Math 191 Long Project: Braid Groups, Representations, and Algebras

Nathan Cheng

May 7th, 2018

1 Introduction

The braid group on n-strands, denoted B_n, is a fundamental algebraic object that has connections to many areas of math including topology, knot theory, and representation theory. It is an algebraic object which encodes the combinatorial data about the ways one can "braid" n parallel strings.

Somewhat recently, there has also been considerable interest in its connections to topological quantum computing [4,5,7]. Setting aside the practical difficulties of implementation, a physical system can theoretically implement a quantum computer if one can show that it can encode a certain class of mathematical properties; particularly, if one can construct a certain class of operators on a Hilbert space. Matrices of complex numbers are the obvious choice, but other algebraic objects can also realize quantum computing if one studies their representations (ways of embedding an object in the space of operators on a vector space). Additionally, some of these alternative ways of realizing quantum computing have turned out to be natural settings for certain implementations and algorithms.

In the case of B_n, it has been shown by Kitaev (reference in [5]) that the braid group can be used to encode data about a theoretical 2-dimensional particle called an anyon, for which he also described a way in which it could realize a physical quantum computing system. Furthermore, since B_n is so closely related to properties of knots (there is a correspondence between braids and knots via braid closure), quantum algorithms for computing the values of certain invariants arise very naturally in this setting. For instance, computing the Jones polynomial of a knot is NP-hard, but values of the Jones polynomial can be approximated in polynomial time by a quantum computer, and in the setting of anyons and B_n, the relationship between the algorithm and the problem is more transparent.

In this report we will survey the representation theory of B_n, covering Hecke algebras, Temperley-Lieb algebras, and the Jones representation. We will begin by covering some preliminary concepts.

2 Representations

Operators on Hilbert spaces (a kind of vector space) are the basic building blocks for modeling quantum mechanical systems. An **operator** on a vector space V is a map from $V \to V$. We will not be too concerned with the exact definition of Hilbert spaces, and will simply think of them as vector spaces, but note that Hilbert spaces carry important properties that general vector spaces do not, and that are essential to modeling quantum mechanical systems.

Definition 2.1: Given a vector space V, we define $GL(V)$ to be the set of all invertible linear operators on V. If $V = S^n$, another way of notating $GL(V)$ is $GL_n(S)$.

Example 2.2: Let $V = \mathbb{R}^2$. A 2×2 matrix represents a linear operator on V. One can also show that any linear operator on V can be expressed as a 2×2 matrix. The invertible maps are given by invertible matrices. Then $GL_2(\mathbb{R})$ is the set of all invertible 2×2 matrices.

Note that $GL(V)$ has a group structure under composition of maps since it always contains the identity operator, and by definition every element is invertible. This observation is important in our next definition.

Definition 2.3: Given a group G and a vector space V, we say that $\phi : G \to GL(V)$ is a **representation** of G on V if it is a homomorphism from G to $GL(V)$.

The motivation for this definition is to generalize the notion of representing a group as a matrix group. The equivalent, abstract notion is embedding a group in the space of operators on an abstractly characterized vector space.

However, operators useful for quantum computing are of a very specific type. Certain operators, when applied to a state of a quantum system, preserve intuitions about probability. These are exactly the unitary operators.

Definition 2.4: An operator U is called **unitary** if $UU^* = U^*U = I$. A representation π is called unitary if $\pi(g)$ is a unitary operator for all $g \in G$.

Example 2.5: Let us consider unitary representations of $\mathbb{Z}$ on $\mathbb{C}$. Note that $GL(\mathbb{C}) = \mathbb{C}$, where $\mathbb{C}$ acts on $\mathbb{C}$ by scalar multiplication. The unitary operators in this case are elements such that $1 = z\bar{z} = |z|^2$; in other words, values in the complex unit circle $\mathbb{T}$. So we are searching for homomorphisms from $\mathbb{Z} \to \mathbb{T}$. Let ϕ be such a homomorphism. $\mathbb{Z}$ has a generator, so the homomorphism is entirely determined by $\phi(1)$. Let $\alpha = \phi(1) \in \mathbb{T}$. A unitary representation of $\mathbb{Z}$ on $\mathbb{C}$ is given by $\phi(n) = \alpha^n$, where α can take on any fixed value in $\mathbb{T}$. Note that if α is an nth root of unity, then ϕ is a homomorphism of $\mathbb{Z}$ onto the nth roots of unity.

