Topological Quantum Computing Report II

Temperley-Lieb Algebras, Universality, and BQP-Hardness

Rachel Castro[†]

***Abstract*—This paper shows a set of relationships between quantum computing, braid groups, and representation theory; and especially how topological quantum computing lies at the center. We discuss a handful of landmark papers and a portion of mathematical foundations needed to understand them, specifically Temperley-Lieb algebras.**

I. Introduction and Motivation

In the background paper on Topological Quantum Computing (TQC), we discussed the physical foundations of the subject, and began looking at the braid group to describe swapping of anyons. We neither discussed nor proved the computational power of TQC, which is the subject of this paper. In fact TQC is a model of computing with rich mathematical structure. Here are three statements we will discuss in this paper:

Theorem I.1. *The Jones Polynomial is an invariant representation of a braid.*

Theorem I.2. *Approximating the Jones Polynomial, while difficult on a classical computer, can be computed efficiently on a quantum computer.*

Theorem I.3. *Braiding anyons is universal for quantum computing.*

These theorems describe the relationships between the braid group, quantum computing, and the Jones polynomial, outlined in the below diagram.

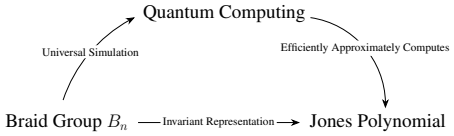

Fig. 1. The motivating relationships of this project.

In this report we will examine each of these connections, starting with the Jones polynomial, as the other two derivations factor through it. We do not have the space to discuss the tractability of calculating the Jones Polynomial in depth, but it should be mentioned in reference to these other two topics.

II. Invariant Representations of Braids

Topological knots have been an object of study since the 1700s, in which mathematicians attempt to classify knots, determine equivalency classes, and identify unique identifying functions for individual knots. While there has been much progress made on determining whether two knots are equal, a perfectly classifying invariant has yet to be identified. One such imperfect invariant of braids is the Jones polynomial. The notion of "imperfect" comes from the fact that two identical braids (under Reidemeister moves) must have the same Jones polynomial, while two distinct braids may have the same polynomial.

As discussed in the background paper to this report, braids are topological objects that can also be understood as elements of a group. For a given number of nodes n, the braid group is generated by the swapping of adjacent notes, subject to two relations. The adjacent swap generators are visualized as

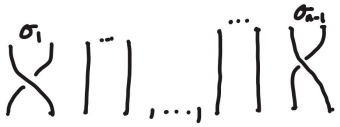

Fig. 2. The generators of B_n

Definition II.1 (The Braid Group B_n)**.** The braid group on n strands is the group on $n-1$ generators [1], subject to restrictions R_1 and R_2 [2]:

$$B_n = \{\sigma_1, ..., \sigma_{n-1} | R_1, R_2\}$$

Where σ_i is a swap of particles p_i and p_{i+1}.

- R_1: Disjoint swaps commute.

$$\sigma_i \sigma_j = \sigma_j \sigma_i \text{ for } |i-j| \geq 2$$

- R_2: Yang-Baxter braid relation.

$$\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}, \forall\ i \in 1, ..., n-2$$

To understand the computational power of braids, we must understand first their Jones representation. The reduction from braid to polynomials factors through Temperley-Lieb Algebras as seen in the figure:

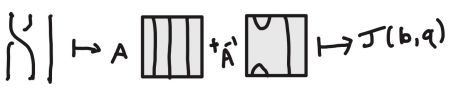

Fig. 3. The reduction through Temperley-Lieb Algebras.

