INTRODUCTION TO TOPOLOGICAL QUANTUM COMPUTATION WITH ANYONS

CHIA-HSUN (PAUL) LEE

ABSTRACT. Topological quantum computing (TQC) is a theoretical model for quantum computation. In this model, information is encoded in the braiding of particles, therefore protecting it from local errors. TQC is a promising solution to the problem of decoherence in quantum computing. In this paper, we introduce the abstract model of topological quantum computation and look at two basic examples of TQC models.

CONTENTS

1. INTRODUCTION

One of the main challenges of quantum computation is mitigating errors. A lot of effort has been put in to developing robust quantum error correcting codes with minimal overhead. One class of error correcting codes, called topological codes, utilizes topological properties to encode information in order to prevent local errors from changing the logical state of a quantum system. Such codes have proven to be quite resilient against errors but often require large amounts of qubits to encode one single logical qubit. Topological quantum computing (TQC) is the application of the same principle to a physical system.

Date: August 25, 2018.

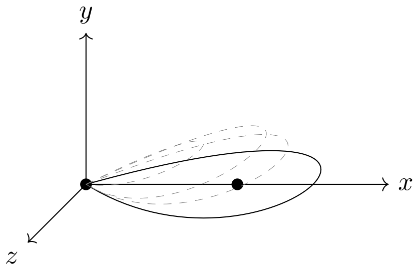


FIGURE 1. The path of a particle travelling around another in 3 or more dimensions is nullhomotopic.

Information in TQC is encoded in the braiding of the worldlines of particles under exchange. The state of particles is encoded by a wave function, and the act of exchanging particles acts on this wave function. The effect on the wave function under exchange is called exchange statistics, or just statistics. The topological nature of a particle system is given by the Aharonov-Bohm effect: a particle travelling along a path around a zero magnetic field gains a phase factor given by the flux of the enclosed magnetic field. In a vacuum with a few particles, the physical significance of the path a particle takes is only defined up to homotopy. Thus local disturbances in a quantum system will not affect the overall quantum state resulting in a topologically protected system. In three or more dimensions, braids in B_n that get sent to the same permutation in S_n, by sending a braid to the permutation of its end points, turn out to be homotopic. This means, that are no non-trivial particle exchanges, shown in Figure 1. Thus, in such a situation, it is the group S_n that is acting on the wave function, resulting in only two particle types, fermions and bosons. In two dimensions, the full braid group B_n acts on the wave function. This allows there to be infinitely many different particles that have arbitrary exchange statistics, called anyons. Topological quantum computing is based on this property that there are infinitely many different particle types. While there is strong experimental evidence that anyons do exist, we have yet to produce any. A slightly more detailed discussion of the mathematical justification for the existence of anyons can be found in my previous REU paper[1].

In principle, particles with different exchange statistics are distinguishable and particles can be split or fused together to form other particles. Further, a pair of particles may have multiple possible fusion outcomes. When necessary, we will call particles that exhibit different exchange statistics as having distinct "topological charges." As the name suggests, the topological charge of a system is a conserved quantity. That is, if we bring all the particles together in a system, no matter what we do, we will always end up with a particle of the same type. As stated in the Aharonov-Bohm effect, in two dimensions, particle exchanges act non-trivially on the wave function of a system. This turns out to influence the probabilities of the different possible outcomes of fusion. We utilize this to do quantum computing: we begin by preparing a set of particles in a known state; we act on it by exchanging

[1] https://math.uchicago.edu/~may/REU2016/REUPapers/Lee.pdf

particles around; in the end we observe the outcomes of fusing particles together. Thus, the information of a system is encoded precisely by the world lines of the particles under exchange. So long as we keep the particles sufficently far apart at all times, the topological nature of the system means that small local errors will not change the information encoded in the system. For the encoded information to change, a large enough error must occur such that the the topological property of a braid is altered.

In this paper, we will be looking at a simplified overview of the model of topological quantum computing. Enough to understand the abstract model and gain working knowledge of how quantum computation is carried out in TQC. The field is at the intersection of many fields in computer science, mathematics and physics and this exposition by no means does it justice. Many details have been left out to keep this at a reasonable length.

In principle, no prior knowledge of physics or quantum computing is required. However, some background may be useful in orienting the reader in the subject. We will begin with a very brief introduction to the basics of quantum computing.

2. Minimal Introduction to Quantum Computation

Before we describe the computational power of anyons, it would be useful to familiarize ourselves with the basic concepts and terminology used in quantum computing. In this section, we will introduce the bare minimum of quantum computing needed in order to understand quantum computation with anyons. A reader already familiar with quantum computation may safely skip this section. A more detailed description of quantum computation may be found in [7], [1], and [5]. Our formalization of quantum computing will closely follow the circuit model of quantum computation. A Quantum Turing Machine formalization exists, but we will not be discussing that in this section. We refer the reader to the references above for details.

We will assume that the reader is familiar with linear algebra and tensor products. References for those topics can be found at [10]. A brief description of Dirac's bra-ket notation will follow now. Dirac's notation is standard among quantum physics and quantum computing.

Notation 2.1. A "ket" written $|\psi\rangle$, is just a vector in a Hilbert space $\mathbb{C}^n$. A "bra" is the conjugate transpose of a "ket," that is, $\langle\psi| = |\psi\rangle^\dagger$. Lastly, $|a\rangle \otimes |b\rangle$ is usually written as $|a\rangle |b\rangle$ or just $|ab\rangle$. To avoid confusion, we will refrain from writing $\langle ab|$ or the likes. We can apply a "ket" to a linear operator, just as you would with a regular vector. For example, $A|\psi\rangle$. We can also apply it to a "bra," resulting in another "bra" like so: $\langle\psi| A = (A|\psi\rangle)^\dagger$. Familiarize yourself with this notation by considering the following objects:

- $A|\psi\rangle$ where $A : \mathbb{C}^n \to \mathbb{C}^m$ is a linear map.
- $\langle\psi|\varphi\rangle$
- $\langle\psi|B|\psi\rangle$ where $B : \mathbb{C}^n \to \mathbb{C}^n$ is an endomorphism.
- $|\psi\rangle \langle\psi|$

Let's recall the circuit model of classical computation. The basic building block of classical computation is the bit. A bit can exist in two states, on or off. Computation is carried out by preparing a collection of bits initialized to a certain state in $\{0, 1\}^n$, manipulated with a sequence of logic gates or boolean functions, such as AND, OR, NOT, and measured at the end to read the output of computation. Quantum computing operates similarly, consisting of initialization of an initial state, manipulation of that state by a sequence of gates, and a measurement of the final state. The key difference is, instead of bits, we have qubits.

