is irreducible.

Chapter 7

Loops and How Rank Affects Action

7.1 Loops

Loops in this section are not the "loops" that are dropped from TL_n multiplications and counted with the x variable. Loops are diagrams similar to TL_n diagrams. Instead of two rows of dots, there is one row of an even number of dots. Edges still connect pairs of dots with the restriction that edges cannot cross. The definition of TL_n diagrams says that the edges cannot leave the rectangle defined by the two rows of dots. With only one row of dots, there is no rectangle. However, all edges must fall either above or below the row. In order to be consistent, the edges of all of the loops in this chapter will fall below the row of dots. Information about these loops, including their connection to alternating sign matrices, can be found in [1].

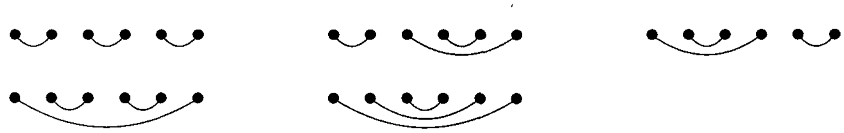


Figure 7.1: Loops with six dots

The loops in Figure 7.1 are recognizable from TL_6 as the top halves of the rank-0 diagrams. Loops such as the first loop in the figure, where each odd dot is connected to the even dot to its right, are very useful. In this paper, they shall be referred to as *simple* loops.

Loops do not have a defined multiplication with each other, but there is a natural way for TL_n diagrams to act on n-loops, following the same rules that apply to multiplication within TL_n. Zinn-Justin has more to say about these actions [9]. Figure 7.2 shows how a TL_n diagram multiplied by an n-loop yields an n-loop.

Figure 7.2: TL_6 diagram acting on 6-loop

Theorem: For every n-loop ℓ, there is a TL_n diagram such that the TL_n diagram acting on the simple n-loop yields ℓ.

The proof of this theorem shows how to construct such a TL_n diagram. In loops other than the simple loop, some edges are nested in others. Sometimes an edge is nested in an edge which is nested in another edge. There is no limit to how many edges can be nested. Sometimes several edges are nested in the same edge but are not nested in each other. There is one 6-loop like this. To begin constructing the TL_n diagram that will take the simple loop to the target loop, consider only those edges that are not nested in any other edge. Draw in bridges between the short edges of the simple loop and extend the ends upwards to make the outer loops. The inner loops will simply be part of the TL_n diagram.

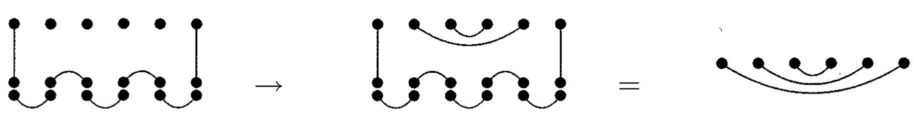
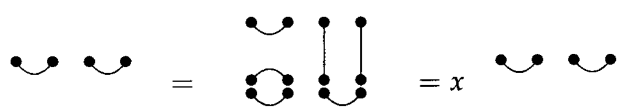


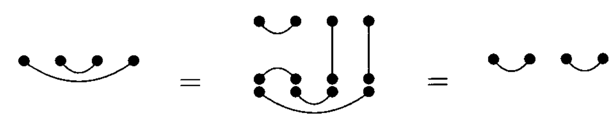
Figure 7.3: Making a nested loop from a simple loop