In this section we will examine each step in the process of obtaining the Jones polynomial for a given braid. An important definition that we will use later is the *writhe* of a braid:

[†]*Department of Mathematics, Tufts University*

Definition II.2 (Writhe)**.** Let $b \in B_n = \langle \sigma_1, ..., \sigma_{n-1} \rangle$, then by definition of generating set,

$$b = \sigma_{i_1}^{k_1} \sigma {i_2}^{k_2} \cdots \sigma {i_m}^{k_m}$$

Then we say the writhe of b is:

$$e(b) = \sum_{j=1}^{m} k_j$$

A. Braids to Links

Braids are unique in that they are open strands attached to some topologically invariant surface. In order to use knot theory to study them, we first must convert them into links, or closed loops that are themselves knots and possibly intertwined. There are two methods studied of converting braids into links: the plat closure and the trace closure [3].

A closure is a joining of the free ends of the knots in order to form closed loops. The plat closure connects adjacent pairs, whereas the trace closure adds an additional side of the loop at the end of the braid, closing each strand onto itself. The difference can be see in the below diagram.

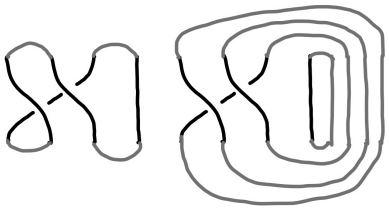

Fig. 4. Left; the plat closure. Right; the trace closure.

If there is an odd number of strands, the plat closure simply connects the final strand to itself. Importantly, any link can be constructed as the plat closure of some braid, and the plat closure of any braid is a valid link–the same follows for trace closures.

B. Temperley-Lieb Algebras

Once a braid $b \in B_n$ has been identified we may further characterize it by its representation in the Temperley-Lieb algebra over d in n strands, $TL_n(A)$. An algebra is a structure like a vector space with well-defined multiplication over the vectors. For example, $GL_n(\mathbb{R})$, the set of all invertible matrices, is an algebra over $\mathbb{R}$. In this case the vectors are the matrices, with multiplication as the usual cross-product of matrices. The scalars are the underlying field $\mathbb{R}$. With this example in mind, we can define the underlying field to be $Q(A)$, the field of rational functions over the Kauffman variable A with complex coefficients. We will denote this field $Q(A) = \mathbb{F}$. [4]

Definition II.3 (Temperley-Lieb Algebra $TL_n(A)$)**.** For $n \in \mathbb{N}$, and $d \in \mathbb{C}$, we define $TL_n(A)$ to be the algebra over **Kauffman diagrams** $\{E_i\}$, subject to the following relations:

1) $E_j E_i = E_i E_j$ for $|i - j| \geq 2$
2) $E_i E_{i+1} E_i . = E_i$
3) $E_i^2 = dE^i$

with an additional "loop variable" $d = -A^2 - A^{-2}$ defined.

We will unpack this definition. First, let us define Kauffman diagrams. Similarly to braid diagrams, we consider a block with n nodes on the top and bottom that must be connected. However, they differ to braid diagrams in two ways: we allow horizontal lines, and we do not allow crossings. Examples of Kauffman diagrams are:

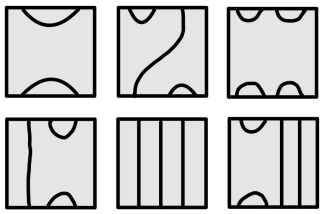

Fig. 5. Examples of Kauffman diagrams.

In fact, these Kauffman diagrams are exactly planar graphs. Then we can consider a generating set of these diagrams as a basis for the algebra as follows:

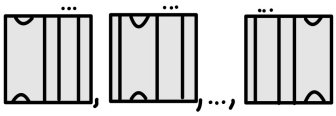

Fig. 6. Basis elements of $TL_n(A)$.

[4]Each diagram can be connected top-to-bottom, similarly to the braid group, to construct further tangle diagrams. This is how "vector multiplication" is defined in this algebra for any Kauffman diagram. In some cases, this can result in a loop being formed: this is not allowed, and thus in simplification, the loop is removed and the diagram gets an additional factor of d, the loop variable.

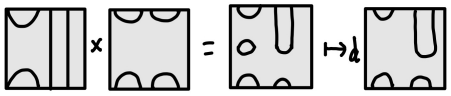

Fig. 7. Vector multiplication in $TL_n(A)$.