2.1. **The Qubit.** The state of a qubit is a unit vector, denoted $|\psi\rangle$, residing in a Hilbert space $\mathbb{C}^2$. Thus, there are continuum many possible states for a single qubit, as opposed to just 2 classically. We choose a set of orthonormal basis vectors $|0\rangle, |1\rangle$ corresponding to the off and on states of a classical bit. We call this the computational basis, or basic states, and all matrices written below are written in this basis. In general, the state of a qubit may be written as,

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$$

where $|\alpha|^2 + |\beta|^2 = 1$. Adding more qubits corresponds to tensoring the state spaces of multiple qubits together. This is expected as, in quantum mechanics, the Hilbert space of the composition of two subsystems is given by the tensor product of the Hilbert spaces of the two subsystems. For example, the state of a 2-qubit system is described by a unit vector residing in $\mathbb{C}^2 \otimes \mathbb{C}^2$. As we fix a computational basis for each of the components, the Hilbert space will have the basis $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. Hence, a state of a 2-qubit system is written as,

$$|\psi\rangle = \alpha |00\rangle + \beta |01\rangle + \gamma |10\rangle + \delta |11\rangle$$

where $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$. In general, the state space of an n-qubit system is $(\mathbb{C}^2)^{\otimes n} = \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2$ with basis $|x\rangle$ where $x \in \{0,1\}^n$ and a state is written as,

$$|\psi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$$

where,

$$\sum_x |\alpha_x|^2 = 1$$

2.2. **Quantum Operations.** Now we've described the state of a collection of qubits, we need a way to manipulate it. Since any operation must take a state of qubits to another state of qubits and be norm preserving, the operation must be a unitary map from $(\mathbb{C}^2)^{\otimes n}$ to itself. As is with the case of classical computation, where gates are boolean functions, we would like to be able to build complex gates from a small set of simple gates, in order to minimize the cost of building a quantum computer. In the classical setting, one single gate is enough, the NAND gate, to build all possible boolean functions from n-bits to m-bits. In the case of quantum computation, it can be shown that a finite set of unitary maps is not enough to generate all unitary maps on n-qubits through a simple cardinality argument. The set of all unitary maps generated from a finite set of maps is countable, while the set of all unitary maps on n-qubits is uncountable. Hence the result. However, two important theorems say that we can do so if we relax certain conditions:

Theorem 2.2 ([1]). *Any unitary map can be decomposed into single qubit gates and the* CNOT *gate given by,*

$$\text{CNOT} : |x_1\rangle |x_2\rangle \mapsto |x_1\rangle |x_1 \oplus x_2\rangle$$

where $\oplus$ is addition modulo 2 and $x_1, x_2 \in \{0,1\}$.

Theorem 2.3 ([7])**.** *We say S is a set of universal gates if any unitary map can be approximated up to ϵ error in the operator norm, for any $\epsilon > 0$, in polylogarithmic in $1/\epsilon$ many gates from S. The following forms a set of universal gates,*

(1) The Hadamard gate, $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

(2) $K = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$

(3) K^{-1}

(4) CNOT

(5) The Toffoli gate, also known as the controlled CNOT*, given by,*

$$\text{TOFFOLI} : |xy\rangle\,|z\rangle \mapsto |xy\rangle\,|(x \times y) \oplus z\rangle$$

where $x, y, z \in \{0, 1\}$ and $\times$ the usual multiplication. That is, flip the bit z if x and y are both 1, otherwise do nothing.

Now that we understand how quantum gates look, we need to address an important issue. All quantum gates are invertible, while classical gates are not necessarily invertible. It is useful, and necessary, to be able to implement classical gates in the quantum computation model. Fortunately, the following theorem allows us to do so,

Theorem 2.4 (Garbage Removal Lemma, [7])**.** *For any classical gate $f : 2^n \to 2^m$ computable by a circuit of size L, there is a permutation $f_\oplus$ on $L+m+n$ bits computable with a circuit of size $2L+m$ such that, $f_\oplus(x, 0, \dots, 0) = f_\oplus(x, f(x), 0, \dots, 0)$ where $x \in 2^n$.*

The proof is simple and can be found in [7]. Usually, a classical boolean function f will be realized by the operator,

$$U_f : |x\rangle\,|y\rangle\,|0^L\rangle \mapsto |x\rangle\,|f(x) \oplus y\rangle\,|0^L\rangle$$

by lifting the classical circuit implementation of $f_\oplus$ directly to quantum gates.

2.3. **Quantum Measurement.** At this point you should be able to see that a single qubit holds vast amounts of information. You could even say that it holds infinite information, as the state of a single qubit generally requires infinitely many classical bits to describe. However, there is a catch, every qubit we prepare only reveals a single bit of information everytime we try to "observe" it. The limitation is given in the following postulate of quantum mechanics:

Postulate 2.5. Any physical observable is associated with a self-adjoint operator A, and the possible outcome of the measurement of an observable A is one of its eigenvalues. Further, if we write a_i as the eigenvalue for the eigenvector $|i\rangle$, then in this basis, a state $|\psi\rangle$ is represented by,

$$|\psi\rangle = \sum_i \alpha_i\,|i\rangle \qquad , \text{ where } \sum_i |\alpha_i|^2 = 1$$

Then, the probability that a measurement yields the outcome a_i is given by $|\alpha_i|^2$. Additionally, if the measurement yields the eigenvalue a_i, the state is projected on to the eigenvector $|i\rangle$. That is, subsequent measurements with the same operator will yield the same results.

In the case of a qubit, we would like to observe it in its computational basis. This corresponds to the following operator,

$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

with eigenvalues 1 corresponding to $|0\rangle$ and -1 corresponding to $|1\rangle$ and the expected value of the measurement of a state $|\psi\rangle$ is given by $\langle\psi|\sigma_z|\psi\rangle$. Hence, with every single qubit you prepare, you can only ever get one single bit of information out of it. To make matters worse, the no-cloning theorem [1] says that the map $\Delta : \mathcal{H} \otimes \mathcal{H} \to \mathcal{H} \otimes \mathcal{H}$ that behaves by $|\psi\rangle\,|\varphi\rangle \mapsto |\psi\rangle\,|\psi\rangle$ on *any states* $|\psi\rangle\,, |\psi\rangle$ is not realizable by a unitary map. Thus, in order to prepare multiple identical qubits, one must start from the beginning from a known state that we can reproduce.