Example 2.6: We ask whether there are any interesting representations of B_n. Without proof, we propose that the following gives a representation of B_n. Let $\sigma_1, \sigma_2, ..., \sigma_{n-1}$ be the canonical generators for B_n. Define ϕ on the generators by

$$\phi(\sigma_i) = \left[\begin{array}{c|cc|c} I_{i-1} & 0 & 0 & 0 \\ \hline 0 & (1-t) & t & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & I_{n-i-1} \end{array}\right]$$

where I_k is the $k \times k$ identity block. Extend ϕ to all elements of B_n by extending the map linearly. This gives a representation of B_n on $(\mathbb{Z}[t, t^{-1}])^n$ known as the **unreduced Burau representation** (ϕ is a map from $B_n \to GL_n(\mathbb{Z}[t, t^{-1}])$).

Note that if we fix $t \in \mathbb{C}^\times$, then ϕ induces a representation of B_n on $\mathbb{C}^n$. Unfortunately, the Burau representation is not unitary, nor is it unitary for any fixed t. However, a unitary representation can be obtained from the unreduced Burau representation. Furthermore, the **Alexander polynomial**, a well-known knot invariant, can be expressed using a related representation known as the **reduced Burau representation**. But we will not explore these topics. More results pertaining to the Burau representation can be found in [1,2,3,4,6].

3 Algebras

There seems to be a bit of a structural imbalance in embedding a group into the space of operators. A group is only endowed with one operation, while operators can be added and composed and even scaled. Take, for instance, the Burau representation of B_3 (Example 2.6). Then the images of σ_1 and σ_2 under ϕ are

$$\phi(\sigma_1) = \begin{bmatrix} (1-t) & t & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \phi(\sigma_2) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & (1-t) & t \\ 0 & 1 & 0 \end{bmatrix}$$

As a homomorphism, ϕ encodes the group operation of B_n as the composition of matrices in the image. But

$$\phi(\sigma_1) + \phi(\sigma_2) = \begin{bmatrix} (2-t) & t & 0 \\ 1 & (1-t) & t \\ 0 & 1 & 1 \end{bmatrix}$$

is a perfectly well-defined object, and "feels" as though it should be the image of $\sigma_1 + \sigma_2$ under ϕ, if we want ϕ to have an analogous linearity. Of course, $\sigma_1 + \sigma_2$ is not yet a well-defined object, but the idea of extending the additive structure of $GL(V)$ motivates the following definitions.

Definition 3.1: A **bilinear product** on a vector space V with underlying field $\mathbb{K}$ is a binary operation $\cdot$ such that for all $x, y, z \in V$ and $a, b \in \mathbb{K}$, the following hold:

- $(x + y) \cdot z = x \cdot z + y \cdot z$
- $x \cdot (y + z) = x \cdot y + x \cdot z$
- $(ax) \cdot (by) = (ab)(x \cdot y)$

Definition 3.2: An **algebra** over $\mathbb{C}$ is a vector space over $\mathbb{C}$ equipped with a bilinear product. We call A a ***-algebra** over $\mathbb{C}$ if A is an algebra with a unary * operation from $A \to A$, called an involution, which satisfies for all $x, y \in A, k \in \mathbb{C}$

- $(x + y)^* = x^* + y^*$
- $(xy)^* = y^* x^*$
- $1^* = 1$
- $(x^*)^* = x$
- $(kx)^* = \overline{k} x^*$

The analogy for involution is essentially that of complex conjugation.

Example 3.3: $C[0, 1]$, the set of continuous functions from $[0, 1] \to \mathbb{C}$, is a *-algebra. It is well-known that $C[0, 1]$ is a vector space under pointwise addition and scalar multiplication of functions. If we further endow it with a pointwise product and pointwise conjugation (i.e. f^* is the function such that $f^*(x) = \overline{f(x)}$), then $C[0, 1]$ becomes a *-algebra.

