Classical Structures in Topological Quantum Computing

Nathaniel Cabot Thomas

New College

Computing Laboratory

University of Oxford

A dissertation for the degree of

Master of Science in Computer Science

Trinity 2013

September 6, 2013

Abstract

Classical structures are a mathematical tool for describing quantum algorithms and protocols at a more abstract level than that of individual qubits. This approach is similar to using high-level programming languages with classical computers. We seek to use these tools to understand topological quantum computing, a implementation strategy for quantum computing that stores quantum information in topologically protected states and manipulates quantum information using robust topological operations. In this thesis, we identify all classical structures within the toric code, one of the simplest and best-understood models of topological quantum computation. After constructing complementary pairs of classical structures in the toric code (that is, pairs that obey a useful algebraic property), we discuss applications of these structures to implementing quantum algorithms and protocols in the toric code.

Thesis Supervisors: Dr. Jamie Vicary and Professor Bob Coecke

Acknowledgements

Thanks to Jamie Vicary and Bob Coecke for all of their time, patience, direction, insight, and enthusiasm. I've gained a new and powerful perspective on nature and computation through their teaching and mentorship.
Thanks to Tess Smidt for editing and Rob Mok for logistical support.
Thanks to the Marshall Aid Commemoration Commission and the people of the United Kingdom for funding my time at the University of Oxford through the Marshall Scholarship.
Thanks of course to my family, who are a constant source of support.

Contents

1 INTRODUCTION

The past century has witnessed an exponentially rapid increase in power and decrease in cost of technologies that enable us to store and manipulate information. This dramatic trend, often called “Moore’s Law”, has arguably enabled much of the technological progress we enjoy today. But this increase in computational power has thus far relied upon a decrease in component size, and it is unclear how this could continue much further once the components are at the atomic scale. Will progress in information technology stagnate when this limit is reached?

Quantum computation is an emerging computational paradigm focused on exploiting uniquely quantum phenomena for use in information processing that may enable us to avoid this stagnation. Since quantum phenomena become increasingly important as length scales of components are decreased, employing quantum techniques will be necessary even to reach this limit. Furthermore, quantum information technology seems to allow massive speedup for certain classical computational problems.

One of the most pressing challenges of experimentally implementing a quantum computer is the challenge of robust quantum information storage and manipulation. In even the state-of-the-art physical implementations of quantum information processors, extensive machinery and fastidious care are required to prevent the computational system of interest from becoming coupled to the surrounding laboratory environment. This coupling generally leads either to errors in the storage or manipulation of quantum information or to the corruption or destruction of a coherent quantum state in a process called ‘decoherence.’ No implementation strategy for performing quantum computation that is currently physically realizable manages to avoid all these pitfalls.

Theoretical physicist and quantum computer scientist Alexei Kitaev proposed an approach to quantum computation in 1997 that addresses these problems in a new way [16]. This approach, called ‘topological quantum computing’, requires the creation of anyons, which are quasiparticles (collective excitations of electrons in a solid) confined to a surface that transform according to the representation of a group when interchanged. The swapping of anyons creates a braid in spacetime, the topology of which encodes the relevant features of the quantum state. By creating pairs of anyons from the vacuum state, we may initialize an initial quantum state upon which to operate. Braiding these particles in a particular way allows us to perform a computation upon that initial quantum information. Finally, annihilating particles together allows us to measure the outcome of that computation and obtain its output.

If a topological quantum computer could be created, it would avoid many of the problems that most implementations of quantum computers suffer. When the quantum state space depends only upon topology (that is, the quantum information stored depends only upon the topological configuration of the constituent anyons), the information these states encode is non-local. Generally, errors that occur in quantum computations are the result of local processes, so topological quantum state storage is inherently highly resistant to uncorrectable errors.

This research is at the interface of computer science, physics, and mathematics. The previous research that has been done on these problems is often written using the terminology within one or another of these fields. The approach taken to understand this topic employs the machinery of modular tensor categories and graphical calculi.

In this thesis, we categorize indecomposable classes of classical structures that can be embedded within the toric code. Informally, classical structures capture the notion of classical bits, embedded within the quantum framework; speaking more precisely, classical structures are special Frobenius algebra-coalgebra pairings (‘special’ is used here as a technical term, which will be defined later). These structures corroborate the fact that classical information can be copied (represented by the comultiplication operation), unlike quantum information, which cannot be copied, as stated in the Quantum No-Cloning Theorem. There exist non-trivial classical structures in the toric code model, which corresponds to the quantum double group of $\mathbb{Z}_2$.

The next section will introduce the notion of quantum computation and the promise of the topological quantum computation approach. This is followed by a brief discussion of the mathematics of topological quantum computers: structures called 'unitary modular tensor categories'. Section 4 defines classical structures and identifies all classical structures in the toric code. Section 5 discusses applications of these classical structures. Section 6 outlines some of the many possibilities for further research.

2 QUANTUM COMPUTATION

We will first review the motivation for quantum computation in general, then focus on topological quantum computation.

2.1 Quantum computation in general

A quantum computer is a computer that can use the full range of phenomena described by quantum theory in its computing processes. Memory states in a classical, or digital, computer can be written as a string of 0s and 1s. In a quantum computer, however, states are described by a linear superposition of (in general) all possible strings of 0s and 1s of some length n. (This is called an n-qubit state.) This superposition is an element in a 2^n-dimensional complex Hilbert space.

This scheme seems to have much more power than classical computers because a single state stores an exponential number of complex numbers (exponential in the number of qubits). However, this is somewhat misleading. A probability distribution over n bit states is similar, just with real numbers between 0 and 1 instead of complex numbers. In neither probabilistic nor quantum computation can each of these numbers be directly accessed, so neither of these approaches provide direct access to some kind of free parallel processing power.

What quantum theory does allow that is an important difference from probabilistic computation is a type of ‘negative’ probabilities are allowed. Speaking very roughly and intuitively, these negative probabilities allow algorithms that first generate a superposition over all possible bit strings, then by exploiting the structure of the problem in question, subtract probability amplitude from non-solution bit strings.

The idea of quantum computation was first suggested by Richard Feynman [13]. The problem he hoped this type of computer could solve is that of the simulation of quantum systems. This idea was proven to be theoretically sound by Seth Lloyd [18]. To efficiently simulate a quantum system with n classical states seems to require a digital computer to store 2^n complex numbers (or be forced to cleverly dissect the system into subunits that do not strongly interact). In contrast, a quantum computer could simulate this system using n qubits. This is a titanic difference in storage. To accurately store even a system consisting of 100 strongly interacting electrons might use $O(2^{100}) \approx O(10^{30})$ bytes of memory, which easily exceeds the storage capacity of all digital computers, ever. In a quantum memory, only $O(100)$ qubits are required. As challenging as making functional quantum memory has been for researchers, it seems at least conceivable that 100 or 1000 qubits could be simultaneously manipulated. The same cannot be said of a 2^{1000} digital bit memory.

Since Feynman’s initial proposal, the idea of quantum computation has been explored and refined. The computational model most generally employed has three steps [12]: initialization, unitary evolution, and measurement. We should be able to reliably initialize the computer to a known state, which is a ray in the above-mentioned complex Hilbert space. Physically implementable evolution (that only involves interactions within the computer and not outside of it) is described by a unitary matrix operator (which is a function of time). This unitary matrix rotates the state vector in the Hilbert space. Finally, the result of the computation is determined by measuring a physical observable of the final state.

Other important algorithms exist besides the simulation algorithm. Most famous is Peter Shor's factoring algorithm [24], which showed that a quantum computer can factor large numbers in a polynomial number of standard quantum operations, or 'gates'. This result generated intense interest in quantum computation because of the importance of factoring numbers in the RSA public key cryptography algorithm, which is widely used for secure commincation on the Internet. The RSA algorithm is only effective because there is no known classical algorithm for factoring a large composite number into its two prime factors – A quantum computer with a few hundred qubits could defeat the this cryptosystem.

In addition to quantum simulation and the Shor factorization algorithms, quantum algorithms and protocols exist for quadratically faster search through unstructured databases (the Grover algorithm), transmission of quantum information using a fixed number of classical bits ('quantum teleportation'), and secure cryptographic key distribution ('quantum key distribution'), among others.

An important disclaimer is in order: There is no proof that the quantum comptuational model is more powerful than the classical model; there is only the evidence that some faster algorithms have been demonstrated in the quantum model. Let's compare the two models: In terms of computability theory, the functions that can be computed by a quantum computer are exactly the same as those that can be computed by a classical computer. Now consider complexity theory: The relevant classes are BPP (bounded-error probabilistic polynomial time), BQP (bounded-error quantum polynomial time), NP (non-deterministic polynomial time), and PSPACE (polynomial space). These are defined as follows [1]:

- BPP: solvable by a probabilistic Turing machine in polynomial time (as a function of the input length) with error probability of less than 1/3

(polynomially reducible to arbitrarily small error probability)

- NP: solvable by a non-deterministic Turing machine in polynomial time
- PSPACE: solvable by a Turing machine in polynomial space (as a function of the input length)
- BQP: solvable by a quantum Turing machine in polynomial time with error probability of less than 1/3 (polynomially reducible to arbitrarily small error probability)

We know that $BPP \subseteq BQP \subseteq PSPACE$ ([3] and [1]), but these are weak constraints. Little has been proven about the complexity hierarchy for classical computers, and we can say no more for quantum computers. If $BPP = PSPACE$, quantum computers may have little or no advantage over probabilistic classical computers. If this is not true, then quantum computers might make tractable important problems that are otherwise intractible and essentially impossible (generally, that means problems outside of BPP).