For simplicity sake, if we know that a vector $|\varphi\rangle$ is observable, we will just write $\langle\varphi|\psi\rangle$ to denote the probability of observing the state $|\psi\rangle$ in the state $|\varphi\rangle$ without going through the quantum measurement machinery.

Measurement of multiple qubits at the same time is possible. Detailed discussion of the measurement of multiple qubits is given in [4]. There are subtle differences between measuring a single qubit individually or a bunch at once. Details may be found in the references for quantum computing listed before.

Additionally, we should mention that a global phase factor on a quantum state has no physical significance, i.e., the two states $|\psi\rangle\,, e^{i\varphi}\,|\psi\rangle$ are physically identical. Hence, we usually factor out global phase by rotating the system so that the $|0\rangle$ vector is on the positive real line if possible. That is, the state of a single qubit up to global phase factor, is written as,

$$|\psi\rangle = \cos(\theta/2)\,|0\rangle + e^{i\varphi}\sin(\theta/2)\,|1\rangle$$

where $\theta \in [0, \pi]$ and $\varphi \in [0, 2\pi)$. This representation is not too important, we usually opt for the more simple representation given earlier, as no measument is able to distinguish a global phase anyway. However, this provides a method of visualizing a single qubit in three dimensions, called the Bloch sphere, given by considering θ, φ as the polar coordinates of the state on the unit sphere. We will not be going through it in this paper but I encourage the reader to look it up [1] if they have trouble thinking about qubits.

2.4. **Summary.** In summary, quantum computation consists of the following steps,

(1) The preparation of n-qubits in a certain state $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$.
(2) Application of a sequence of unitary operations, $|\psi\rangle \mapsto |\varphi\rangle = U_n \cdots U_0\,|\psi\rangle$.
(3) Measurement of qubits of the final state φ.

3. Ribbon Unitary Fusion Category

Quantum computation is more powerful than classical computation. It can speedup many problems that are hard classically. Typical examples of quantum speedup are the unstructured database search problem and the integer factorization problem. However, quantum computation as it stands has its limitations. Keeping a quantum system intact remains a huge technological hurdle. Current quantum error correcting methods add a huge amount of overhead to algorithms that make implementing them not feasible or barely advantageous. Topological quantum computing promises to fix this by encoding information topologically, therefore protecting information from being destroyed by local errors.

Recall that topological quantum computing with anyons is facilitated by braiding particles. Particles can be fused together to form another particle and braiding changes the probabilities of the outcomes of fusion. The act of fusing the particles together and observing the outcomes is the measurement of quantum state. The abstract model of topological quantum computation is captured in the unitary modular tensor category. The structure of it is depicted roughly in Figure 2. For the purposes of a simple introduction, we will relax some constraints and look just at ribbon unitary fusion categories. It turns out that this category already holds a huge amount of data.

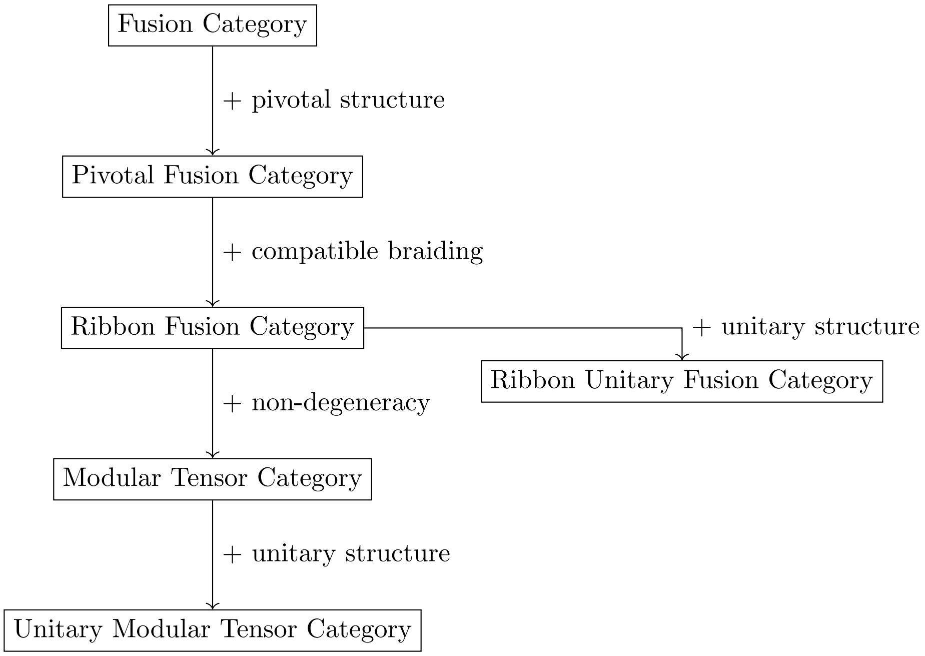


Figure 2. The structure of a unitary modular tensor category (UMTC)

We begin by looking at fusion categories. We start with a few basic definitions.

Definition 3.1 (Tensor Category)**.** A category $\mathcal{C}$ is a tensor category if,

(1) there is a bifunctor $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$,
(2) there is a unit object 1,
(3) for every object x, y, z there is a natural isomorphism $\alpha_{x,y,z} : (x \otimes y) \otimes z \cong x \otimes (y \otimes z)$, that satisfies the pentagon equations,

$$\begin{tikzcd}
 & (a \otimes b) \otimes (c \otimes d) \arrow[dr, "\alpha_{a,b,c\otimes d}"] & \\
 & & a \otimes (b \otimes (c \otimes d)) \\
((a \otimes b) \otimes c) \otimes d \arrow[uur, "\alpha_{a\otimes b,c,d}"] \arrow[ddr, "\alpha_{a,b,c} \oplus \mathrm{id}_d"] & & \\
 & & a \otimes ((b \otimes c) \otimes d) \arrow[uu, "\mathrm{id}_a \otimes \alpha_{b,c,d}"'] \\
 & (a \otimes (b \otimes c)) \otimes d \arrow[ur, "\alpha_{a,b\otimes c,d}"] &
\end{tikzcd}$$

(4) for every object x there are natural isomorphisms $\rho_x : 1 \otimes x \cong x$, $\lambda_x : x \otimes 1 \cong x$ such that the triangle diagram commutes,

$$\begin{tikzcd}
x \otimes y & & (x \otimes 1) \otimes y \arrow[ll, "\lambda_x \otimes \mathrm{id}_y"'] \arrow[dl, "\alpha_{x,1,y}"] & y \otimes x & & y \otimes (1 \otimes x) \arrow[ll, "\mathrm{id}_y \otimes \rho_x"'] \arrow[dl, "\alpha_{y,1,x}"] \\
 & x \otimes (1 \otimes y) \arrow[ul, "\mathrm{id}_x \otimes \rho_y"] & & & (y \otimes 1) \otimes x \arrow[ul, "\lambda_y \otimes \mathrm{id}_x"] &
\end{tikzcd}$$

For convenience, we will require all the above isomorphism to be identities, so that we can write $a \otimes b \otimes c$ without ambiguity. In principle, it is not required for topological quantum computing. This gives us the following definition,

Definition 3.2 (Strict Tensor Category)**.** A tensor category $\mathcal{C}$ is a strict tensor category if, the natural isomorphisms $\alpha_{x,y,z}, \lambda_x, \rho_x$ are all identities.