7.2 The Loop Representation

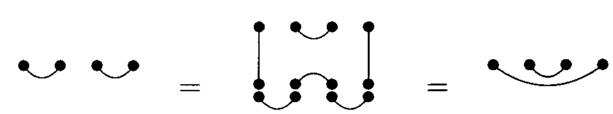
Taking the n-loops to be basis vectors of a vector space, we can act on the n-loops with TL_n diagrams to come up with a representation of TL_n, at least when n is even. Unlike the tableaux representations, we can find the matrices corresponding to each TL_n diagram directly. In the tableaux vector space, we had only defined an action on the vector space for the generator diagrams. In contrast, every TL_n diagram can act on an n-loop. There is also much less involved in showing that the resulting matrices are a representation. Whereas for the tableaux vector space it was necessary to prove that $(gh) \cdot t = g \cdot (h \cdot t)$ where g and h are TL_n diagrams and t is a tableaux, it is apparent that $(gh) \cdot \ell = g \cdot (h \cdot \ell)$ where ℓ is a loop by the fact that TL_n is associative.

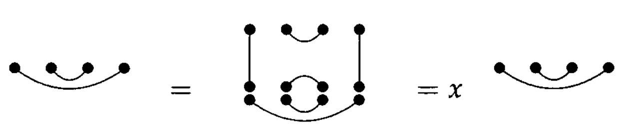
Although it is not necessary to find the generator matrices to find other matrices for the TL_n monoid, it is still a useful exercise. For example, the three generators of TL_4 are e_1, e_2, and e_3. The two 4-loops are the simple loop and the nested loop.

$e_1 \cdot$ 

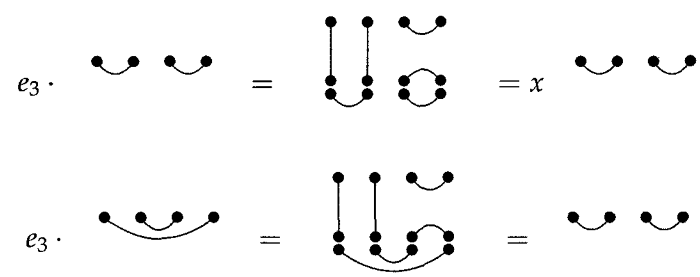
$e_1 \cdot$ 

$$e_1 \longrightarrow \begin{pmatrix} x & 1 \\ 0 & 0 \end{pmatrix}$$

$e_2 \cdot$ 

$e_2 \cdot$ 

$$e_2 \longrightarrow \begin{pmatrix} 0 & 0 \\ 1 & x \end{pmatrix}$$

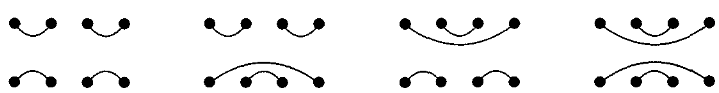


$$e_3 \longrightarrow \begin{pmatrix} x & 1 \\ 0 & 0 \end{pmatrix}$$

This representation is not identical to any of the tableaux representations for TL_4, but in the last chapter we showed that the tableaux representations were all of the irreducible representations. Therefore the loop representation must be isomorphic to a tableaux representation of the same dimension or reduce to a direct sum of tableaux representations of smaller dimension.

Theorem: The loop representation of TL_n is isomorphic to the (k,k) tableaux representation where $2k = n$.

To prove this, consider the rank 0 diagrams of TL_n. For example, there are four rank 0 diagrams in TL_4:



The span of the rank 0 diagrams form a two-sided ideal, since the rank can never go up in multiplication and an ideal is defined as a subset such that multiplying anything in the subset by anything in the set will yield something in the subset. We can make a representation for TL_n algebra by acting on the span of rank 0 diagrams. Acting on the rank 0 diagrams in the order listed above, the matrices corresponding to the generator diagrams of TL_4 are as follows:

$$e_1 \longrightarrow \begin{pmatrix} x & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & x & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad e_2 \longrightarrow \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & x & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & x \end{pmatrix} \quad e_3 \longrightarrow \begin{pmatrix} x & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & x & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

These are block diagonalized, with each 2×2 block equivalent to the loop representation. This makes sense, as the generator diagrams were only acting on the loops that made up the top half of the rank 0 diagrams. Because loop 0 diagrams are TL_n diagrams, it is possible to find their matrix in the Young tableaux representations, which gives a connection between the Young tableaux representations and the loop representation.