One of the primary reasons that a large-scale quantum computer has not yet been constructed is the high rate of errors that plague any system involving the creation of coherent quantum states. Consider examples of implementations of qubits such as ion traps or superconducting Josephson junctions. For states to remain coherent in these systems, there must be nearly no interaction between the quantum memories and the environment. Furthermore, the unitary operations that constitute the computation must be executed at high levels of precision – If a particular gate is only precise to within 1 part in 1000, then any computations longer than 1000 of such gates may be unreliable.

How do we deal with errors? Two general strategies exist: Use error-correcting software (called 'quantum error-correcting codes') or error-resistant hardware. Calderbank, Shor, and Steane developed a quantum error-correcting code [21]. These codes enable us to successfully perform arbitrarily long quantum computations when error rates (prior to the use of the error-correcting scheme) are less than or on the order of 10^{-4} to 10^{-6} per gate [1]. Achieving this low error rate is unfortunately a formidable experimental obstacle.

Digital computers use a combination of error-resistant hardware (via the digital abstraction in microprocessor design) and error-correcting software (such as parity bit checks). Quantum computation may benefit significantly from advances in error-resistant hardware for qubit storage and manipulation. The most developed candidate for this error-resistant hardware is to store qubits in topologically-protected states and use topological operations to execute unitary operations. This is called 'topological quantum computation', and will be the focus of the rest of this thesis.

2.2 Topological Quantum Computation

Topological quantum computation is a paradigm of quantum computation where we use topologically non-trivial quantum systems to achieve more robust quantum information storage and manipulation.

A basic requirement for a working error-resistant quantum computer is that the state of the system remain in a designated subspace of the whole Hilbert space of possible states. The stationary states in a quantum system are its energy eigenstates, that is, the eigenstates of the Hamiltonian. In general, quantum systems will have an infinite tower of energy eigenstates that they could be in. We only want the system to be in one of a particular finite subset of these eigenstates; all other states are considered illegal and are therefore errors. A necessary quality of an error-resistant quantum system is one for which transitions out of this legal subset are suppressed.

How do we suppress these transitions? We take the most naive approach: Construct a system with n degenerate (or very nearly degenerate – that is, much smaller than the energy corresponding to the timescales on which the computer will operate) energy eigenstates at the lowest energy (without loss of generality, we will set this energy to zero). These will be our legal states. The lowest illegal state will have energy ΔE above the lowest state. If ΔE is large – much larger than the energy corresponding to the timescales on which gates are executed – then transitions out of the legal subspace will be highly suppressed.

Now that we have found a condition that keeps legal states from transitioning into illegal states, we want to find how to perform stable and robust operations on our n dimensional legal subspace. This question will be answered by investigating the type of system we have constructed. If ΔE is sufficiently large, we may approximate our system as having a Hamiltonian of $H = 0$, as long as we only are manipulating states that lie within the ground state subspace. According to the Schrödinger equation

$$i\hbar\frac{\partial}{\partial t}|\psi\rangle = H|\psi\rangle,$$

this means our states will not undergo any continuous evolution. This certainly seems to lessen the possibility of unwanted transitions between states, but how do we perform computations?

Since our requirements do not permit continuous evolution of the quantum system, we consider discrete dynamics. These are possible when there are identical particles in the ground state that may be interchanged without changing the overall energy. This can be achieved (to a good approximation) by moving them adiabatically and keeping them well-separated.

The dynamics of such a system become illuminated when we employ the path integral formulation instead of the Schrödinger formalism. Then it becomes clear that the amplitudes only depend upon the topology of the paths of these identical particles. The action decomposes into components corresponding to distinct topologies, and these components are otherwise constant. Such a theory is called a 'topological quantum field theory.' Unlike general quantum field theories, a mathematically precise formulation of topological quantum field theories exists [26].

For point particles in 3+1 dimensions (meaning 4 dimensions: 3 spatial dimensions and 1 time dimension) where the only topologically relevant features are the point particles, there are only two possible operations that can occur when two identical particles are exchanged. This is because exchanging two identical particles twice is topologically identical to moving a single particle in a loop that winds around the other particle. This loop can always be homotopically deformed to a point, which means this interchange is topologically trivial. Since only topological operations change the action, the action is invariant under two swaps, so $S^2 = I$, yielding possible eigenvalues ± 1. This yields the familiar result that fundamental particles are either fermions (whose two-particle wavefunctions receive a multiplicative factor of -1 when interchanged) or bosons (whose wavefunctions are unchanged; that is, are multiplied by $+1$).

This is sufficiently rich to generate some of the most interesting phenomena in physics, such as the Pauli exclusion principle. However, in 2+1 dimensions, far richer phenomena result from considering only topological operations: In 2+1 dimensions, the double interchange of two identical particles is no longer topologically trivial – in fact, they can in gerneral be exchanged an arbitrary number of times. Adding more identical particles makes the range of possiblities even richer.

The simplest of these new possibilities is that a complex phase factor $e^{i\phi}$ is acquired when two of these identical particles are interchanged by, say, counterclockwise adiabatic rotation. This phase can have any value of $\phi \in [0, 2\pi)$. Thus m such rotations yields a phase factor of $e^{im\phi}$. Such particles are called 'anyons' – The joke is that a class of such particles can have any phase (not just ± 1) when swapped. This type is labeled 'Abelian' because in a collection of many of these particles, the order of such braiding operations is irrelevant – The overall phases ϕ are simply added together to yield the overall phase.

A helpful analogy to anyon physics is the Aharonov-Bohm effect. In an electromagnetic system, when charged particles are passed by a magnetic solenoid (with negligible magnetic field outside of the solenoid – this can be thought of as an isolated magnetic charge), they will acquire a path-dependent phase, part of which is attributable to whether a component of the charged particle wavefunction passed above or below the solenoid, or wound around the solenoid clockwise 17 times, or any other possible topologically distinguishable configuration. Anyonic phenomena are similar to Aharonov-Bohm phenomena, but now we consider both particle and solenoid to behave like charged particles that we may move as we like.

More interesting than Abelian anyons are non-Abelian anyons – These generally yield a unitary matrix operation on the wavefunction when particles are braided. These matrices are in general non-commutative, so many possible matrix operations may result from different orders of braiding operations. If the resulting set of possible matrix operations is dense in the Lie group $U(N)$, where N is the number of anyons, then we can approximate (with arbitrary precision) any unitary matrix operation on the wavefunction of the system by braiding the anyons in the appropriate way. A quantum computer is universal if it can perform any unitary operation during its computation, so non-Abelian anyon systems with this property can be used to perform any quantum computation. In fact, in anyon models that enable universal quantum computation, it is possible to keep all anyons but one static and move only a single anyon to execute all braiding patterns [25].

The mathematics of anyon braiding is encapsulated in the representation theory of the braid group, which we will now briefly explain. The 'braid group' of N strands is defined by combinations of simple braid elements: σ_i is a group element that corresponds to switching strand i with strand $i+1$ by moving the first over the second. This corresponds to rotating two neighboring anyons around each other. The inverse of this operation is switching the strands by instead moving the second over the first. When these group elements obey the following relations (the second of which is illustrated in Figure 1)

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

$$\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1} \text{ for all } i,$$

then they define the braid group.

The unitary operations that act on the wavefunctions of anyons when braided

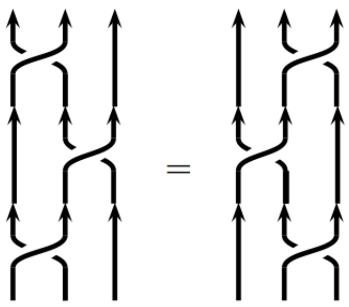


Figure 1: The non-trivial braid group relation (figure from [20])

are exactly representations of the braid group. One-dimensional representations of the braid group correspond to Abelian anyons, and higher-dimensional irreducible representations of the braid group correspond to non-Abelian anyons.

It is important to note that the topological quantum computation paradigm is neither more nor less powerful than the quantum circuit model (the model most commonly used to describe standard quantum computation). This means that a topological quantum computer can efficiently simulate (that is, with at most a polynomial increase in complexity) a sequence of gates proscribed in the quantum circuit model [14]. Conversely, a quantum circuit can efficiently simulate a computation performed by a topological quantum computer. This means that the complexity classes of topological quantum computers exactly coincide with those of any other quantum computer.