The above definition of $TL_n(A)$ may appear familiar: relation 1 is the same as in the braid group B_n, and relation 2 is similar to the Yang-Baxter relation. However, the third relation is unique to $TL_n(A)$, and the Yang-Baxter relation describes a symmetry rather than a reduction. However, there are further connections between B_n and $TL_n(A)$, as show in the rest of this section.

1) Kauffman Bracket: Consider the group algebra over the braid group and a field $\mathbb{F}$,

$$\mathbb{F}[B_n] := \{\sum_f a_b b | b \in B_n, a_b \in \mathbb{F}\}$$

Similarly to other algebraic structures, we can relate algebras by homomorphisms. Thus in order to study the relationship between the braid group and the Temperley-Lieb algebra, we consider the algebra of the braid group. In fact, we define the Kauffman bracket as a homomorphism between the two:

Definition II.4 (Kauffman Bracket)**.** The Kauffman bracket is an algebra homomorphism

$$\langle \cdot \rangle : \mathbb{F}[B_n] \to TL_n(A)$$

such that every crossing in the braid is replaced by one of the following tiles [4]:

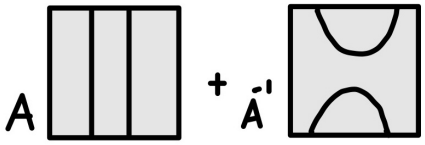

Fig. 8. Braid Crossing Replacements.

Where A is the Kauffman variable inherited from the underlying field. This can be written in the form of $\langle \cdot \rangle : b \mapsto \sum_i AI + A^{-1}E_i$, for each crossing i.

Below is an example of the Kauffman bracket applied to a simple braid σ_1.

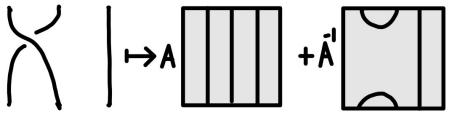

Fig. 9. The Kauffman Bracket in Action.

Importantly, this allows us to convert any braid to an element of $TL_n(A)$ under a homomorphism. [4] This means that for some matrix representation V of $TL_n(A)$ and mapping $\rho' : TL_n(A) \to V$, we may similarly extend this (pullback?) homomorphism to B_n. This implies that V is also a representation of the braid group, which is desirable as we are in search of of a representation.

C. Markov Trace

Next, given any element of $TL_n(A)$, we can calculate its trace by completing the trace closure, then counting the number of loops, or the loop umber d^n. Thus $Tr : TL_n(A) \to \mathbb{F}$, with the standard algebraic trace property $Tr(AB) = Tr(BA)$. [4]

D. Jones Polynomials

In the search for a braid invariant, mathematicians identified the Jones Polynomial. In fact, it is a specific Laurent polynomial, which means it is in fact a polynomial with fractional exponent terms $A^{1/2}$ as well as integer exponents.

Definition II.5 (The Jones Polynomial)**.** Given a braid $b \in B_n$, the Jones polynomial is defined as follows:

$$J(b,q) = \frac{(-A^{-3})^{e(b)} Tr(\langle b \rangle)}{d}$$

Where $q = A^{-4}$ is the variable the polynomial is defined in, $Tr\langle b \rangle$ is the Markov trace of the Kauffman bracket of the braid, and $e(b)$ is the writhe of the braid. [5]

We work through an example to calculate the Jones polynomial of a braid. Consider $b \in B_3$ as $\sigma_1\sigma_2$. First, we identify the writhe as $e(b) = 2$ by application of the definition. Then we find the Kauffman bracket of this braid as follows:

An example of the Kauffman bracket is as follows:

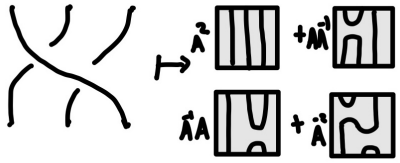

Fig. 10. The evaluation of $\langle b \rangle$.