It has already been shown that any loop can be made by acting on the simple loop with TL_n diagrams. Therefore, any rank 0 diagrams can be made by multiplying TL_n diagrams by the rank 0 TL_n diagram whose top and bottom rows are both simple loops.

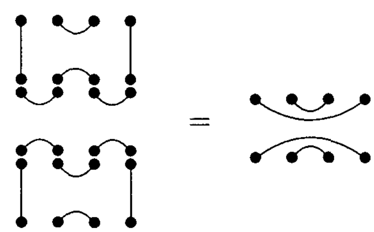


Figure 7.4: Building on to simple rank 0 diagram

The goal is to show that rank 0 diagrams acting on the Young tableaux send all tableaux *not* of shape (k, k) to zero. To do this, show that the rank 0 diagram made of two simple loops sends all tableaux not of shape (k, k) to zero. If the simple diagram acts by sending a tableau to zero, then anything that can have a factor of the simple diagram must also send that tableau to zero.

7.2.1 The simple diagram sends non (k, k) tableaux to zero

The simple diagram of TL_4 is equal to $e_1 e_3$. In general, the simple diagram of TL_n equals $\prod e_i$ where i is an odd integer less than n. The goal is to find the tableau that the simple diagram *doesn't* send to zero. Acting on this tableau with an odd generator would have to return a non-zero product.

We start with

1

The 2 can either go to the right of the 1 or under the 1. If 2 goes to the right of 1, then e_1 goes to zero. Therefore, 2 has to go under the one.

1
2

The 3 cannot go to the right of the 2 because there would be no number that could go over the 3 that would be less than 3. Therefore, the 3 goes to the right of the 1.

1	3
2	

The 4 can go either to the right of the 3 or under the 3. If it goes to the right of the 3, then e_3 acting on the tableau goes to zero. Therefore, 4 has to go under the 3.

1	3
2	4

Continuing on in this way, the only tableau that the simple diagram *doesn't* send to zero is

1	3	5	7	$\cdots$	n-1
2	4	6	8	$\cdots$	n

The simple diagram sends all other tableaux to zero, including other tableaux of the shape (k, k). The generator matrices will have mostly entries that are zero, but with an x in the row and column corresponding to this tableau. However, the matrix representation for the simple diagram, and thus all rank 0 diagrams of TL_n, is the all-zeros matrix for tableaux representations where the tableaux are not of shape (k, k). Thus, the loop representation is equivalent to the (k, k) shape tableaux representation.

7.2.2 The odd case

The case above showed that the matrices corresponding to diagrams of rank 0 in tableaux representations where the tableaux were not of shape (k, k) were the zero matrices. Because rank 0 diagrams only exist when n is even, this begs the question of whether there is a similar situation for when n is odd.

Theorem: Diagrams of rank 1 acting on tableaux of shape other than $(k+1, k)$ where $n = 2k+1$ go to zero.

It would be good to show that every rank 1 diagram can be gotten by taking a product that includes the diagram $\prod e_i$ where i is an odd number less than n. (In this case, n has to be odd, since TL_n where n is even has no rank 1 diagrams.)

Each rank 1 diagram has, by definition, exactly one edge connecting a dot in the top row to a dot in the bottom row. In order to turn this diagram into other rank 1 diagrams by means of diagram multiplication, it is necessary to make bridges along the top row between the dot n and the dot in the top row that should connect to the bottom row. Likewise, make bridges between the dot $n+1$ and the dot in the bottom row that should connect