In our discussion of quantum computation in general, we said that the three stages of a quantum comptuation are initialize, apply a unitary operation, and measure part or all of the final state. We now describe how this maps onto the topological quantum computing paradigm in particular. To initialize a topological quantum computer, we create anyon quasiparticle–anti-quasiparticle pairs from the vacuum. To execute unitary operations, we braid the anyons we have created in a particular way, as discussed earlier. To measure our result, we might do some combination of bringing some anyons closer together and measuring how the energy of the system changes, allowing some pairs to annihilate, and performing anyon interferometry.

2.3 Topological Phases of Matter

There are no fundamental anyons in nature, but anyons can emerge as quasiparticle excitations in topological phases of matter. A phase of matter is labeled ‘topological’ if its low-energy effective field theory is a topological field theory. Topological phases of matter that can perform topological quantum computation have not yet been realized in the laboratory, but candidates include fractional quantum Hall effect materials and frustrated magnetic systems. There is a lot to say about the experimental realizations of topological phases of matter, but this is outside the scope of this thesis.

Despite the fact we are investigating topological quantum computation as a means to mitigate errors in quantum computers using hardware, some types of errors may still occur. Here are two examples: Fluctuations occur in finite temperature systems; these fluctuations can lead to anyon pair creation, which could braid in undesireable ways with our other anyons without our knowledge, corrupting the calculation. Second, topological charges could become trapped (unbeknowst to us) on defect or impurity sites within our topological material – Our other anyons would braid around these charges, again corrupting the computation. These errors would have to be supressed, avoided, or reliably detected to achieve working, error-free topological quantum computation.

2.4 Toric code quantum liquid model

Here we will introduce the model that we will use as our primary example throughout this thesis: the toric code. It is called a ‘code’ because it is a physical implementation of an error-correcting code. The toric code as a physical model was first discussed in the context of quantum computation by Kitaev [16]. Our description here will follow Pachos [22]; see either of these sources for further details. In the next section on unitary modular tensor categories, we will give a mathematically precise description of the toric code.

The toric code is a model of a particular type of quantum spin liquid. A quantum liquid is a liquid dominated by quantum fluctuations, as opposed to thermal fluctuations; a quantum spin liquid is a quantum liquid where the degrees of freedom are spins.

The toric code quantum spin liquid is constructed on a square lattice that lies on the surface of a torus (hence the name). Spin-$\frac{1}{2}$ objects lie at the midpoints of the links between lattice vertices. Each vertex is surrounded by 4 spins. Similarly, each of the squares defined by the lattice, which we call 'plaquettes', are bordered by spins in a similar manner. (We call the lattice defined by the centers of these squares the 'dual' of the original lattice.) This can all be seen in Figure 2.

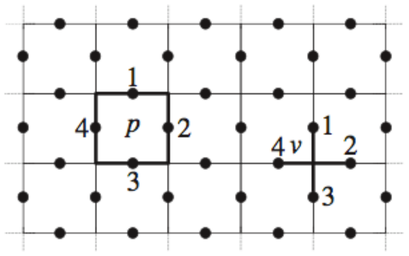


Figure 2: Toric code lattice, including vertices and plaquettes (figure from [22])

The Hamiltonian that defines the toric code is

$$H = -\sum_v A(v) - \sum_p B(p)$$

where

$$A(v) = \sigma^x_{v,1}\sigma^x_{v,2}\sigma^x_{v,3}\sigma^x_{v,4}$$

and

$$B(p) = \sigma^z_{p,1}\sigma^z_{p,2}\sigma^z_{p,3}\sigma^z_{p,4}.$$

The $\sigma^j_{v,i}$ denotes the operator that acts upon the ith neighbor of vertex v with the jth Pauli spin operator and is the identity operator on all other spins. A similar definition applies to $\sigma^j_{p,i}$ for plaquette p. The Pauli spin operators are defined as

$$\sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Let's find the ground state(s) and excited energy states of this Hamiltonian. One of the lowest-energy states is given by

$$|\Phi\rangle = \prod_v \frac{1}{\sqrt{2}}(I + A(v))|00...0\rangle,$$

where I is the identity operator. We will identify other states with the same energy soon. Note that $A(v)$ and $B(p)$ commute with each other for all v and p, and that $|\Phi\rangle$ is an eigenstate with eigenvalue 1 of the $A(v)$ and $B(p)$ operators for each vertex and plaquette.

What are the quasiparticle excitation in the toric code? We say that a state $|\Psi_e\rangle$ of the toric code system contains a quasiparticle of type e located at vertex v if

$$A(v)|\Psi_e\rangle = -|\Psi_e\rangle.$$

Similarly, we say that a state $|\Psi_m\rangle$ of the toric code system contains a quasiparticle of type m located at plaquette p if

$$B(p)|\Psi_m\rangle = -|\Psi_m\rangle.$$

So the e quasiparticles live on the vertices, and the m quasiparticles live on the plaquettes. If we bring an e and an m quasiparticle together so the e is located on one of the corners of the plaquette containing the m particle, we may treat the two as a single excitation of a new type, which we will call an ϵ quasiparticle.

How do we create excited states from this ground state? We create excited states by producing pairs of anyon quasiparticles, and we produce pairs by acting on a ground state spin with a Pauli operator. If we act on a single spin with a σ^x operator, we create a pair of e quasiparticles located at the neighboring vertices (call them v_1 and v_2), as we see that

$$A(v_i)\sigma^x_*|\Phi\rangle = -\sigma^x_*|\Phi\rangle$$

for $i = 1, 2$, where by σ^x_* we mean an operator that is the identity at every spin except for the one located between v_1 and v_2. Analogously, if we act with a σ^z operator, we create a pair of m quasiparticles located in the two neighboring plaquettes. A system with 2 quasiparticles is at a fixed and finite energy above the ground state energy (specifically, 2 units of energy, as two of the A or B operators now have eigenvalue -1 instead of 1, and the Hamiltonian is the negative of the sum of all the As and Bs). It therefore satisfies that necessary (but not sufficient) requirement for a system to be topological.

Once we have created anyons, we can move them around (see [22] for details). Creating and moving quasiparticles in the toric code is illustrated in Figure 3.

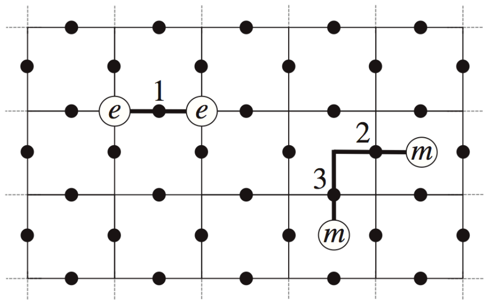


Figure 3: Creation and movement of e and m quasiparticles (figure from [22]) – A σ^x operator at 1 creates a pair of e quasiparticles, and a σ^z operator creates a pair of m quasiparticles

Like we can produce pairs of quasiparticles, we can fuse pairs by bringing them together. For example, when we bring two e quasiparticles onto the same vertex, they annihilate, leaving no quasiparticles. The rules for how quasiparticles fuse ('fusion rules') are the following:

$$e \otimes e = m \otimes m = \epsilon \otimes \epsilon = I, e \otimes m = \epsilon, e \otimes \epsilon = m, m \otimes \epsilon = e,$$

where I denotes the local state with no particles and $\otimes$ is a commutative operation that represents the physical process of bringing two quasiparticles together.

If we fuse a pair of e quasiparticles without moving them far from their initial locations, the system returns to the ground state $|\Phi\rangle$. However, if we transport one of the quasiparticles all the way around the torus before fusing, we obtain a topologically distinct ground state. It is a ground state because there are no excitations, but it is topologically distinct because the loop created by the quasiparticle pair is not contractable to a point. We can create two other ground states in this manner (by making a loop in the other direction, and by making a loop in each direction), for a total of 4 ground states. It turns out that loops operators constructed with m or ϵ quasiparticles can be written as linear combinations of these 4 loop operators (where we are including the no-loop operator) [22].

Since we have 4 distinct ground states, we can use this system as a quantum memory to store 2 qubits: we could call the 4 ground states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$. It is a robust quantum memory because any errors on a few isolated spins can be corrected. To cause an error that can't be detected and corrected, errors must occur such that they form (or nearly form) a loop. But the probability of this happening becomes exponentially smaller as the size of the torus is increased. Therefore if our local error rate is reasonably small and our torus size can be made large, the toric code can be used as a robust quantum memory. A higher genus surface can store more quantum information because there are more possible loop combinations.

It can be shown by straightforward (if long) calculations where we manipulate the spins that the e and m particles are bosons – by interchanging two by incrementally moving each particle, we find that the state picks up no phase factor. In contrast, these same type of calculations show that the ϵ quasiparticles are fermions, and that looping an m around an e (or vice versa) also produces a factor of -1. This demonstrates concretely that 2+1 dimensional anyon models can have new kinds of particle statistics: Here, the two types of bosons can act like fermions when brought together or looped around each other, phenomena that don't occur in 3+1 dimensional systems.

The toric code is the simplest non-trivial quantum double model. It is the quantum double of the group $\mathbb{Z}_2 = \{1, -1\}^\times$, denoted $D(\mathbb{Z}_2)$. Quantum double groups $D(G)$ can be defined for any group G: Each lattice site corresponds to a superposition over all of the elements of the group G, instead of just superpositions of 0 and 1 (the elements of $\mathbb{Z}_2$). We will focus on the toric code $D(\mathbb{Z}_2)$ as our primary example in this thesis, but future work may extend to more general quantum double groups.