Then, we calculate the Markov trace of each of these diagrams:

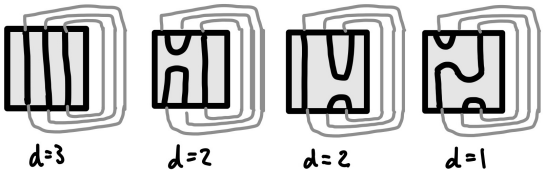

Fig. 11. The Markov Trace of Each Diagram

Then we combine each of these loop variable with their accumulated Kauffman variables left over from the bracket. Thus the total trace is

$$Tr\langle b \rangle = A^2d^3 + 2d^2 + A^{-2}d$$

Then combining all of this information,

$$\begin{aligned} J(b,q) &= \frac{(-A^{-}3)^{e(b)} Tr\langle b \rangle}{d} \\ &= \frac{(-A^{-}3)^2(A^2d^3 + 2d^2 + A^{-2}d)}{d} \\ &= A^{-6}(A^2d^2 + 2d + A^{-2}) \\ &= A^{-4}d^2 + 2d + A^{-2} \end{aligned}$$

with the definition $q = A^{-4}$, we have

$$J(b,q) = qd^2 + 2d + q^{1/2}$$

Simplifying further, with $d = -A^2 - A^{-2}$ we obtain

$$J(b,q) = q(-A^2 - A^{-2})^2 + 2(-A^2 - A^{-2}) + q^{1/2}$$

$$= q(A^4 + A^{-4}) + 2(-A^2 - A^{-2}) + q^{1/2}$$
$$= q(q^{-1} + q) - 2q^{-1/2} - 2q^{1/2} + q^{1/2}$$
$$= q^2 - 2q^{-1/2} - q^{1/2}$$

Thus we have calculated our Jones polynomial. It is clear that even for such a simple braid, the polynomial calculation is nontrivial. In fact, using a classical computer, even approximating the Jones Polynomial is #P-complete, a fact we will discuss in more detail in later sections.

III. Universal Quantum Simulation through Braids

In the last report, we claimed that quantum computation could be performed by swapping anyons. Now we give an overview of the proof of this statement. We have already discussed how swapping anyons gives rise to the braid group, and how the braid group can be represented in the $TL_n(A)$ by the Kauffman bracket. In fact, this representation is a homomorphism, which preserves the operations such that any representation of the algebra is also a representation of the braid group.

This is a fact we will use to show that B_n can execute universal quantum computing operations. We recall from preliminary quantum computing courses the definition of computational universality:

Definition III.1 (Quantum Computational Universality)**.** A gateset G of unitary matrices on n qubits is said to be universal for quantum computation if it generates a dense subgroup of $SU(2^n)$, the special unitary group of $2^n \times 2^n$ matrices. [6]

Specifically, any gate may be **efficiently** approximated by a universal gateset by the **Solovay-Kitaev theorem** [6], discussion of which is beyond the scope of this paper. However, it motivates why we search for universal gatesets: we can efficiently compute any quantum gate using them. When considering topological quantum computation, universality depends on the anyon model being used. Some, like the Fibonacci anyons, are universal, while Majoranas alone generate only the Clifford group. We will discuss this more in-depth later in this section.

Next, we will consider the Jones-Wenzl theorem as discussed in [4]

Theorem III.1 (Jones-Wenzl)**.**

$$TL_n(A) \cong \bigoplus_{\lambda} M_\lambda(\mathbb{C})$$

Where λ are the Young Diagrams of n, and M is complex matrices.

This theorem is useful because in order to prove universality, we need to show a representation of B_n that is dense in $SU(2^n)$. Thus we already have an easy mapping into matrix groups, now we just must show that in fact this corresponds to the Lie group we desire.