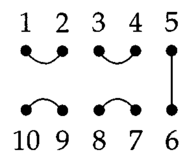


Figure 7.5: Basis rank 1 diagram for TL_5

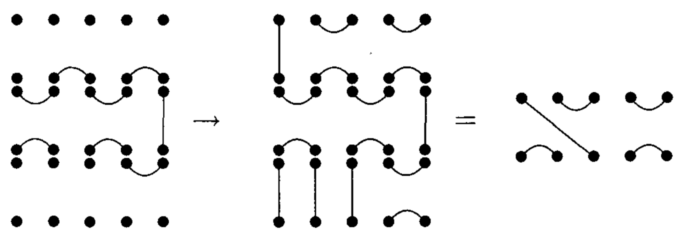


Figure 7.6: Connect dots 1 and 8

to the top row. Fill in the rest of the edges so that the diagram is vertically symmetric. For instance, Figure 7.6 shows the case where the monoid is TL_5 and the rank 1 diagram has a connection between dots 1 and 8. This diagram has the required edge from the top to the bottom and simple loops of various sizes along the other dots. In Figure 7.6, dots 2 through 5 have a simple loop of size 4, while there are two simple loops of size 2 along the bottom row. As has already been shown, simple loops can be converted to any other kind of loop through TL_n multiplication, so multiplying by further diagrams can achieve any desired rank 1 diagram with that ranking edge.

As apparent from Figure 7.5, there are no bridges that can make an even dot on the top row or an odd dot on the bottom row the endpoint for the ranking edge. This is as it should be. An endpoint of the ranking edge has to divide the row of dots it is in so that there are an even number of dots on both sides. If there were an odd number of dots to one side, then one of those dots would have to connect to the other row, and then it would no longer be a rank 1 diagram. It is possible to make bridges so that any acceptable ranking edge can be achieved.

This proves that diagrams like the one in Figure 7.5, $\prod e_i$ where i is an odd number less than n, can be a factor for any rank 1 diagram. By the exact same inductive process used in the even case, these diagrams take every tableaux to zero except for the tableau

1	3	5	7	···	n-2	n
2	4	6	8	···	n-1	

7.2.3 The general case

There are several things that have been shown specifically in this section.

1. The simple diagram generates the ideal.
2. Any rank 1 diagram can have a factor of the diagram that is the simple diagram except for a vertical edge at the end.
3. A rank 0 diagram acting on a tableau not of shape (k, k) where $2k = n$ yields zero.
4. A rank 1 diagram acting on a tableau not of shape $(k+1, k)$ where $2k+1 = n$ yields zero.

These can be generalized as follows.

Theorem: Any rank j diagram can have a factor of the diagram that is the simple diagram except for j vertical edges at the end.

Theorem: Any rank j diagram acting on a tableau where the top row extends more than j boxes past the bottom row yields zero.

To show why the first theorem is true, figure that any given rank j diagram has j edges connecting the top and bottom rows, and because these edges don't cross, they can be ordered from left to right. Start by making the same sort of bridges as previously described to move the left-most vertical edge to the left-most ranking edge. Then move the second edge to the left to the second ranking edge. The property that makes this possible is that the edges of a diagram do not cross. This can be done for as many edges as necessary, so the value of j does not matter.

The diagram described in the second theorem equals $\prod e_i$ where $i < n - j$. The theorem follows the same inductive path as the specific cases.

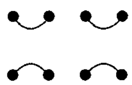
7.3 Rank 0 diagrams isomorphic to full matrix algebra

Any 2×2 matrix can be written as a linear combination of the 2×2 matrices with a one in one entry and zeros in the other entry.

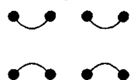
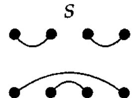
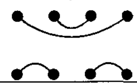
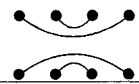
$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = a\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + b\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} + c\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} + d\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

These four matrices make up the basis for an algebra, and that algebra is isomorphic to the algebra with the basis of the four rank-0 diagrams in TL_4. The task of this section is to discover what that isomorphism is.