Let us now add structure that captures the notion of fusion.

Definition 3.3 (Strict Fusion Category)**.** A category $\mathcal{C}$ is a strict fusion category (over $\mathbb{C}$) if,

(1) it is $\mathbb{C}$-linear, i.e. every Hom-set is a $\mathbb{C}$-vector space,
(2) it is a strict tensor category such that $\otimes$ is bilinear on morphisms,
(3) the unit object 1 is simple, i.e. $\mathrm{Hom}(1, 1) \cong \mathbb{C}$,
(4) there are finitely many isomorphism classes of simple objects,
(5) it is semisimple, i.e. every object is a finite direct sum of simple objects,
(6) every object has left and right duals, i.e. for every object x, there are objects $x^*, {}^*x$ along with morphisms, $\eta_x : 1 \to x \otimes x^*, \epsilon_x : x^* \otimes x \to 1$ (right rigidity) and $\eta'_x : 1 \to {}^*x \otimes x, \epsilon'_x : x \otimes {}^*x \to 1$ (left rigidity) such that the following

are identities,

$$\begin{tikzcd} x \arrow[r, "\eta_x \otimes \mathrm{id}_x"] & x \otimes x^* \otimes x \arrow[r, "\mathrm{id} \otimes \epsilon"] & x \end{tikzcd}$$

$$\begin{tikzcd} x^* \arrow[r, "\mathrm{id}_{x^*} \otimes \eta_x"] & x^* \otimes x \otimes x^* \arrow[r, "\epsilon_x \otimes \mathrm{id}_{x^*}"] & x^* \end{tikzcd}$$

$$\begin{tikzcd} x \arrow[r, "\mathrm{id}_x \otimes \eta'_x"] & x \otimes {}^*x \otimes x \arrow[r, "\epsilon'_x \otimes \mathrm{id}"] & x \end{tikzcd}$$

$$\begin{tikzcd} {}^*x \arrow[r, "\eta'_x \otimes \mathrm{id}_{{}^*x}"] & {}^*x \otimes x \otimes {}^*x \arrow[r, "\mathrm{id}_{{}^*x} \otimes \epsilon'_x"] & {}^*x \end{tikzcd}$$

Let us examine the structure of this category and how it relates to topological quantum computing. In the category, a particle is represented by a simple object and a type of particle is represented by an isomorphism class of simple objects. Suppose we have two simple objects, or particles, a and b. If there is a morphism $a \otimes b \to c$, where c is another simple object, we say a, b may fuse to c. If there is a morphism $c \to a \otimes b$, we say c may split to a and b. Suppose $a \otimes b = c \oplus d \oplus e$, where c, d, e are simple objects due to semisimplicity, then it means there are morphisms $a \otimes b$ to c, d, e respectively. Alternatively, we could say c, d, e are the possible outcomes of fusing particles a and b. As one would expect of fusion, the outcome of fusion should be independent of the order we fuse the particles, hence, $\otimes$ is associative. Semisimplicity also says that the possible outcomes of fusion is finite. Further, rigidity says that there exist particle/antiparticle pairs. The term particle/antiparticle pair implies some sort of isomorphism between a particle and its double dual. Indeed, they are automatically isomorphic [3]. However, this isomorphism is not necessarily natural. In order to get the full power of quantum computing, we require that there be a nice isomorphism between objects and their double duals. This is called a pivotal structure and it is essential for more advanced operations (quantum trace). We will define it below after we look at the properties of the fusion category. The following is an important consequence of semisimplicity.

Proposition 3.4. *Every Hom-set in a fusion category is finite dimensional.*

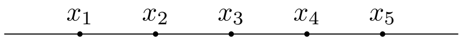
Proof. This is a direct consequence of (1), (3), and (4) in the definition of a fusion category. We have $\mathrm{Hom}(x, x) \cong \mathbb{C}$ if, and only if, x is simple. If y is not simple, it is a direct sum of finitely many simple objects, $y = a_1 \oplus \cdots \oplus a_n$. So, $\mathrm{Hom}(x, y) \cong \mathbb{C}^k$ is finite dimensional where k is exactly the number of i such that $x \cong a_i$. □

This result justifies the later use of intermediate fusion outcomes to label basis vectors for $\mathrm{Hom}(x, y)$. For instance, suppose that $y \otimes z$ can fuse to i, j and both can fuse with x to 1. Then $x \otimes (y \otimes z) = x \otimes (i \oplus j \oplus \cdots) = 1_i \oplus 1_j \oplus \cdots$ where $1_i = 1_j = 1$ and the labels are to identify its source. In particular,

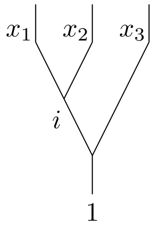
$$\mathrm{Hom}(1, x \otimes (y \otimes z)) \cong \mathrm{Hom}(1, 1_i) \oplus \mathrm{Hom}(1, 1_j) \oplus \cdots$$

So we have at least two basis vectors of $\mathrm{Hom}(1, x \otimes (y \otimes z))$ distinguished by the outcomes of the fusion of $y \otimes z$. In particular, when we add a unitary structure, these fusion trees are going to be labeling an orthonormal basis.