Example 3.4: When $\mathcal{H}$ is a Hilbert space, $GL(\mathcal{H})$ is an *-algebra. The product is composition of operators, and conjugation is given by the Hermitian adjoint. Recall that the Hermitian adjoint is defined via inner product, which Hilbert spaces have.

We can also define representations on algebras in an intuitive manner: simply requiring homomorphisms that preserve the additive, multiplicative, and scalar structures in the space of operators. A representation of a *-algebra also respects the involution, and such a representation is called a ***-representation**. Note that in this case, we no longer require the image of a representation to be contained in the invertible elements, but rather we only require containment in the space of operators, denoted $\mathrm{End}(V)$.

With the goal of making groups into more natural structures for studying representations, we define the group algebra, which is essentially a group with the natural algebraic structure imposed.

Definition 3.5: The group algebra of G is denoted $\mathbb{C}[G]$, and consists of all finite formal sums of the form

$$\sum_{g\in G} a_g g, \text{ where } a_g \in \mathbb{C}$$

We drop the domain of summation where it doesn't cause confusion. All the operations are the natural operations on formal sums, combined with the group operations. For completeness, we document them here:

- $\sum a_g g + \sum b_g g = \sum (a_g + b_g) g$
- $(\sum a_g g)(\sum b_g g) = \sum_{g\in G}(\sum_{h\in G} a_h b_{h^{-1}g})g$
- $k(\sum a_g g) = \sum (k a_g) g$ for $k \in \mathbb{C}$
- $(\sum a_g g)^* = \sum \overline{a_{g^{-1}}} g$

Under these operations, $\mathbb{C}[G]$ is a *-algebra.

Example 3.6: Let $G = \mathbb{Z}_2 = \{e, s\}$, where e is the identity and s is the generator. We will illustrate the multiplication operation. Take the elements $ie + 4s$ and $1e + (1+i)s$ in $\mathbb{C}[G]$. Then

$$\begin{aligned}(ie + 4s)(1e + (1+i)s) &= [i(1) + 4(1+i)]e + [i(1+i) + 4(1)]s \\ &= (5i + 4)e + (i + 3)s\end{aligned} \tag{1}$$

by applying the definition above. Note that this is equivalent to multiplying out the formal sum and collecting terms:

$$\begin{aligned}(ie + 4s)(1e + (1+i)s) &= [i(1) + 4(1+i)]e + [i(1+i) + 4(1)]s \\ &= i(1)e^2 + i(1+i)es + 4(1)se + 4(1+i)s^2 \\ &= ie + (i-1)s + 4s + (4+4i)e \\ &= (5i + 4)e + (i + 3)s\end{aligned} \tag{2}$$

This brings us to the main idea of this section.

Lemma 3.7: The unitary representations of G are in bijection with the *-representations of $\mathbb{C}[G]$.

Proof: Let U be a unitary representation of G on a Hilbert space $\mathcal{H}$. Define $\pi_U : \mathbb{C}[G] \to \mathrm{End}(\mathcal{H})$ by $\pi_U(\sum a_g g) = \sum a_g U(g)$. That additivity, scalar multiplication, and involution are preserved under π_U is fairly easy. We will verify multiplication.

$$
\begin{aligned}
\pi_U((\sum a_g g)(\sum b_g g)) &= \pi_U(\sum(\sum a_h b_{h^{-1}g})g) \\
&= \sum(\sum a_h b_{h^{-1}g})U(g) \\
&= (\sum a_g U(g))(\sum b_g U(g)) \\
&= \pi_U(\sum a_g g)\pi_U(\sum b_g g)
\end{aligned} \tag{3}
$$

So π_U is a *-representation of $\mathbb{C}[G]$ on $\mathcal{H}$.