First, any element in the rank-0 algebra can be written as

a  $+b$  $+c$  $+d$ 

In order to streamline the algebra, let s, t, u, v be the respective diagrams. The following table shows multiplication of these diagrams, with the diagram in the row coming before the diagram in the column.

	r 	s 	t 	u 
r	x^2r	x^2s	xr	xs
s	xr	xs	x^2r	x^2s
t	x^2t	x^2u	xt	xu
u	xt	xu	x^2t	x^2u

The first task is to find the identity element. Multiplying an element by the identity returns the element. Let $I = ar + bs + ct + du$. Then,

$$\begin{aligned} rI &= r(ar+bs+ct+du) = ax^2r + bx^2s + cxr + dxs \\ &= r(ax^2+cx) + s(bx^2+dx) \\ sI &= s(ar+bs+ct+du) = axr + bxs + cx^2r + dx^2s \\ &= r(ax+cx^2) + s(bx+dx^2) \end{aligned}$$

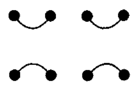
This is enough to find the values for a, b, c, and d. Because rI must equal r, $ax^2 + cx = 1$ and $bx^2 + dx = 0$. Because sI must equal s, $ax + cx^2 = 0$ and $bx + dx^2 = 1$. The augmented matrix for this system of equations is

$$\left(\begin{array}{cccc|c} x^2 & 0 & x & 0 & 1 \\ 0 & x^2 & 0 & x & 0 \\ x & 0 & x^2 & 0 & 0 \\ 0 & x & 0 & x^2 & 1 \end{array}\right)$$

which row reduces to

$$\left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & \frac{1}{x^2-1} \\ 0 & 1 & 0 & 0 & \frac{1}{x-x^3} \\ 0 & 0 & 1 & 0 & \frac{1}{x-x^3} \\ 0 & 0 & 0 & 1 & \frac{1}{x^2-1} \end{array}\right)$$

Thus the identity element is

$\frac{1}{x^2-1}$  $+\frac{1}{x-x^3}$  (7.1)

$+\frac{1}{x-x^3}$  $+\frac{1}{x^2-1}$ 

The next task is to find exactly which elements in the rank-0 algebra correspond to the 2×2 matrices with a single one entry. We have a 2×2 representation for the diagrams in TL_4, namely the loop representation. The matrices for the generator diagrams in this representation have already been found.

$$e_1 \longrightarrow \begin{pmatrix} x & 1 \\ 0 & 0 \end{pmatrix} \quad e_2 \longrightarrow \begin{pmatrix} 0 & 0 \\ 1 & x \end{pmatrix} \quad e_3 \longrightarrow \begin{pmatrix} x & 1 \\ 0 & 0 \end{pmatrix}$$

It is easy to verify that $r = e_1e_3$, $s = e_1e_3e_2$, $t = e_2e_1e_3$, and $u = e_2e_1e_3e_2$. By matrix multiplication, it is possible to find the matrices corresponding to the rank-0 diagrams.

$$r \longrightarrow \begin{pmatrix} x^2 & x \\ 0 & 0 \end{pmatrix} \quad s \longrightarrow \begin{pmatrix} x & x^2 \\ 0 & 0 \end{pmatrix} \quad t \longrightarrow \begin{pmatrix} 0 & 0 \\ x^2 & x \end{pmatrix} \quad u \longrightarrow \begin{pmatrix} 0 & 0 \\ x & x^2 \end{pmatrix}$$

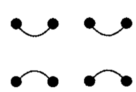
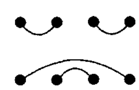
These matrices already have a strong resemblance to the single-entry matrices. The linear combination for $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ will involve r and s with a coefficient a for r and a coefficient b for s such that

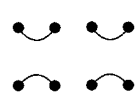
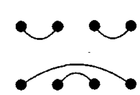
$$ax^2 + bx = 1 \quad \text{and} \quad ax + bx^2 = 0.$$

The coefficients that have already been found for the identity reappear here.