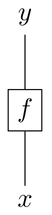
To ease talking about fusion categories, we employ diagrams. In our diagrams, we following the convention in physics that time flows upwards. We denote $x_1 \otimes \cdots \otimes x_n$ as points on a line. For instance, the following is the diagram for $x_1 \otimes \cdots \otimes x_5$:



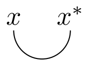
As mentioned above, we identify vectors in a basis given by some order of fusion with the intermediate fusion outcomes. Hence, we use fusion trees to denote basis vectors. The following denotes a basis vector in $\mathrm{Hom}(1, (a \otimes b) \otimes c)$:

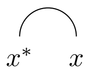


We denote morphisms with just boxes. For example, the diagram for $f : x \to y$ is given by,

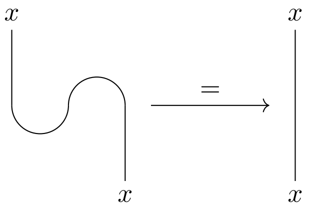


A plain line denotes the identity morphism. We have diagrams for right birth and death,

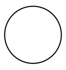




To denote composition, we stack diagrams together from bottom to top consistent with the convention that time flows upwards. In particular, we have intepretations for the following, corresponding to one of the rigidity constraints.

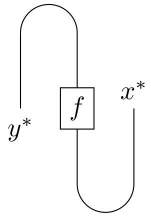


Note, at the moment, there is no interpretation for the following diagram. Since we have no diagram for left birth and death.



In order to define the above, we need a pivotal structure.

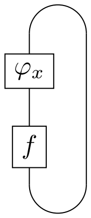
Definition 3.5 (Strict Spherical Fusion Category)**.** A strict fusion category $\mathcal{C}$ is a strict spherical fusion category if it has a nice pivotal structure. Make $*$ into a contravariant functor by defining $f^* : y^* \to x^*$ by the diagram,

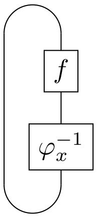


A pivotal structure is a collection of isomorphisms, $\varphi_x : x \to x^{**}$ such that,

(1) for all objects x, y, $\varphi_{x\otimes y} = \varphi_x \otimes \varphi_y$
(2) for all morphisms f, $f^{**} = f$

Additionally, for all morphisms $f : x \to x$ the following two diagrams, called the trace, must agree.





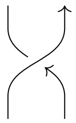
For convenience, we will denote the above by just drawing the box for f with the understanding that the φ_x is inserted appropriately.

With this we can interpret the circle as the trace of the identity. Let us now add structure for braiding.

Definition 3.6 (Strict Braided Fusion Category)**.** A strict fusion category $\mathcal{C}$ is braided if for every pair of objects x, y, there is a natural isomorphism $c_{x,y} : x \otimes y \cong y \otimes x$ such that the hexagon diagram below commutes,

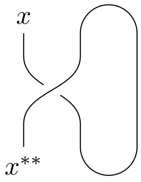
$$\begin{tikzcd}
 & (b \otimes a) \otimes c \arrow[r, "\alpha_{b,a,c}"] & b \otimes (a \otimes c) \arrow[dr, "c_{a,c}"] & \\
(a \otimes b) \otimes c \arrow[ur, "c_{a,b}"] \arrow[dr, "\alpha_{a,b,c}"'] & & & b \otimes (c \otimes a) \\
 & a \otimes (b \otimes c) \arrow[r, "c_{a,b\otimes c}"] & (b \otimes c) \otimes a \arrow[ur, "\alpha_{b,c,a}"'] &
\end{tikzcd}$$

We denote braiding $x \otimes y$ with $c_{x,y}$ by the following diagram,



Note, the braid pictured is in fact a right handed braid, where the strand on the right crosses over the strand on the left. Since time is flowing upwards this may be contrary to other diagrams where braids grow downwards. Arrows have been added to this particular diagram as a reminder of that fact.

Note, the definition differs from that of a symmetric tensor category since $c_{y,x}c_{x,y}$ is not necessarily the identity. Note that braiding gives another isomorphism $\psi_x : x^{**} \to x$ between an object and its double dual, given by the diagram,



Again this isomorphism need not be natural nor does it need to be compatible with the pivotal structure, if it exists, in general. If a spherical fusion category has a compatible braiding then it is called a ribbon fusion category.

Definition 3.7 (Strict Ribbon Fusion Category)**.** A braided spherical fusion category is a ribbon fusion category if for all objects x, $\theta_x = \psi_x \varphi_x : x \to x$ satisfies $\theta_{x^*} = \theta_x^*$.

Lastly, we add a unitary structure in order to perform some basic quantum computation.

Definition 3.8 (Strict Ribbon Unitary Fusion Category, [9])**.** A strict braided fusion category $\mathcal{C}$ is a strict braided unitary fusion category if it has the additional structure,

(1) Hom-sets are Hilbert spaces,
(2) there is conjugation, i.e. a contravariant endofunctor $\overline{}$ acting as the identity on objects and such that $\overline{\overline{f}} = f$, $\overline{f \otimes g} = \overline{f} \otimes \overline{g}$, $\overline{fg} = \overline{g}\overline{f}$ for all morphisms f, g,
(3) for all f, $\overline{f}f = 0$ implies $f = 0$
(4) $\overline{\eta_x} = \epsilon'_x$ and $\overline{\epsilon_x} = \eta'_x$
(5) $\overline{c_{x,y}} = c_{x,y}^{-1}$
(6) $\overline{\theta_x} = \theta_x^{-1}$
(7) for all f, the trace of $f\overline{f}$ is non-negative.

3.1. Quantum Computation with the Ribbon Unitary Fusion Category.

Now we briefly describe how to perform quantum computation in this category. Recall that there are three steps to quantum computation. Initial state preparation, unitary evolution of the state, final measurement. These actions correspond to the following,

(1) To prepare the initial state, we pick a state vector $|\psi\rangle$ in some $\mathrm{Hom}(b, a_1 \otimes \cdots \otimes a_n)$.

(2) We operate on this initial state by picking a sequence of unitary morphisms $\mathrm{Hom}(a_1 \otimes \cdots \otimes a_n, a_{\sigma(1)} \otimes \cdots \otimes a_{\sigma(n)})$ and composing it with $|\psi\rangle$ to get a new state $|\varphi\rangle \in \mathrm{Hom}(b, a_{\sigma(1)} \otimes \cdots \otimes a_{\sigma(n)})$.
(3) Measure $|\varphi\rangle$ in $\mathrm{Hom}(b, a_{\sigma(1)} \otimes \cdots \otimes a_{\sigma(n)})$.

In particular, the morphism in (2) is usually a sequence of braiding operators. Hence, a quantum algorithm in this model is given by a braid and a fusion tree (the choice of basis).