3 UNITARY MODULAR TENSOR CATEGORIES

Unitary modular tensor categories (UMTCs) are mathematical structures that will help us understand the physics of topological quantum field theories; using these mathematical structures can help us design better algorithms and protocols within the topological quantum computing paradigm. Grounding the discussion of quantum algorithms in categorical language is useful when the underlying category is that of finite-dimensional Hilbert spaces (called **FHilb**). We would like to know whether results extend when considered in the context of modular tensor categories.

The reason we are interested in unitary modular tensor categories in particular is because all modular tensor categories leads to a 2+1 dimensional topological quantum field theory (actually, each modular tensor categories leads to two 2+1 dimensional topological quantum field theories) [26]; the converse relationship, that all 2+1 dimensional topological quantum field theories can be extended uniquely to a compatible modular tensor category, is conjectured [28].

We will employ the graphical calculus described in [7] and [15].

3.1 Basic categorical definitions

A *category* $\mathcal{C}$ is comprised of

- a collection of objects $Ob(\mathcal{C})$

- for each A,B in $Ob(\mathcal{C})$, a collection of morphisms $\mathcal{C}(A,B)$
- for each f in $\mathcal{C}(A,B)$ and g in $\mathcal{C}(B,C)$, a morphism $g \circ f$ in $\mathcal{C}(A,C)$
- for every A in $Ob(\mathcal{C})$, a morphism id_A in $\mathcal{C}(A,A)$

such that $h \circ (g \circ f) = (h \circ g) \circ f$ and $f \circ id_A = id_B \circ f$.

As an example, in the category **Set**, the objects are sets, the morphisms are functions between sets, the identity morphism on any object is the identity function on that set, and morphism composition is given by function composition.

Our primary running example will be **FHilb**, the category of finite-dimensional Hilbert spaces. In this category, objects are finite-dimensional Hilbert spaces, morphisms are bounded linear maps, the identity morphism is the identity map, and composition is given by composition of linear maps.

The structure of a category contains objects and all the relationships they share, encoded in the morphisms. The most discussed relationship between caetories is that of a 'functor'. Given categories $\mathcal{C}$ and $\mathcal{D}$, a *functor* $F : C \to D$ is defined by the following:

- for each object $A \in Ob(\mathcal{C})$, an object $F(A) \in Ob(\mathcal{D})$
- for each morphism $f \in \mathcal{C}(A,B)$, a morphism $F(f) \in \mathcal{D}(F(A), F(B))$

such that $F(id_A) = id_{F(A)}$ for every $A \in Ob(\mathcal{C})$ and $F(g \circ f) = F(g) \circ F(f)$ for all morphisms $f \in \mathcal{C}(A,B)$ and $g \in \mathcal{C}(B,C)$ (given any A, B, C in $Ob(\mathcal{C})$).

For functors $F, G : \mathcal{C} \to \mathcal{D}$, a *natural transformation* α assigns (for each $A \in Ob(\mathcal{C})$) a morphism $\alpha_A : F(A) \to G(A)$ such that $\alpha_B \circ F(f)(F(A)) = G(f) \circ \alpha_A(F(A))$. If for all $A \in Ob(\mathcal{C})$, α_A is an isomorphism, then α is called a 'natural isomorphism'.

We now define a notion of two categories having essentially the same mathematical structure: An *equivalence of categories* between two categories $\mathcal{C}$ and $\mathcal{D}$ is a pair of functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ such that $F \circ G$ is naturally isomorphic to $Id_{\mathcal{D}}$, and $G \circ F$ is naturally isomorphic to $Id_{\mathcal{C}}$.

3.2 Monoidal categories

A *monoidal category* is a category $\mathcal{C}$ along with the following data:

- a functor $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ called the *tensor product*
- a distinguished object $I \in Ob(\mathcal{C})$ called the *unit object*
- a natural isomorphism $\alpha_{A,B,C} : (A \otimes B) \otimes C \to A \otimes (B \otimes C)$ for all $A, B, C \in Ob(\mathcal{C})$
- a natural isomorphism $\rho_A : A \otimes I \to A$ for all $A \in Ob(\mathcal{C})$
- a natural isomorphism $\lambda_A : I \otimes A \to A$ for all $A \in Ob(\mathcal{C})$

that satisfies the *Coherence Property*: all well-formed equations constructed from $\circ$, $\otimes$, identity functions, and the above natural isomorphisms and their inverses are satisfied. A property that involves all well-formed equations may sound challenging to check, but fortunately the Coherence Theorem due to Mac Lane greatly simplifies this task [19].

Monoidal, or ‘tensor’, categories are useful for describing combined systems that can be operated upon in parallel. Morphism composition denotes serial operations and tensor composition denotes processes that occur in parallel and systems that are treated as side-by-side.

This sort of description is used all the time in physics, especially in quantum physics, which is why the paradigm of monoidal categories is useful in this case. We can naturally give **FHilb** a monoidal structure: The tensor product is the Kronecker product $\otimes$, the unit object is the 1-dimensional Hilbert space (identical to the complex numbers without 0), α is trivial because the Kronecker product is associative, and ρ and λ are defined by scalar multiplication.

Unitary modular tensor categories are monoidal categories with additional structure and properties. For full technical details, see [28]. We will give an informal summary of these properties:

- **Braided** This provides a swapping (in 3 dimensions, like a braid) between neighboring objects connected by a tensor product. An R-matrix describes how a braided pair of systems is related to an unbraided pair. The *symmetric* property means that the 3rd dimension is irrelevant (braiding over is the same as braiding under). **FHilb** has this property, but the categories that describe anyons do not.

- **Rigid** Rigidity means there is a dual object to every object. This allows us to capture the notion of antiparticles.

- **Twisted** Introducing a twist structure allows us to incorporate the non-trivial transformations to the wavefunction that result when a particle is rotated by 2π.

- **Abelian and semisimple** This means that there are simple objects (analogous to irreducible representations, they denote the anyons for our applications), and every object can be constructed from a finite direct sum (or biproduct $\oplus$; see [19]) of simple objects. There are fusion rules associated with every pair of objects, such as $e \otimes m = \epsilon$.

- **Modular** This implies that the braiding is non-trivial, in a specific sense.

- **Unitary, or †** This implies that there is a functor † from the oposite of the category (see [19]) to the original category. When applied twice, the † operator gives the identity. In the graphical calculus, it flips a diagram, bottom-to-top.

3.3 Toric code category

We are specifically interested in quantum double categories $D()$. The definition of quantum double categories can also be found in [28]. We will deal with only $D(1) =$ **FHilb** (where 1 denotes the trivial group) and $D(\mathbb{Z}_2)$, which is the toric code.

We provide for completeness the full mathematical data that specify the toric code quantum double unitary modular tensor category (from [23]):

Anyon types: $\{I, e, m, \epsilon\}$

Fusion rules: $e \otimes e = m \otimes m = \epsilon \otimes \epsilon = I, e \otimes m = \epsilon, e \otimes \epsilon = m, m \otimes \epsilon = e$

Quantum dimensions: $\{1, 1, 1, 1\}$

Twists: $\theta_1 = \theta_e = \theta_m = 1, \theta_\epsilon = -1$

Braidings: $R_1^{e,e} = R_1^{m,m} = R_\epsilon^{e,m} = 1, R_1^{\epsilon,\epsilon} = R_m^{e,\epsilon} = R_e^{m,\epsilon} = R_\epsilon^{m,e} = -1$

S-matrix: $= \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$

F-matrices: $F_d^{a,b,c} = (1)$ for all a, b, c, d

4 CLASSICAL STRUCTURES

We now discuss the motivation for studying classical structures. Most quantum computation and quantum information science research is discussed at the level of qubits and general unitary operations on those qubits. This seems analogous to discussing digital computer code at the level of assembly or machine language – manipulating individual bits, bytes, words, and registers. But in the case of digital computers, this level of writing code and thinking about computation is not scalable to large or interesting programs: It is difficult for humans to easily write or understand anything but the simplest programs written in such a language because it is so low-level.

Higher-level programming languages were developed so we could more easily write, reason about, and debug software. Could a similar move be as useful in the field of quantum computation? It certainly seems possible. We investigate classical structures because they are an important part of such a higher-level apparatus for understanding quantum algorithms and protocols.

Classical structures play the role of classical bits (equivalently copyable states, or orthonormal basis states) within the larger space of quantum states. This is important because even in a quantum computer we generally want to input and read out classical information from our computations or protocols. As we will demonstrate, these classical structures are crucial to the quantum algorithms and protocols that have been discovered so far, including quantum teleportation and the Deutsch-Jozsa, Grover, and hidden subgroup (including as a specific case Shor) algorithms.

The Quantum No-Cloning Theorem states that there is no physical process that can exactly clone all quantum states. However, processes exist that clone particular quantum states: specifically, a set of basis states. Concretely, in **FHilb** such a unitary is defined by $U|i\rangle|0\rangle \mapsto |ii\rangle$ for all basis states $|i\rangle$. (Note that U fails to successfully copy any superposition of basis states.)