This was first shown in [7] for a specific case, then proven in the general case in [8] using their ingenious "two eigenvalue" technique. The argument discussed in this paper goes as follows:

1) We have a homomorphism $\rho : \mathbb{F}(B_n) \to TL_n(A)$.
2) We have an isomorphism $TL_n(A) \cong \bigoplus_\lambda M_\lambda(\mathbb{C})$
3) Based on the rules of the $TL_n(A)$ (specifically $E_i^2 = dE_i$), the image of each braid generator must have exactly two eigenvalues.
4) The only valid groups with this property are $U(N)$, $SU(N)$, $Sp(N)$, $SO(N)$, a few finite groups.
5) By selecting the specific partitions λ, our image is dense in $SU(N)$.

In fact, selecting a specific partition type is done by choosing a specific anyon fusion model. If one selects the Young diagrams that are "hook-shaped" (the first row is the only one with more than one element), you can obtain specifically the mapping into $SU(N)$.

IV. Efficiently Approximating the Jones Polynomial

Finally, we discuss one important result in Quantum Complexity Theory: that calculating Jones Polynomial for a given braid is efficient to approximate using a quantum computer. We define JonesApprox as the following problem:

Definition IV.1 (JonesApprox)**.** Given a braid b on n strands with m crossings, approximate its Jones Polynomial at any primitive root of unity $e^{2\pi i/k}$.

A. A Short Introduction to (Quantum) Complexity Theory

As discussed above, approximating the Jones Polynomial using classical computation falls into #P, pronounced "sharp-P" [5]. This is the class of functions which count the number of solutions to NP problems: for example, consider $\#SAT$, the problem of counting the number of satisfying assignments to a boolean formula. This class is at least as hard as NP but considered to be much harder; in any case, it is not though to be simple to compute classically [9].

However, this notion of difficultly (or maybe the power of quantum computing) was soon changed. In 2006, Aharonov, Jones, and Landau published the AJL algorithm, which solves JonesApprox efficiently using a quantum computer. Following this, Aharonov and Aradi proved in 2011 that JonesApprox is at least as hard as any other problem in BQP. Specifically, the two groups proved that JonesApprox is BQP-complete, which we will define as follows:

Definition IV.2 (BQP Complexity Class)**.** A language $\mathcal{L} \in$ BQP if it can be solved in bounded-probabilistic polynomial time on a quantum computer.

The class BQP serves in quantum complexity as the definition of "efficiency", analogous to P for classical complexity theory. In fact, BQP is much more closely related to BPP, but extensive discussion of this fact is beyond the scope of this paper.

We will briefly discuss the notion of completeness for a complexity class.

Definition IV.3 (Completeness)**.** A problem P is C-complete for a complexity class C if it is both $P \in C$ and C-hard.

Essentially, this means that if P is C-complete, it is both able to be solved in the constraints of C and at least as hard as every other problem in C. Then we can formally state the theorem of this section:

Theorem IV.1. JonesApprox *is* BQP*-Complete. [10], [5]*

Proof. 1) JonesApprox $\in$ BQP. The proof of this is discussed in the AJL Algorithm section.
2) JonesApprox is BQP-hard. The proof of this is shown in the Circuit Reduction section.

□

B. The AJL Algorithm

For any braid $b \in B_n$, with m crossings, we can determine its representation $\langle\cdot\rangle : B_n \to TL_n(A)$. From there, the authors describe another homomorphism $\Phi : TL_n(A) \to \mathcal{H}^{n*}$, the dual of the Hilbert space acting on qubits. Specifically, this maps each element of the braid group to a matrix in $SU(N)$ under the mapping described in the universality section. Then for any image $\Phi(b)$ of $b \in B_n$, there exists some quantum circuit $Q(b)$ which efficiently implements this matrix. [5] Then the AJL algorithm is as follows: For $j \in [0, poly(n, m)]$

1) Generate $|\alpha\rangle = |0101...\rangle$
2) Using the Hadamard test, output x_j such that

$$\mathbb{E}(x_j) = Re \langle\alpha| Q(n) |\alpha\rangle$$

3) Similarly, output y_j such that

$$\mathbb{E}(y_j) = Im \langle\alpha| Q(n) |\alpha\rangle$$

Then if r is the average over all $x_j + y_j$, the Jones polynomial is

$$J(b, q) = \frac{(-A^3)^{e(b)} d\lambda r}{d}$$

where λ is an irrational value. [5]

This runs in polynomial time because the bulk of the calculation is being done by performing the Hadamard test polynomially many times. The Hadamard test is a simple quantum circuit that can calculate either the real or the imaginary part of the output of an arbitrary gate $\mathcal{U}$.