As you can see, all the components of quantum computation are there. However, it's not very helpful if you actually want to *do* quantum computation as there is a lot of structure to specify. It turns out that the structure of a ribbon unitary fusion category is determined (almost completely) by how different types of particles fuse together. This allows us to specify a model of topological quantum computing with far less data. We now turn our attention to that formalization.

4. Braided 6j Fusion Systems

In a fusion category, there are finitely many isomorphism classes, or particle types. The structure of a braided fusion category turns out to be determined by how these particle types fuse pairwise. To describe how particles of different types fuse together, we write them down as fusion rules. The following formalization is given in [9].

Definition 4.1 (Braided 6j Fusion System)**.** A 6j Fusion System consists of the following,

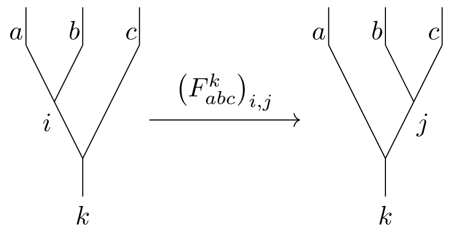
(1) A set of symbols L, whose elemnts we will call particle types, with a distinguished element 1 and an involution $-^* : L \to L$. We will call the distinguished element the vacuum.
(2) A binary operation $\otimes : L \times L \to \mathbb{N}^L$, called the fusion rule. For convenience, we denote $(a \otimes b)(c)$ by N_{ab}^c and write formally $a \otimes b = N_{ab}^a a \oplus N_{ab}^b b \oplus \cdots$ do denote the function given by $a \otimes b$.
(3) A total fusion channel $t \in L$.

satisfying the following,

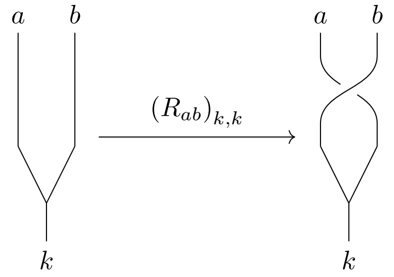
(1) $1^* = 1$.
(2) For all $a, b, c \in L$, $(a \otimes b) \otimes c = a \otimes (b \otimes c)$. In other words, $\sum_x N_{ab}^x N_{xc}^d = \sum_x N_{bc}^x N_{ax}^d$ for all $d \in L$.
(3) For every particle type, there is exactly one other particle type that corresponds to the antiparticle. That is, for all $a, b \in L$,
$$N_{ab} = N_{ba} = \begin{cases} 1 & a^* = b \\ 0 & \text{otherwise} \end{cases}$$
(4) The vacuum fuses trivially with all particle types. That is, for all $a, c \in L$,
$$N_{a1}^c = N_{1a}^c = \begin{cases} 1 & a = c \\ 0 & \text{otherwise} \end{cases}$$

Additionally, the structure needs to be compatible with that of a unitary braided fusion category. Associated with every fusion, $a_1 \otimes \cdots \otimes a_n \to b$ is a Hilbert space $\mathbb{C}^n$ where n is the number of "fusion paths" to b, i.e. the number of ways to fuse to b. Given a fixed order of fusion the Hilbert space has an orthonormal basis parameterized by the intermediate outcomes of fusion [9]. Further, there are unitary matrices F^d_{abc}, R_{ab}, such that analogues of the triangle diagrams, the pentagon diagrams, and the hexagon diagrams given in the section above commute. These equations are given explicitly below.

We use fusion trees to denote vectors of a particular basis of our Hilbert space. The F-matrix relates the two different choices of basis for $a \otimes b \otimes c$, it acts on the basis vectors as follows:



The R-matrix is the braiding matrix. It is always diagonal and acts on the basis vectors as follows:



In particular, the triangle equation is simple: $F^d_{a,b,c} = 1$ if any one of a, b, c is 1. This is what we expect: suppose $b = 1$, then the intermediate outcome of fusing $a \otimes 1$ must be a and the intermediate outcome of fusing $1 \otimes c$ must be c. Now given the above diagrammatic representation, we are able to give the pentagon and hexagon equations diagramatically, given in Figures 3 and 4.

It turns out specifying the fusion rule is almost enough to determine the structure of a braided 6j fusion system.

Theorem 4.2 (Ocneaunu rigidity, [9])**.** *Given a fixed fusion rule, there are only finitely many braided fusion systems with that fusion rule.*

Given a fusion rule, one can find a consistent fusion system by solving the hexagon and pentagon equations. Generally, solving the pentagon and hexagon equations given a fusion rule to find a consistent model is a computationally difficult task [9].

If a braided 6j fusion model gives a ribbon unitary fusion category, we can perform quantum computation with it. A fusion model gives a ribbon fusion category under some conditions (see [9]). For simplicity, assume that the model we are working with does in fact have the required structure. Quantum computation in a braided 6j fusion model is carried out in the following steps,

(1) Prepare a set of particles in a known state $|\psi\rangle$. That is, we've chosen an order of fusion, and we know what the state of the particle is in that basis.

(2) Operate on this initial state by braiding particles.
(3) Fuse the particles together in our chosen order of fusion and observe the intermediate outcomes.

Let us consider two simple fusion rules and examine their computation power:

Example 4.3. When we write out fusion rules, we usually omit the trivial ones and write out the non-trivial ones.

(1) Ising fusion rule: $L = \{1, \sigma, \psi\}$ with,

$$\sigma \otimes \sigma = 1 \oplus \psi \qquad \psi \otimes \sigma = \sigma \otimes \psi = \psi \qquad \psi \otimes \psi = 1$$

(2) Fibonacci fusion rule: $L = \{1, \tau\}$ with,

$$\tau \otimes \tau = 1 \oplus \tau$$

4.1. **Ising Fusion System.** The Ising fusion rules are simple enough to be able to compute the F and R-matrices by hand. For detailed computation see [6]. In particular, there is only one non-trivial F-matrix: $F^{\sigma}_{\sigma\sigma\sigma}$. This matrix is two dimensional ($\sigma \otimes \sigma \otimes \sigma = 2\sigma$) and is,

$$F^{\sigma}_{\sigma\sigma\sigma} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

in the basis given by the fusion trees in the order $\sigma, \sigma \to 1$ and $\sigma, \sigma \to \psi$ distinguished by the outcome of the first fusion. There is also only one non-trivial R-matrix: $R_{\sigma\sigma}$. This matrix is again two dimensional ($\sigma \otimes \sigma = 1 \oplus \psi$) and is given by,

$$R_{\sigma\sigma} = e^{i\varphi} \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

given in the same basis.