We will define general classical structures, then define a relationship between two classical structures called *complementarity*. This relationship is useful for explaining the structure of common algorithms. We then explicitly identify classical structures and write the condition for complementarity in the toric code.

4.1 Definition of classical structure

Here we give a mathematically precise definition of a classical structure: (We will follow [8] and [15], using figures from [27].) In one sentence, a *classical structure* in a †-monoidal category is a commutative special †-Frobenius algebra. We'll define each of these new technical terms:

Before we can define a Frobenius algebra, we should define *monoids* and *comonoids* in monoidal categories.

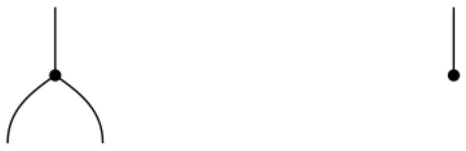
A monoid in a monoidal category $(\mathcal{C}, \otimes, I)$ is a triple (A, m, u) where $A \in Ob(\mathcal{C})$, and $m : A \otimes A \to A$ and $u : I \to A$ satisfy

$$m \circ (m \otimes id_A) = m \circ (id_A \otimes m)$$

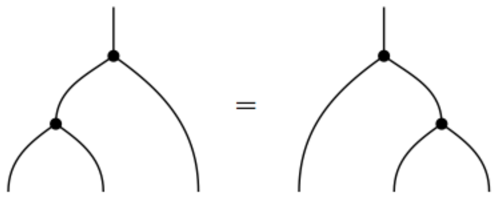
and

$$m \circ (id_A \otimes u) = m \circ (u \otimes id_A) = id_A,$$

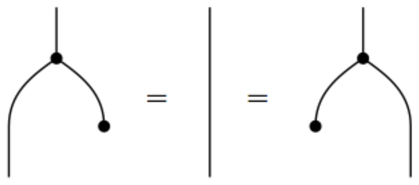
If we represent M and u as



then these conditions can be depicted as



and



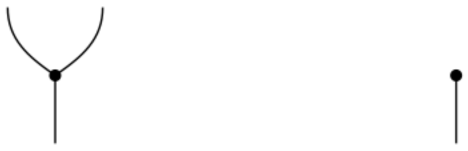
Analogously, a comonoid in a monoidal category $(\mathcal{C}, \otimes, I)$ is a triple (A, d, e) where $A \in Ob(\mathcal{C})$, $d : A \to A \otimes A$, and $e : A \to I$ satisfy

$$(d \otimes id_A) \circ d = (id_A \otimes d) \circ d$$

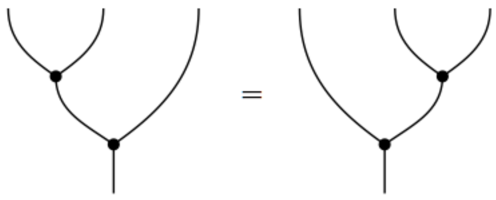
and

$$(id_A \otimes e) \circ d = (e \otimes id_A) \circ d = id_A,$$

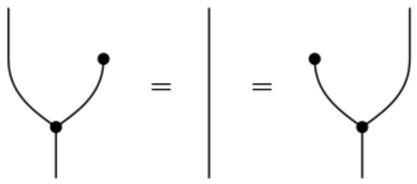
We represent d and e as



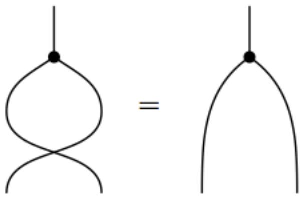
respectively. The comonoid conditions are then depicted in the following diagrams:



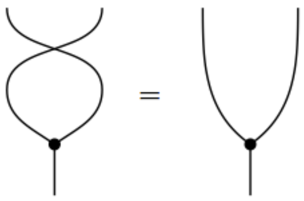
and



A monoid (A, m, u) is called *commutative* when $m \circ \text{swap} = m$. A comonoid (A, d, e) is *cocommutative* when $\text{swap} \circ d = d$. A Frobenius algebra is commutative when its monoid and comonoid are both commutative. The commutativity condition may be depicted in the graphical calculus as



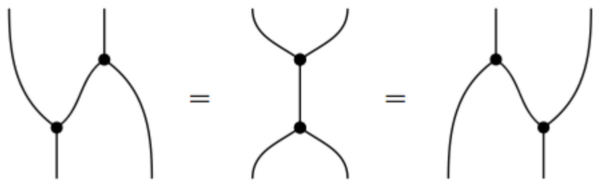
where the swap operation on the left hand side is ambiguous because it does not matter whether the braid goes over or under for the condition to be satisfied. Cocommutativity is written analogously:



A *Frobenius algebra* (which might be more appropriately called a 'Frobenius pair') in a monoidal category $(\mathcal{C}, \otimes, I)$ is a monoid and comonoid pair (A, m, u) and (A, d, e) that satisfy

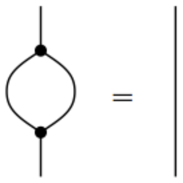
$$(m \otimes id_A) \circ (id_A \otimes d) = d \circ m = (id_A \otimes m) \circ (d \otimes id_A).$$

In the graphical calculus:



We call a Frobenius algebra a †-Frobenius algebra when $m = d^\dagger$ and $u = e^\dagger$.

A Frobenius algebra is *special* when $m \circ d = id_A$, depicted as



There is an equivalence of categories between commutative †-Frobenius algebra pairs and 1+1 dimensional topological quantum field theories [17]. We are therefore seeking to categorize (special) 1+1 topological quantum field theories that are within 2+1 topological quantum field theories.

In **FHilb**, classical structures are in a one-to-one correspondence with orthonormal bases [8]. Let $\{|i\rangle | 1 \leq i \leq N\}$ be the orthonormal basis corresponding a classical structure $(\mathcal{H}, m, u)$, where $\mathcal{H}$ is a Hilbert space of (finite) dimension N. Then

$$m\left(\sum_i D_i|ii\rangle + \sum_{i\neq j} c_{ij}|ij\rangle\right) = \sum_i D_i|i\rangle,$$

which implies

$$m^\dagger\left(\sum_i D_i|i\rangle\right) = \sum_i D_i|ii\rangle$$

for any coefficients D_i and c_{ij} that specify a legal state in $\mathcal{H}$, and

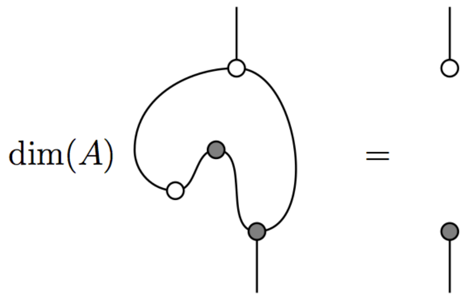
$$u(\cdot) = \sum_i |i\rangle,$$

where we have disregarded the normalization of states. It is straightforward to check that these definitions satisfy the definition of a classical structure (up to normalization of states).

4.2 Complementary bases

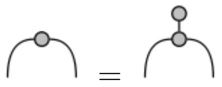
The higher-level quantum programming concept of a classical structure captures the notion of classical bits or basis states. The concept of a 'complementary bases' captures the notion of bases that are (in a particular sense) maximal superpositions of these basis states. This construct appears frequently in quantum algorithms, and we formalize it now.

Definition. Two classical structures, denoted by black and white dots, in a category $\mathcal{C}$ over object $A \in Ob(\mathcal{C})$ are *complementary* exactly when the following equation is satisfied:



(1)

where we define



for both the black and white classical structures, and for both 'cups' and 'caps'.

Once again we will use **FHilb** as an example: Let $\{|\phi_i\rangle\}$ and $\{|\psi_j\rangle\}$ be basis elements of a finite-dimensional Hilbert space $\mathcal{H}$. These bases are complementary if and only if

$$|\langle \phi_i | \psi_j \rangle|^2 = \frac{1}{\dim(\mathcal{H})}$$

for all basis element indices i, j. A simple example is the pair of bases

$$\{|0\rangle, |1\rangle\} \text{ and } \{|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\},$$

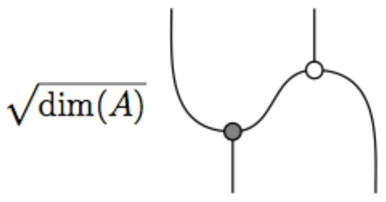
which are complementary because $|\langle + | 0\rangle|^2 = |\langle - | 0\rangle|^2 = |\langle + | 1\rangle|^2 = |\langle - | 1\rangle|^2 = \frac{1}{2}$.

Theorem 4.1. *Two classical structures are complementary if and only if*

$$\sqrt{\dim(A)} \times (id_A \otimes M_2) \circ (M_1^\dagger \otimes id_A)$$

is unitary [15].