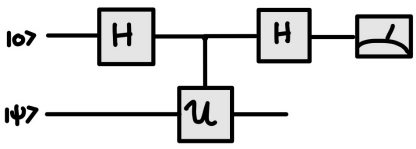

Fig. 12. The Hadamard Test.

C. A Circuit Reduction

In 2011, Aharonov and Arad published their landmark paper "The BQP-hardness of approximating the Jones Polynomial" [10], proving that JonesApprox is BQP-hard. The problem that they reduce from specifically is Universal-Circuit, which we will define as follows:

Definition IV.4 (Universal-Circuit)**.** Given a quantum circuit $\mathcal{U}$ with n qubits and m gates, calculate the probability of the output being $|0\rangle^{\otimes n}$. Ie, calculate

$$|\langle 0|^{\otimes n} \mathcal{U} |0\rangle^{\otimes n}|^2$$

This problem is BQP-Hard [10], thus all we need to show is that by solving an instance of JonesApprox, we are also solving an instance of Universal-Circuit. The proof, unsurprisingly, also depends on the representation of a braid in $TL_n(A)$. The specifics of this proof are beyond the scope of this report, however they proceed by proving that for every quantum circuit $\mathcal{U}$, there exists some $b \in B_n$ such that its Jones polynomial is proportional

$$|\langle 0|^{\otimes n} \mathcal{U} |0\rangle^{\otimes n}|^2 \propto J(b, q)$$

This of course relies on the fact that every circuit can be condensed to one matrix in $SU(N)$ and as the rest of this paper has discussed, there is a strong relationship between B_n and $SU(N)$.

V. Conclusion and Future Work

In this paper we discussed the relationships between quantum computing, braid groups, and invariant representations in Jones polynomials. At the center of these relationships lies derivations through $TL_n(A)$. For future work, I am curious if specific anyon models can be selected for more efficient computation.

References

[1] J. Preskill, "Lecture notes for physics 229, chapter 9: Topological quantum computation." https://www.preskill.caltech.edu/ph229/notes/chap9.pdf, 2004.

[2] C.-H. P. Lee, "Mathematical foundations of topological quantum computation," reu paper, University of Chicago, Department of Mathematics, 2016. Mentors: Peter May and various graduate students.

[3] V. Jones, "The jones polynomial." https://math.berkeley.edu/ vfr/jones.pdf, 2005.

[4] Z. W. Colleen Delaney, Eric C. Rowel, "Local unitary representations of the braid group and their applications to quantum computing." https://people.tamu.edu/ rowell/DRWv10.pdf, 2011.

[5] D. Aharonov, V. Jones, and Z. Landau, "A polynomial quantum algorithm for approximating the jones polynomial," 2006.

[6] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge: Cambridge University Press, 10th anniversary ed. ed., 2010.

[7] M. Freedman, M. Larsen, and Z. Wang, "A modular functor which is universal for quantum computation," 2000.

[8] M. H. Freedman, M. J. Larsen, and Z. Wang, "The two-eigenvalue problem and density of jones representation of braid groups," *Communications in Mathematical Physics*, vol. 228, no. 1, p. 177–199, 2002.

[9] S. Arora and B. Barak, *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[10] D. Aharonov and I. Arad, "The bqp-hardness of approximating the jones polynomial," *New Journal of Physics*, vol. 13, p. 035019, Mar. 2011.