Now suppose π is a *-representation of $\mathbb{C}[G]$ on $\mathcal{H}$. Consider a map $U : G \to \mathrm{End}(\mathcal{H})$ given by $U(g) = \pi(1g)$. Then for all $g, h \in G$, we have

$$
\begin{aligned}
U(gh) &= \pi(1(gh)) \\
&= \pi((1g)(1h)) \\
&= \pi(1g)\pi(1h) \\
&= U(g)U(h)
\end{aligned} \tag{4}
$$

Furthermore, this is a unitary representation, since

$$
\begin{aligned}
U(g)^* &= \pi(1g)^* \\
&= \pi((1g)^*) \\
&= \pi(\overline{1}g^{-1}) \\
&= \pi(1g^{-1}) \\
&= U(g^{-1})
\end{aligned} \tag{5}
$$

and so $U(g)U(g)^* = U(g)U(g^{-1}) = U(gg^{-1}) = U(1) = 1$ (and likewise for $U(g)^*U(g)$). From this, we can also conclude that U is in fact a map from $G \to GL(\mathcal{H})$.

Remark: We have made many implicit assumptions about the topology of G and the types of representations being considered. In the most general case, a similar statement holds but we have to be careful about the topology we place on G, and the continuity and non-degeneracy of the representations. For our purposes, since B_n is treated as a discrete group (that is: the elements of B_n are essentially completed separated as points in a topological space), things work out nicely, and we don't have to take care of too many of these details.

Lemma 3.7 tells us that studying the representation theory of G is equivalent to studying the representation theory of $\mathbb{C}[G]$.

4 The Temperley-Lieb Algebra

Equipped with a nice idea, we are very excited to look at $\mathbb{C}[B_n]$. But it turns out that $\mathbb{C}[B_n]$ is an *infinite dimensional* object (that is: as a vector space, it has no finite basis set). In general, our techniques for understanding finite dimensional objects are far more robust than for understanding infinite dimensional objects. But not all is lost; looking at a finite dimensional quotient of $\mathbb{C}[B_n]$ may prove a fruitful pursuit, since

Lemma 4.1: If $\phi : A \to B$ is *-homomorphism of *-algebras, and if π is a *-representation of B on $\mathcal{H}$, then π extends to a *-representation $\widetilde{\pi}$ of A on $\mathcal{H}$.

Proof: Define $\widetilde{\pi}$ as $\widetilde{\pi} = \pi \circ \phi$. Composition of homomorphisms is a homomorphism, so $\widetilde{\pi}$ is a homomorphism of A into $\mathrm{End}(\mathcal{H})$, i.e. $\widetilde{\pi}$ is a *-representation of A on $\mathcal{H}$.

Lemma 4.1 suggests that if we can find a quotient space of $\mathbb{C}[B_n]$ with enough of the original structure, then we may still be able to recover interesting representations of B_n. We have a good candidate for this.

Definition 4.2: Let A be some fixed value in $\mathbb{C}$ where A is not a root of unity. Let $d = -A^2 - A^{-2}$. The **Temperley-Lieb algebra** is given by the presentation

$$\begin{aligned} < u_1, u_2, ..., u_{n-1} \mid\ & u_i u_j = u_j u_i \text{ if } |i-j| \geq 2, \\ & u_i u_{i\pm 1} u_i = u_i, \\ & u_i^2 = d u_i > \end{aligned} \tag{6}$$

and is denoted $TL_n(A)$.

Note that $u_1, u_2, ..., u_{n-1}$ are understood as generators of an algebra, so $Tl_n(A)$ consists of finite formal sums with coefficients in $\mathbb{C}$ and variables in words formed by the generators under multiplication. The reason we require A to not be a root of unity is because this presentation may not be well-defined if A is a root of unity. This ends up becoming important later.

$TL_n(A)$ is a nicer algebra to work with, since it is finite dimensional.

Lemma 4.3: $TL_n(A)$ is finite dimensional.