In graphical notation, this unitary is denoted [15]



In **FHilb**, if we use classical structures corresponding to the $\{|0\rangle, |1\rangle\}$ and $\{|+\rangle, |-\rangle\}$ bases, this unitary is the CNOT gate that is frequently used in quantum algorithms:

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

4.3 Classical structures in the toric code

The results presented in these sections are novel calculations. First, we will explicitly construct 3 isomorphism classes of classical structures in the toric code. Then we will prove that these are the only indecomposable classical structures.

4.3.1 Explicit construction of classical structures in the toric code

Let's look for classical structures over the object $I \oplus e$. (As we will later discuss a bit more, the mathematics of this object will be essentially the same as that of $I \oplus m$.) To construct an algebra over this object, we need to identify monoid morphisms $M : (I \oplus e) \otimes (I \oplus e) = (I \otimes I) \oplus (I \otimes e) \oplus (e \otimes I) \oplus (e \otimes e) = 2I \oplus 2e \to I \oplus e$ and $u : I \to I \oplus e$.

We want to translate these abstract morphisms to concrete matrices. This is simplified vastly for the toric code because the associator operations are trivial; in general, we must use the appropriate F matrices when changing the associative structure. Transitions between odd and even quasiparticle-number subspaces are forbidden, so our monoid matrices will have the following form:

$$M = \begin{pmatrix} m_{II} & 0 & 0 & m_{ee} \\ 0 & m_{Ie} & m_{eI} & 0 \end{pmatrix}, u = \begin{pmatrix} u_I \\ 0 \end{pmatrix}$$

To satisfy the commutative condition, conditions for $(I \oplus e, M, u)$ to be a monoid, and Frobenius condition for monoid $(I \oplus e, M, u)$ and comonoid $(I \oplus e, M^\dagger, u^\dagger)$, we have

$$m_{II} = m_{Ie} = m_{eI} = \frac{1}{u_I}.$$

To further satisfy the specialness condition, we have

$$|m_{II}| = |m_{ee}| = \frac{1}{\sqrt{2}}.$$

A general solution to these equations yields the following monoid:

$$m = \frac{e^{i\phi}}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & e^{i\delta} \\ 0 & 1 & 1 & 0 \end{pmatrix}, u = \begin{pmatrix} e^{-i\phi}\sqrt{2} \\ 0 \end{pmatrix}$$

This defines a comonoid $(I \oplus e, M^\dagger, u^\dagger)$. Together these constitute a general classical structure in the toric code over the object $I \oplus e$.

All of these classical structures are isomporphic to each other, as we will now demonstrate. Consider two classical structures of the form above with monoids $(I \oplus e, M, u)$ and $(I \oplus e, W, n)$, which are specified by the phases ϕ_m,δ_m and

ϕ_w, δ_w, respectively. We require that there is a morphism $\sigma : I \oplus e \to I \oplus e$, which we may write as,

$$\sigma = \begin{pmatrix} s_1 & 0 \\ 0 & s_e \end{pmatrix}$$

(where s_1 and s_e are non-zero) such that

$$\sigma \circ M = W \circ (\sigma \otimes \sigma), \sigma \circ u = n$$

Such a σ is given by

$$\sigma = \begin{pmatrix} \exp\left[i(\phi_m - \phi_w)\right] & 0 \\ 0 & \pm \exp\left[\frac{i}{2}(\delta_m - \delta_w + 2\phi_m - 2\phi_w)\right] \end{pmatrix}.$$

Therefore there is an isomorphism between each of these possible classical structures.

Before considering classical structures over $I \oplus m$, we mention the following fact: The e and m objects behave in essentially the same ways. More precisely, there is a fully faithful functor (where fully faithful means that there is a bijection between the sets of morphisms; see [19] for a precise definition) F from the toric code to itself that maps $e \mapsto m$ and $m \mapsto e$. This functor allows us to construct an algebra $(I \oplus m, F(\mu), F(\eta))$ that has the same properties as $(I \oplus e, \mu, \eta)$, so the latter is a classical structure. Furthermore, the classical structures derived in this way have a one-to-one correspondence with the classical structures over $I \oplus e$. We have therefore also fully characterized the classical structures over $I \oplus m$.

Finally, we note the existence of the trivial classical structure on the object I. We are looking for monoid (I, M_0, u_0) where $M_0 : I \otimes I = I \to I$ and $u_0 : I \to I$. These functions therefore only be complex scalar multiplication. Call these multiples m and u, respectively. The classical structure conditions imply

$$|m|^2 = 1 \text{ and } um = 1,$$

which has solutions

$$m = e^{i\theta}, u = e^{-i\theta}$$

for any $\theta \in [0, 2\pi)$.

4.3.2 Proof that these are all of the indecomposable classical structures in the toric code

We say a classical structure (C, μ_C, η_C) is *decomposable* if there exist classical structures (A, μ_A, η_A) and (B, μ_B, η_B) (where neither A nor B are the zero object) such that $C = A \oplus B$. A classical structure is *indecomposable* if such a decomposition is not possible. If we classify all of the indecomposable classical structures, then we have by extention classified all classical structures because the rest can be constructed from the indecomposable classical structures.

Our classification is aided by a theorem below. To state this theorem, we must first define 2-cocycles:

Definition. Let $\gamma : F \times F \to \mathbb{C}^\times$ be a function such that for every $f, g, h \in F$,

$$\gamma(f, g)\gamma(fg, h) = \gamma(g, h)\gamma(f, gh)$$

This is called a *2-cocycle* of F. The set of 2-cycles (over $\mathbb{C}$) is denoted $Z^2(F, \mathbb{C})$

Lemma [2]. Every 2-cocycle (of F) is equivalent to one with the following properties:

1. $\gamma(e, k) = \gamma(k, e) = 1$ (This is called a *normalized* 2-cocycle.)

2. $\gamma(k, k^{-1}) = 1$

3. $|\gamma(k, l)| = 1$

4. $\gamma(k^{-1}, l^{-1}) = \gamma(l, k)^{-1}$,

for all $k, l \in F$, where e is the identity in F.

A correspondence between classical structures and properties of the group G in $D(G)$ is established in the following theorem, due to Davydov [10]:

Theorem 4.2. *Indecomposable special commutative Frobenius algebras in* $D(G)$ *correspond to quadruples* (H, F, γ, ϵ)*, where* H *is a subgroup of* G*,* F *is a normal* *subgroup of H, γ is a 2-cocycle in $Z^2(F,\mathbb{C})$, and $\epsilon : H \times F \to \mathbb{C}$ satisfies*

$$\epsilon_{gh}(f) = \epsilon_g(hfh^{-1})\epsilon_h(f), g, h \in H, f \in F$$

$$\gamma(f,g)\epsilon_h(fg) = \epsilon_h(f)\epsilon_h(g)\gamma(hfh^{-1}, hgh^{-1}), h \in H, f, g \in F$$

$$\gamma(f,g) = \epsilon_f(g)\gamma(fgf^{-1}, f), f, g \in F$$

Let's consider what this theorem implies for $D(\mathbb{Z}_2)$. We see that there are 3 candidates for (H,F) pairs, where H and F are as specified in the statement of the theorem: We have $(1,1)$, $(\mathbb{Z}_2,\mathbb{Z}_2)$, and $(\mathbb{Z}_2,1)$, where 1 is the trivial group $\{e\}$.

Now we must find which unique γ and ϵ correspond to each (H,F) pair. It turns out that there is only one unique possibility for each pair. For completeness, we will show the details of this calculation, starting with the pair $(1,1)$. By the previous lemma, we may assume that γ is normalized, so $\gamma(0,0) = 1$. Then the third condition in the theorem implies $\epsilon_0(0) = 1$. Therefore, our first quadruple is $(1,1,1,1)$.

Moving on to the pair $(\mathbb{Z}_2,1)$, we may use the previous lemma again to show that $\gamma(0,0) = 1$ by the first property. And once again the third condition in the theorem implies that $\epsilon_0(0) = 1$. By the first and second conditions respectively, we have $\epsilon_0(0) = \epsilon_1(0)^2$ and $\epsilon_1(0) = \epsilon_1(0)^2$, so $\epsilon_1(0) = 1$. Our second quadruple is thus $(\mathbb{Z}_2,1,1,1)$.

Finally, we consider perform this calculation for the pair $(\mathbb{Z}_2,\mathbb{Z}_2)$. As before, $\gamma(f,g) = 1$ for all $f,g \in \mathbb{Z}_2$ by the lemma. By the third condition of the theorem, this means that $\epsilon_f(g) = 1$ for all $f,g \in \mathbb{Z}_2$. Our third and final quadruple is therefore $(\mathbb{Z}_2,\mathbb{Z}_2,1,1)$.

We have explicitly constructed 3 indecomposable classical structures in the toric code. The theorem implies that there should be exactly 3 indecomposable commutative special Frobenius algebras in the toric code, so therefore at most 3 indecomposable classical structures (which add the requirement of being †-Frobenius). We have therefore explicitly constructed all indecomposable classical structures in the toric code.