In practice, four σ particles are employed to represent a qubit. Fixing an order of fusion, the state corresponding to the first pair of σ fusing to 1 is the logical $|0_L\rangle$ and the state corresponding to fusing to ψ is $|1_L\rangle$.

4.2. **Fibonacci Fusion System.** As for the Fibonacci Fusion System, there is again only one non-trivial F-matrix: $F^{\tau}_{\tau\tau\tau}$ since ($\tau \otimes \tau \otimes \tau = 1 \oplus 2\tau$). This matrix is two dimensional and again ordered by intermediate outcomes $1, \tau$:

$$F^{\tau}_{\tau\tau\tau} = \begin{pmatrix} \varphi^{-1} & \sqrt{\varphi} \\ \sqrt{\varphi} & -\varphi^{-1} \end{pmatrix}$$

There is also only one non-trivial R-matrix. It is two dimensional and given by,

$$R_{\tau\tau} = \begin{pmatrix} e^{4\pi i/5} & 0 \\ 0 & -e^{2\pi i/5} \end{pmatrix}$$

Detailed computation can be found in [8]. Computation of the F and R matrices for this model is simple and the reader can certainly try to derive it. Braids approximating the Hadamard and CNOT gates can be found in [2].

In practice, three τ particles are grouped together to form a qubit. Fixing an order of fusion, we use the same convention, the state corresponding to the first pair fusing to 1 is $|0_L\rangle$ and the state corresponding to fusing to τ is $|1_L\rangle$.

4.3. **Simulating a Traditional Quantum Computer.** The question to ask is whether or not a topological quantum computer is as powerful as a traditional quantum computer. In order to simulate a traditional quantum computer, we need to be able to perform any computation it can do. That is, we need to find a fusion system with braid group representation $\rho : B_n \to \mathrm{SU}(d)$ such that for every unitary U we can find a braid $b \in B_n$ such that the following commutes, up to arbitrary precision,

$$\begin{tikzcd}
(\mathbb{C}^2)^{\otimes n} \arrow[r, hook, "i"] \arrow[d, "U"] & V_n \arrow[d, "\rho(b)"] \\
(\mathbb{C}^2)^{\otimes n} \arrow[r, hook, "i"] & V_n
\end{tikzcd}$$

where i is a chosen embedding of the logical n-qubit space in our fusion space V_n. In particular, if the image of the braid group representation is dense, then the above is achievable. A conjecture of a sufficient condition for universality can be found in [9].

In particular, the Ising model does not support universal quantum computation while the Fibonacci model does. However, one advantage the Ising model has is that there is a simple braid for implementing the CNOT gate exactly. It turns out that if we can perform single qubit rotations through external means, we get quantum computation. Additionally, the Ising model is currently the one closest to experimental realization.

On the other side of the question whether topological quantum computers are as strong as traditional quantum computers, is the question whether they are stronger. As it turns out, they are not, and a quantum computer can efficiently simulate a topological quantum computer [9].

5. Analog Computation of the Jones Polynomial

The topological nature of TQC provides an analog algorithm for approximating the Jones polynomial at some root of unity q. Suppose we have a link L given as the trace closure of some braid B. Recall that the Jones representation ρ_A is unitary for $A = \pm i e^{\pm 2\pi i/4r}$ [9] and that the Jones polynomial is given by,

$$V_L(A) = (-A)^{3\,\mathrm{Wr}(L)} d^{n-1}\,\mathrm{Tr}(\rho_A(B))$$

where Wr is the writhe map and $d = -A^2 - A^{-2}$. In particular, given an appropriate model of TQC with a braid group representation given by $\rho_A(B)$, the trace $\mathrm{Tr}(\rho_A(B))$ is given by the diagram of the trace closure of the braid B. Specifically, let $|\psi\rangle$ denote the state corresponding to the pairwise creation of particle and antiparicle pairs in $\mathrm{Hom}(1, x_1 \otimes \cdots \otimes x_1^*)$. The diagram can be written as $\langle\psi|\rho_A(B)|\psi\rangle \in \mathrm{Hom}(1,1)$. The quantum mechanical interpretation of this value is the probability that pairwise fusion of the particles after evolution $\rho_A(B)$ yields the vacuum. Hence, we can additively approximate this value to compute the Jones polynomial (see Figure 5 for an example).

The problem of additive approximation of the Jones polynomial of a link L at $e^{\pm 2\pi i/r}$ where $r > 6$ turns out to be BQP-complete [9], where BQP is considered to be the feasible class of quantum computation. It is the class of languages that can be decided by a polynomial sized quantum circuit with success probability at least

2/3. It is the quantum analog of BPP. For more information on the complexity of those classes see [7].

6. Conclusion

TQC is a topologically protected model of quantum computation. Although not yet experimentally confirmed, the principles have been applied to quantum error correcting codes. Further, the topological nature facilitates analog computation. Classically, analog computation has been unsuccessful due to its noisy nature. Small errors can quickly snowball into large errors. Since TQC is immune to local changes, there is built in protection from small errors, perfect for analog computation. The model provides a novel way of thinking about quantum computation that is distinct from the traditional circuit model.

Further, the topological nature of the model may be advantageous to dealing with problems that are topological in nature. The approximation of the Jones polynomial is the prime example for this. Perhaps there are other classically hard topological problems that may be solved efficiently on a quantum computer.

Acknowledgements

I would like to thank my mentor, Randy Van Why, for providing me guidance in writing this exposition. I would also like to thank my friends, Josef Klafka and Anand Abraham, for providing feedback and Claudio Gonzales, for introducing me to this subject.

All diagrams and figures have been drawn with Ti*k*Z, the `tikz-cd` package and the `braids` package. These packages can be found on CTAN.

References

[1] Giuliano Benenti, Giulio Casati, and Giuliano Strini. *Principles of quantum computation and information: Volume II: Basic Tools and Special Topics.* World Scientific, 2007.

[2] N. E. Bonesteel. "Braid Topologies for Quantum Computation". In: *Physical Review Letters* 95.14 (2005). DOI: `10.1103/PhysRevLett.95.140503`.

[3] César Galindo. "On braided and ribbon unitary fusion categories". In: *arXiv preprint arXiv:1209.2022* (2012).

[4] Daniel F. V. James et al. "Measurement of qubits". In: *Physical Review A* 64.5 (Oct. 2001). ISSN: 1094-1622. DOI: `10.1103/physreva.64.052312`. URL: `http://dx.doi.org/10.1103/PhysRevA.64.052312`.