Proof: Consider any finite word formed by the generators

$$u_{a_1}^{k_1} u_{a_2}^{k_2} u_{a_3}^{k_3} ... u_{a_m}^{k_m} \tag{7}$$

where $a_i \in \{1, 2, ..., n-1\}, k_i \in \mathbb{Z} \setminus \{0\}$. By the last relation in Definition 4.2 ($u_i^2 = du_i$), called the **Hecke relation**, we can kill off any powers that are higher than 1 at the cost of including a factor d. In other words, we can get our word into the form

$$C u_{a_1}^{\pm 1} u_{a_2}^{\pm 1} u_{a_3}^{\pm 1} ... u_{a_m}^{\pm 1} \tag{8}$$

where C is some scalar coefficient. Now we show that we can collect all the u_1 terms at the left, so they can be combined into a single term. Restrict our attention to the part of the word that contains a u_1 that is not already at the left

$$...u_{a_i}^{\pm 1} u_1^{\pm 1}... \tag{9}$$

Now if $|1 - a_i| = 0$, then $a_i = 1$ and the two terms can be combined and the entire word can be shortened and then put back into the above form using the Hecke relation. If $|1 - a_i| \geq 2$ or the exponents do not match, then far commutativity (the first relation of Definition 4.2) can be applied and the positions of the elements can be exchanged. If $|1 - a_i| = 1$, then the second relation of Definition 4.2 applies. Note that the second relation is equivalent to $u_{\pm 1} u_i = 1$, so this again means that the entire word can be shortened. In any case, either a term of u_1 disappears or it can be moved to the left, so we can eventually collect them all at the left and then reduce the exponent using the Hecke relation. The same can be done for any u_i using the process we just described, so our word can eventually be put in the form

$$C u_1^{e_1} u_2^{e_2} ... u_{n-1}^{e_{n-1}} \tag{10}$$

where $e_i \in \{-1, 0, 1\}$ (the option for 0 appearing in the case where u_i never showed up in the original word at all). We have just shown that every word generated by multiplication of the generators is equivalent to a scalar multiple of a word of the above form. So the set of all possible words of the above form (ignoring the coefficient C) will span the formal sums in $TL_n(A)$. There are 3^{n-1} such words; in other words, we have a finite spanning set, and hence $TL_n(A)$ is finite dimensional.

The next lemma, stated without proof, relates $\mathbb{C}[B_n]$ to $TL_n(A)$.

Lemma 4.4: An explicit homomorphism from $\mathbb{C}[B_n]$ to $TL_n(A)$ is given by the **Kauffman bracket** $\langle \cdot \rangle : \mathbb{C}[B_n] \to TL_n(A)$ defined by

$$\langle \sigma_i \rangle = A + A^{-1} u_i \tag{11}$$

An important property of $TL_n(A)$ is that it is isomorphic to some subalgebra of a matrix algebra. We briefly describe an example to illustrate this.

Example 4.5: $TL_2(A)$ is isomorphic to the set of complex 2×2 diagonal matrices. Explicitly, we can send the element $1 - \frac{1}{d} u_1$ to $\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ and $\frac{1}{d} u_1$ to $\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$, and extending linearly to all elements. One can verify that this is well-defined and gives a one-one correspondence.

Definition 4.6: The **Jones representation** of B_n is its embedding in the matrix subalgebra associated to $TL_n(A)$.

However, there is an issue with the Jones representation, which is that if we choose A to be a root of unity, the matrix decomposition of $TL_n(A)$ is not always well-defined. But choosing A to be a root of unity is often what we want, in accordance with the idea of getting unitary representations for quantum computing. This can be fixed by passing to yet another quotient known as the **Temperley-Lieb Jones algebra**. However, we will not delve into this. For details on why $TL_n(A)$ fails to be well-defined and how to fix this, please see [4].

5 Temperley-Lieb Algebra Diagrams

Our final discussion will be concerning a visual depiction of the Temperley-Lieb algebra that is analogous to the physical interpretation of the braid groups as braided strands.

Definition 5.1: A **diagram** in $TL_n(A)$ is a square with n points on the top edge and n points on the bottom edge marked, and each of these $2n$ points are connected via non-intersecting lines. There may also be simple closed loops in the diagram.