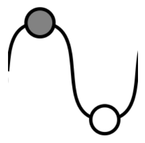
4.4 Complementary bases in the toric code

We now search for complementary bases in the classical structures corresponding to the $I \oplus e$ object. (Our results will carry over to the classical structures on the $I \oplus m$ object.) These must satisfy Equation 1, as discussed previously. Pick arbitrary parameter values for two classical structures, and determine the conditions (if any) under which they are complementary:

$$m = \frac{e^{i\phi_m}}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & e^{i\delta_m} \\ 0 & 1 & 1 & 0 \end{pmatrix}, u = \begin{pmatrix} e^{-i\phi_m}\sqrt{2} \\ 0 \end{pmatrix}$$

$$w = \frac{e^{i\phi_w}}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & e^{i\delta_w} \\ 0 & 1 & 1 & 0 \end{pmatrix}, n = \begin{pmatrix} e^{-i\phi_w}\sqrt{2} \\ 0 \end{pmatrix}$$

To simplify the calculation, we first calculate this subdiagram (figure from [15]):



which yields

$$\begin{pmatrix} e^{2i(\phi_w-\phi_m)} & 0 \\ 0 & e^{2i(\phi_w+\delta_w-\phi_m-\delta_m)} \end{pmatrix}$$

The left side of the complementarity condition in Equation 1 is

$$e^{i(\phi_w-\phi_m)} \begin{pmatrix} 2 & 0 \\ 0 & 1+e^{i(\delta_w-\delta_m)} \end{pmatrix}$$

and the right side is

$$e^{i(\phi_w-\phi_m)} \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}.$$

Therefore, the two classical structures are complementary exactly when the following condition is satisfied:

$$1 + \exp\left[i(\delta_w - \delta_m)\right] = 0$$

which can be written as

$$\cos\left[\frac{\delta_w - \delta_m}{2}\right] = 0.$$

If we set $\phi_m = \phi_w = 1$, then we obtain a pair of complementary classical structures:

$$M = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, u = \sqrt{2}\begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

This classical structure copies the states

$$|+\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ and } |-\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix};$$

that is,

$$M^\dagger(|+\rangle) = |+\rangle \otimes |+\rangle \text{ and } M^\dagger(|-\rangle) = |-\rangle \otimes |-\rangle.$$

$$W = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, n = \sqrt{2}\begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

This classical structure copies the states

$$|\leftarrow\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ i \end{pmatrix} \text{ and } |\rightarrow\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -i \end{pmatrix},$$

so

$$W^\dagger(|\leftarrow\rangle) = |\leftarrow\rangle \otimes |\leftarrow\rangle \text{ and } W^\dagger(|\rightarrow\rangle) = |\rightarrow\rangle \otimes |\rightarrow\rangle.$$

These classical structures copy states corresponding to X and Y directions on the Bloch sphere. There is no classical structure that copies the state corresponding to the Z direction because this would violate conservation of anyon charge.

5 APPLICATIONS OF CLASSICAL STRUCTURES

Many of the most important quantum algorithms and protocols can be expressed in a topological form, constructed from classical structures, using the graphical calculus [27]. These algorithms and protocols are specified over the category **FHilb**. We will begin to explore how they may be generalized to work over other modular tensor categories by giving some first results for the toric code.

5.1 Quantum teleportation protocol

We will first describe the standard quantum teleportation protocol, then discuss how an anyonic teleportation protocol might be implemented in the toric code.

5.1.1 Standard teleportation protocol

The standard quantum teleportation protocol allows one party (who we call Alice) to send a qubit to another party (who we call Bob) by sending two classical bits. This protocol requires that Alice and Bob share a bipartite (*i.e.* two-qubit) state called a 'Bell state', which is one of the following 4 states:

$$|\Phi_\pm\rangle = |00\rangle \pm |11\rangle$$

$$|\Psi_\pm\rangle = |01\rangle \pm |10\rangle$$

Alice possesses the first qubit and Bob possesses the second. We will assume that they specifically share the $|\Psi\rangle$ state. Note that this state and the other Bell states are all entangled; that is, they cannot be written as a tensor product of two single-qubit states. Suppose Alice and Bob measure their state in the computational basis (*i.e.* $\{|0\rangle, |1\rangle\}$): If she measures $|0\rangle$, then Bob is guaranteed to measure $|0\rangle$; similarly, if she measures $|1\rangle$, then Bob is guaranteed to measure $|0\rangle$.

The following is the quantum teleportation protocol: Alice and Bob share the Bell state $|\Psi\rangle$, and Alice has a qubit $|\chi\rangle$ that she wishes to transmit to Bob. First, she performs a projective measurement of her two qubits ($|\chi\rangle$ and her half of the Bell state). She gets one of four outcomes. She sends the outcome to Bob, encoded in two classical bits. Let's say they have predetermined that

00 means $|\Phi_+\rangle$ was measured, 01 means $|\Phi_-\rangle$ was measured,

10 means $|\Psi_+\rangle$ was measured, and 11 means $|\Psi_-\rangle$ was measured.

When Bob receives this information, he applies the appropriate unitary to his qubit:

$$I_2 \text{ for } 00, \sigma^z \text{ for } 01$$

$$\sigma^x \text{ for } 10, \text{ and } i\sigma^y \text{ for } 11$$

Bob then possesses a qubit in the same state as the one that Alice originally had (which was destroyed in this processes).

5.1.2 Teleporting toric code anyons

Using the complementary classical structures we have found in the toric code, we can see that the teleportation protocol, specified in the completely positive map formalism as described in [5] (using the directed graph conventions from [6]) is valid. We need to retain the directed graph conventions because the toric code classical structures form a Y/X calculus and not a Z/X calculus, as mentioned in Section 4.4. Due to the charge conservation condition, there is no way to transform one into the other.

One of the primary differences in the toric code is that to satisfy conservation of anyon charges, Alice must have a channel that sends a charge to Bob. But they don't need to maintain this charge in a coherent quantum state, only a binary 'charge / no charge' channel would be necessary. We can obtain this by bringing together the two $I \oplus e$ states at a vertex v and performing a measurement of charge state $A(v)$. The other classical channel would contain the result of a measurement of the phase between the two $I \oplus e$ input channels.

It is not yet clear how to connect the complementary classical structure formalism with the physical steps of the protocol. Anyon teleportation is discussed from a different perspective by Bonderson [4].

5.2 Deutsch-Jozsa algorithm

The Deutsch-Jozsa algorithm [11] was the first quantum algorithm discovered that was provably better than any possible deterministic classical algorithm. It solves the Deutsch-Jozsa problem: We are given a function $f : \{0,1\}^n \to \{0,1\}$ which is guaranteed to be either constant ($f(x) = a$ for all x with $a \in \{0,1\}$ a constant) or balanced (f outputs 0 for half of all possible inputs and 1 for the other half). The task is to determine whether the function is constant or balanced. For a deterministic classical computer, this can only be achieved with certainty after $2^{n-1} + 1$ evaluations of f. With a quantum computer, only a single evaluation of f is necessary. See [21] for further details.

We will show here that the Deutsch-Jozsa algorithm cannot be implemented directly in the toric code because the only permissible functions are balanced functions. This means the Deutsch-Jozsa problem cannot even be posed in a non-trivial way because only one answer is possible.

5.2.1 Standard Deutsch-Jozsa algorithm and comonoid homomorphisms

Our explanation of the details of the Deutsch-Jozsa algorithm will follow [27] because the algorithm is presented at a high level, in terms of classical structures. Functions that are permitted for the Deutsch-Jozsa algorithm must be comonoid homomorphisms.

Definition. A function $f : A^{\otimes n} \to A$, where $A^{\otimes n} = A \otimes ... \otimes A$ (n times), is a *comonoid homomorphism* for a comonoid (A, d, e) if the following equation holds:

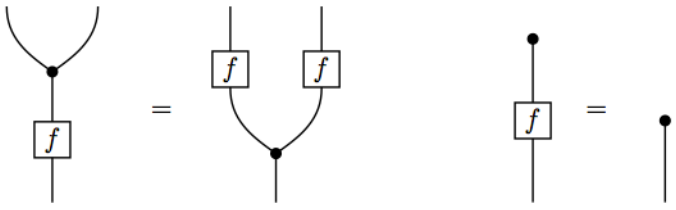
$$d \circ f = (f \otimes f) \circ P_n \circ d^{\otimes n} \tag{2}$$

$$e \circ f = e^{\otimes n}, \tag{3}$$

where

$$P_n = (id_A^{\otimes(n-1)} \otimes \text{ swap } \otimes id_A^{\otimes(n-1)}) \circ (id_A^{\otimes(n-2)} \otimes \text{ swap } \otimes \text{ swap } \otimes id_A^{\otimes(n-2)}) \circ \\ ... \circ (id_A \otimes \text{ swap}^{\otimes(n-1)} \otimes id_A).$$

This is the graphical form of the conditions for $n = 1$ (figure from [27]):



The notion of comonoid homomorphism is all that is needed to argue that the Deutsch-Jozsa problem may become trivial and uninteresting in the toric code, which we will explain in the next section.