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \frac{1}{x^2-1}r + \frac{1}{x^3-x}s$$

In full, these coefficients are used to form every single-entry matrix.

$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \frac{1}{x^2-1}$  $+\frac{1}{x^3-x}$ 

$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \frac{1}{x^3-x}$  $+\frac{1}{x^2-1}$ 

$\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} = \frac{1}{x^2 - 1}$  $+ \frac{1}{x^3 - x}$ 

$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = \frac{1}{x^3 - x}$  $+ \frac{1}{x^2 - 1}$ 

Chapter 8

Conclusion

Our method for finding representations of the Temperley-Lieb algebra has proven to be interesting in several ways. First, we can find all of the irreducible representations for $TL_n(x)$ for all n in such a way that we can see that they are unique and irreducible based on their underlying structure. The connections in Young's lattice show how $TL_{n-1}(x)$ fits inside $TL_n(x)$. Furthermore, these representations have a rich contextual setting. The representations are indexed by partitions of n of length two, and their bases are indexed by standard Young tableaux. Their derivation from the Hecke algebra, which is a generalization of the seminormal representations of the symmetric group, allows us to identify them as the seminormal representations of the Temperley-Lieb algebra.

It is striking to note how much simpler the loop representation looks, with entries that are either 0 or some power of x, rather than fractions involving x, about half of which also involve square roots. We show, in the last chapter , how the $(n/2, n/2)$ representation was isomorphic to the loop representation.

The way that TL diagrams act on a tableau follows a pattern based on the rank of the diagram and the extension of the top row of the tableau past the bottom row of the tableau. Perhaps this means that there is also a pattern of isomorphism between the tableaux representations and nicer representations. It may also be the case that, like the loop representation, there are simple ways of finding certain, but not all, of the irreducible representations. Having a complete set of irreducible representations could facilitate finding connections between those simple methods. Finding the connection to single-entry matrices would generalize the problem still further.

A final note: As I researched the Temperley-Lieb algebra, I frequently found the literature very difficult to read. One of my objectives for this paper is that it provide an introduction to the topic for those who are just starting to learn about this amazing algebra. I hope that this paper has met that objective.

Bibliography

[1] J. DE GIER, *Loops, matchings and alternating-sign matrices*, ArXiv Mathematics e-prints, (2002).

[2] F. M. GOODMAN, P. DE LA HARPE, AND V. F. R. JONES, *Coxeter graphs and towers of algebras*, vol. 14 of Mathematical Sciences Research Institute Publications, Springer-Verlag, New York, 1989.

[3] T. HALVERSON, M. MAZZOCCO, AND A. RAM, *Commuting Families in Temperley-Lieb Algebras*, ArXiv e-prints, 710 (2007).

[4] V. JONES, *Hecke algebra representations of braid groups and link polynomials*, The Annals of Mathematics, 126 (1987), pp. 335–388.

[5] P. MARTIN AND H. SALEUR, *On Algebraic Diagonalization of the XXZ Chain*, International Journal of Modern Physics B, 8 (1994), pp. 3637–3644.

[6] A. RAM, *Seminormal representations of weyl groups and iwahori-hecke algebras*, (1997).

[7] H. N. V. TEMPERLEY AND E. H. LIEB, *Relations between the 'Percolation' and 'Colouring' Problem and other Graph-Theoretical Problems Associated with Regular Planar Lattices: Some Exact Results for the 'Percolation' Problem*, Royal Society of London Proceedings Series A, 322 (1971), pp. 251–280.

[8] B. WESTBURY, *The Representation Theory of the Temperley-Lieb Algebras*, Mathematische Zeitschrift, 219 (1995), pp. 539–565.

[9] P. ZINN-JUSTIN, *Loop model with mixed boundary conditions, qKZ equation and alternating sign matrices*, Journal of Statistical Mechanics: Theory and Experiment, 1 (2007), pp. 7–+.