It turns out that every element of $TL_n(A)$ can be expressed as a formal sum of diagrams. As an example, the diagrams corresponding to the generators of $TL_n(A)$ are

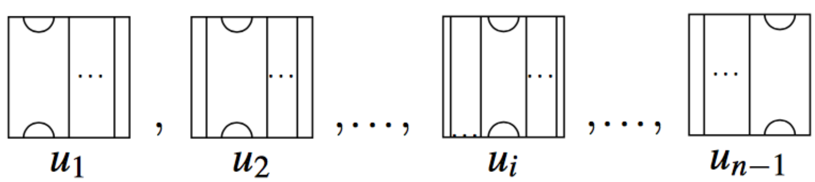

Figure 1: Figure borrowed directly from [4]

We can multiply two diagrams by stacking the second on top of the first and rescaling to get a square. If we multiply two generators u_i, we get

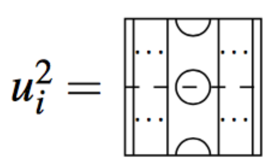

Figure 2: Figure borrowed directly from [4]

Anticipating that u_i^2 should equal du_i, we say that a diagram with a simple closed loop is equivalent to a diagram with that simple closed loop removed but with an extra factor of d in front of the formal sum. In other words, we can exchange a simple closed loop for a factor of d. This property is called d-isotopy.

These diagrams give a visual justification for using the terminology *Kauffman bracket* in Lemma 4.4. Consider B_3, then the Kauffman bracket of σ_2 is $\langle \sigma_2 \rangle = A + A^{-1}u_2$. Note that A is a shorthand for AI, where I is the identity diagram, which consists of 3 parallel strands. Diagrammatically, this gives

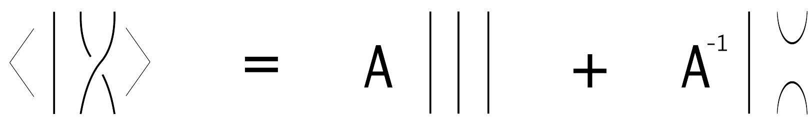

Figure 3: Boundaries of the squares are omitted.

Then it becomes apparent that the Kauffman bracket as we've defined it actually encodes the relation that defines the familiar bracket polynomial. Applying the Kauffman bracket to σ_2 is exactly how we would resolve the crossing in its braid closure $\widehat{\sigma_2}$ if we were computing its bracket polynomial. This relationship expresses the deeper connections between the algebras we have constructed, the representations we are considering, and the Jones polynomial.

6 Conclusion

At the end of the day, we have the following chain of maps

$$B_n \to \mathbb{C}[B_n] \to TL_n(A) \to M_n \tag{12}$$

where M_n is the decomposition of $TL_n(A)$ into a matrix subalgebra. With a few extra details, this turns out to be a sufficiently structure-rich object to theoretically implement a quantum computer. Along the way, we have discussed some results in the representation theory of braid groups, which is important in many areas of math and physics. We have also described connections between the algebraic objects we are considering and the topological objects they represent. We hope that this provides an interesting and correct introduction to braid groups, representation theory, and algebras.

7 References

1. A. Beridze, P. Traczyk. *Forks, Noodles and Burau Representation for $n = 4$.* Journal of Knot Theory and Its Ramifications (2017)

2. S. Bigelow. *The Burau representation is not faithful for $n = 5$.* http://arxiv.org/pdf/math/9904100v2.pdf (1999)

3. Joan S. Birman. *Braids, Links, and Mapping Class Groups.* Annals of Mathematics Studies, Vol. 82, Ch. 3, pg. 102-147 (1975)

4. Colleen Delaney, Eric C. Rowell, Zhengan Wang. *Local Unitary Representations of the Braid Group and Their Applications to Quantum Computing.* Revista Colombiana de Matemáticas, Vol. 50(2), pg. 211-276 (2016)

5. V.T. Lahtinen, J.K. Pachos. *A Short Introduction to Topological Quantum Computation.* SciPost Phys. 3(021) (2017)

6. D. D. Long, M. Pato. *The Burau representation is not faithful $n \geq 6$.* Geometry and Topology, Vol. 3, pg. 397–404 (1999)

7. J. K. Pachos. *Introduction to Topological Quantum Computation.* Lecture Notes in Computer Science (Springer, Berlin, Heidelberg), Vol. 7938 (2013)