5.2.2 The Deutsch-Jozsa problem in the context of the toric code

In the toric code, the permissible functions for the $I \oplus e$ channels are of the form

$$f_{n=1} = \begin{pmatrix} F_I & 0 \\ 0 & F_e \end{pmatrix}, f_{n=2} = \begin{pmatrix} F_{II} & 0 & 0 & F_{ee} \\ 0 & F_{Ie} & F_{eI} & 0 \end{pmatrix},$$

$$f_{n=3} = \begin{pmatrix} F_{III} & 0 & 0 & F_{Iee} & 0 & F_{eIe} & F_{eeI} & 0 \\ 0 & F_{IIe} & F_{IeI} & 0 & F_{eII} & 0 & 0 & F_{eee} \end{pmatrix}, \ldots$$

To satisfy the second comonoid homomorphism condition (Equation 3), we must have $F_I = F_{II} = ... = 1$ with all other top-row entries set to 0.

In the cases $n = 1$, $n = 2$, and $n = 3$, the functions that both obey the comonoid homomorphism property for the classical structures associated with $I \oplus e$ and not violate conservation of anyon number are exactly those that satisfy $f(x) \oplus f(1^n \oplus x) = 1$ for all $x \in \{0, 1\}^n$. We are abusing notation here: by x we mean a representation of the bits of x in the natural basis for these functions: $\{|+\rangle, |-\rangle\}$. Thus 00...0 corresponds to $|+++...+\rangle$, 100...0 corresponds to $|-++...+\rangle$, etc. We may conjecture that this equation is satisfied for all n-to-1 functions.

For any function f that satisfies $f(x) \oplus f(1^n \oplus x) = 1$ for all x, f is a balanced function. This is because for every x such that $f(x) = 0$, there is a distinct $y = 1^n \oplus x$ such that $f(y) = 1$. If the only legal functions are balanced functions, then the Deutsch-Jozsa problem is trivial and uninteresting for the toric code: The algorithm will always output the only possible answer.

Though it seems that the Deutsch-Jozsa problem is uninteresting in the context of the toric code, the Grover search algorithm, Simon's algorithm, and hidden subgroup problems in general might still be interesting. Their high-level forms in terms of classical structures are similar to the Deutsch-Jozsa algorithm [27], so understanding the toric code form of these algorithms is a logical next step.

6 SUMMARY OF RESULTS AND DISCUSSION OF FUTURE RESEARCH DIRECTIONS

In this thesis, we reviewed the motivation for quantum computation in general and topological quantum computation in particular; we reviewed basic definitions and results in monoidal category theory and discussed how unitary modular tensor categories relate to the physics of topological systems; and we introduced the concept of classical structures and explained how they are useful in quantum algorithms and protocols. This material is a literature review. The new results presented were

- the classification of all classical structures in the toric code model
- the construction of all complementary pairs of classical structures in the toric code model – the fact that complementary pairs exist immediately means that it may be possible to implement standard quantum algorithms using anyons in the toric code
- preliminary applications of these tools to implementing quantum algorithms in the framework of the toric code.

These results are just the beginning of a program to discover classical structures within topological quantum field theories and use classical structures to develop new algorithms and protocols that specially take advantage of the robustness of topological quantum computers. Here are some future directions to be pursued as a part of that program:

- Determine how other standard quantum algorithms and protocols can be implemented with classical structures in anyon models. We discussed teleportation and Deutsch-Jozsa algorithms in the toric code, but the Grover search and hidden-subgroup algorithms have similar forms to Deutsch-Jozsa [27]. The quantum key distribution protocol for cryptography relies primarily on complementary classical structures [9], so this would be another candidate.

- Search for classical structures and complementary pairs in other low-rank unitary modular tensor categories, such as the Fibonacci anyon model. This seems like a promising direction because if classical structures exist, the simplest may be discoverable by semi-automated brute-force search. Finding (or failing to find) simple classical structures in other low-rank unitary modular tensor categories may point to more general conjectures about the existence of classical structures and complementary pairs.

- Construct classical structures and complementary bases in $D(S_3)$. This model is interesting because S_3 is the smallest non-abelian group. Since a general theorem that categorizes indecomposable classical structures in quantum double models has already been produced by Davydov [10], it should be possible to determine when all indecomposable classical structures have been constructed. It seems possible that these classical structures will have novel applications because $D(S_3)$ has a richer structure than $D(\mathbb{Z}_2)$.

- Prove results about classical structures and complementary bases in quantum double categories in general.

- Determine how braiding is related to unitaries built from classical structures. Demonstrations of computational universality in anyon systems generally rely upon showing that some set of braids generates a dense subset of a unitary Lie group, implying that these braids can approximate any gate within that Lie group arbitrarily well. Are there models for which interesting gates can be implemented exactly by executing certain braids? Perhaps this question could be approached is by elucidating relationships between braids and classical structures in particular models or in general.

Topological quantum computers present the possibility of intrinsically robust quantum information storage and manipulation. Classical structures point toward the possibility of a more intuitive language for programming quantum computers. Using these approaches together, we can hope to develop new, more effective approaches to quantum computer engineering.

References

[1] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[2] S. Beigi, P. W. Shor, and D. Whalen. The Quantum Double Model with Boundary: Condensations and Symmetries. *Communications in Mathematical Physics*, 306:663–694, Sept 2011.

[3] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411–1473, October 1997.

[4] P. Bonderson. Measurement-only topological quantum computation via tunable interactions. *Phys. Rev. B*, 87(3), January 2013.

[5] B. Coecke and R. Duncan. Interacting Quantum Observables: Categorical Algebra and Diagrammatics. *ArXiv e-prints*, June 2009.

[6] Bob Coecke, Ross Duncan, Aleks Kissinger, and Quanlong Wang. Strong complementarity and non-locality in categorical quantum mechanics. *Logic in Computer Science*, 2012.

[7] Bob Coecke, Eric Oliver Paquette, and Dusko Pavlovic. Classical and quantum structuralism. *Semantic techniques for quantum computation*, page 43, 2008.

[8] Bob Coecke, Dusko Pavlovic, and Jamie Vicary. A new description of orthogonal bases. *Mathematical Structures in Computer Science*, 23:555–567, 6 2013.

[9] Bob Coecke, Quanlong Wang, Baoshan Wang, Yongjun Wang, and Qiye Zhang. Graphical calculus for quantum key distribution (extended abstract). *Electron. Notes Theor. Comput. Sci.*, 270(2):231–249, February 2011.

[10] A. Davydov. Modular invariants for group-theoretical modular data. I. *ArXiv e-prints*, August 2009.

[11] D. Deutsch and R. Jozsa. Rapid Solution of Problems by Quantum Computation. *Royal Society of London Proceedings Series A*, 439:553–558, December 1992.

[12] D. P. DiVincenzo. The physical implementation of quantum computation. *fortschritte Der Physik-progress*, 98(9-11):771, 2000.

[13] R. P. Feynman. Simulating physics with computers. *International Journal of Theor. Physics*, 21:467, 1982.

[14] Michael Freedman, Alexei Kitaev, Michael Larsen, and Zhenghan Wang. Topological quantum computation. *Bulletin of the American Mathematical Society*, 40(1):31–38, 2003.

[15] Chris Heunen and Jamie Vicary. Lectures on categorical quantum mechanics, Sept 2013. https://www.cs.ox.ac.uk/files/4551/cqm-notes.pdf.

[16] A. Yu. Kitaev. Fault tolerant quantum computation by anyons. *Annals Phys.*, 303:2–30, 2003.

[17] Joachim Kock. *Frobenius Algebras and 2-D Topological Quantum Field Theories*. Cambridge University Press, 2003.

[18] Seth Lloyd. Universal Quantum Simulators. *Science*, 273(5278):1073–1078, August 1996.

[19] Saunders Mac Lane. *Categories for the Working Mathematician*. Graduate Texts in Mathematics. Springer, 2nd edition, Sept 1998.

[20] Chetan Nayak, Steven H. Simon, Ady Stern, Michael Freedman, and Sankar Das Sarma. Non-abelian anyons and topological quantum computation. *Rev. Mod. Phys.*, 80:1083–1159, Sept 2008.

[21] Michael Nielsen and Isaac Chuang. *Quantum computation and quantum information.* Cambridge University Press, New York, NY, USA, 2000.

[22] Jiannis K. Pachos. *Introduction to Topological Quantum Computation.* Cambridge University Press, New York, NY, USA, 1st edition, 2012.

[23] Eric Rowell, Richard Stong, and Zhenghan Wang. On classification of modular tensor categories. *Commun. Math. Phys.*, 292(2):343–389, 2009.

[24] Peter W. Shor. Polynomial time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J.Sci.Statist.Comput.*, 26:1484, 1997.

[25] S.H. Simon, N.E. Bonesteel, M.H. Freedman, N. Petrovic, and L. Hormozi. Topological quantum computing with only one mobile quasiparticle. *Phys.Rev.Lett.*, 96:070503, 2006.

[26] V. Turaev. *Quantum Invariants of Knots and 3-Manifolds.* Walter de Gruyter and Co., first edition, 1994.

[27] Jamie Vicary. The topology of quantum algorithms. http://arxiv.org/abs/1209.3917.

[28] Zhenghan Wang. *Topological Quantum Computation.* American Mathematical Society, first edition, 2010.