[5] Alexei Yu Kitaev et al. *Classical and quantum computation.* 47. American Mathematical Soc., 2002.

[6] Jiannis K Pachos. *Introduction to topological quantum computation.* Cambridge University Press, 2012.

[7] Alexander Razborov. *Lecture Notes on Quantum Computing.* URL: `http://people.cs.uchicago.edu/~razborov/teaching/QuantumComputing/notes.pdf` (visited on 08/15/2018).

[8] Simon Trebst et al. "A Short Introduction to Fibonacci Anyon Models". In: *Progress of Theoretical Physics Supplement* 176 (2008), pp. 384–407. ISSN: 0375-9687. DOI: `10.1143/ptps.176.384`. URL: `http://dx.doi.org/10.1143/PTPS.176.384`.

[9] Zhenghan Wang. *Topological quantum computation.* 112. American Mathematical Soc., 2010.

[10] Takeo Yokonuma. *Tensor spaces and exterior algebra.* 108. American Mathematical Soc., 1992.

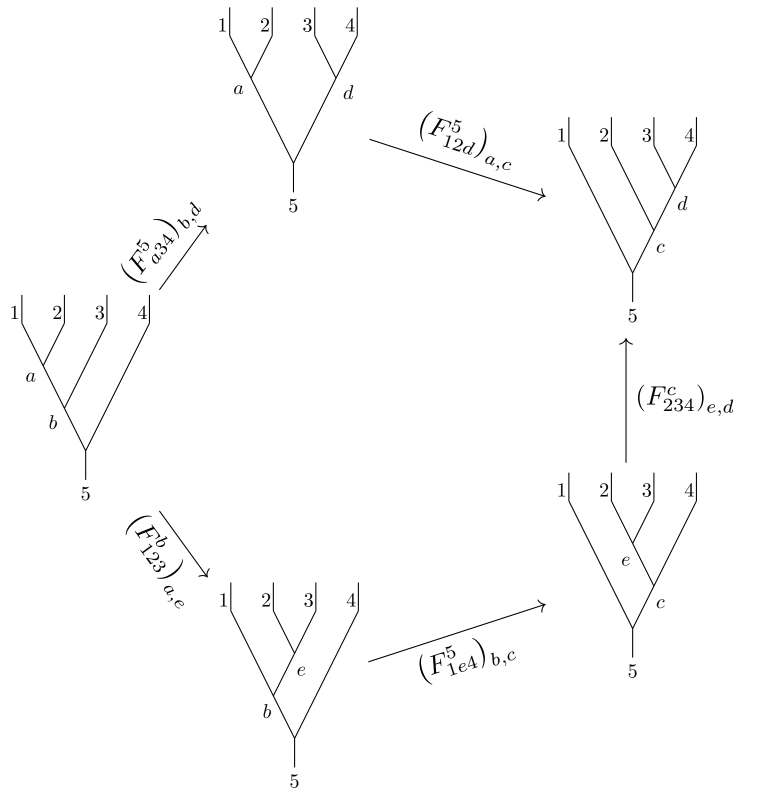

1 2 3 4
a
d
5
$\left(F^{5}_{12d}\right)_{a,c}$
1 2 3 4
d
c
5
$\left(F^{5}_{a34}\right)_{b,d}$
1 2 3 4
a
b
5
$\left(F^{c}_{234}\right)_{e,d}$
1 2 3 4
e
c
5
$\left(F^{b}_{123}\right)_{a,e}$
1 2 3 4
e
b
5
$\left(F^{5}_{1e4}\right)_{b,c}$

FIGURE 3. The pentagon equation for the 6j fusion system. The diagram must commute. It is the analogue of the pentagon equation for a fusion category.

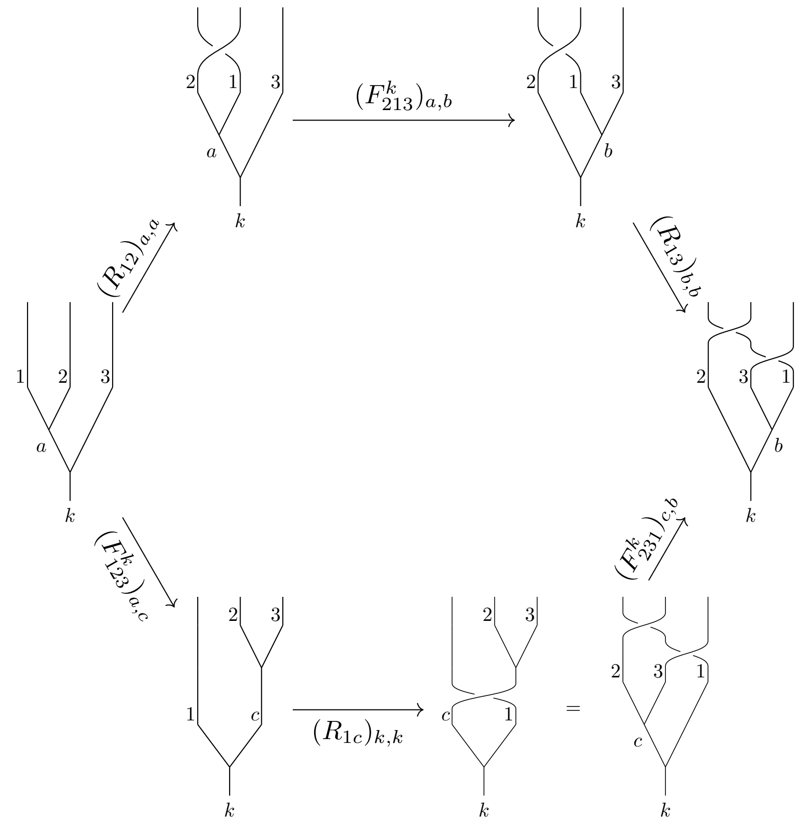


FIGURE 4. The hexagon equation for the 6j fusion system. The diagram must commute. It is the analogue of the hexagon equation for a braided fusion category.

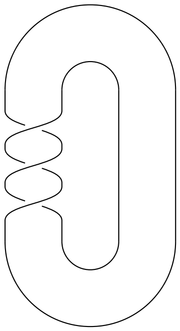


FIGURE 5. The trace closure of a braid resulting in the trefoil knot. In an appropriate ribbon unitary fusion category, this diagram corresponds exactly to $\mathrm{Tr}(\rho_A(B))$ of the trefoil knot. Physically, the value corresponds to the probability of obtaining the vacuum after pairwise fusion of the particles, created from the vacuum, after braiding.