1 Quantum Computation

1.1 Topological Quantum Computation

1.1.1 From categories to computation

In the previous section we saw that categories can be interpreted as physical process theories. In a very similar way, we can interpret objets as data types and morphisms as computational processes, so that any category corresponds to a theory of computation. Monoidal categories are ones where parallel computation is possible. Quantum computation is a model in which data is encoded in the state of quantum systems and processes are quantum evolutions of the system. A computation consists of the preparation of some quantum states, their manipulation and measurement. Usually this procedure is repeated in order to collect statistics and approximate density distributions. The unit of information in quantum computation is called qubit by analogy with the classical bit. A qubit is a two-level quantum system, that is a Hilbert space of dimension 2 which is denoted by $\mathbb{C}^2$. Similarly a qudit is a d dimensional system $\mathbb{C}^d$.
Modular categories are models for topological quantum computation (TQC) in the sense of [1] or [2]. TQC has been studied extensively in recent years as it allows for fault-tolerant quantum computation. In this model data is encoded in non-abelian anyons and quantum gates are obtained by braiding those particles. Topologically equivalent braids implement the same quantum process so that small perturbations of particle world-lines do not affect the computation and gates (and quantum information) are topologically protected from decoherence. Another reason for studying topological quantum computation is that some TQC models allow to approximate the Jones polynomial in polynomial time, a problem that is believed to be untractable classically.
The problem of building a topological quantum computer has been addressed by various authors [1] [cite freedman]. Non-abelian anyons have never been achieved experimentally, much of the difficulty arises in the two-dimensional nature of anyons.
A topological quantum computer runs as follows [2]:

Definition 1 (TQC)
1. *Creation of anyon pairs from the vacuum to encode the information as a quantum state.*
2. *Braiding those anyons performs a quantum gate on the state.*
3. *fusing neighbouring anyons and observing the resulting anyon type corresponds to a projective measurement on the system.*

The computation result is the approximation to a probability distribution (over measurement outcomes) obtained by repeating the procedure polynomially many times and recording the output anyon types. Note that if we postselct on the vacuum sector to be the output anyon type we are effetively approximating an invariant of links. Indeed any process in TQC starting and ending in the vacuum sector is a link, formed by the particle trajectories in space-time (i.e the

briading process). The operations we can perform on a system V are unitaries, so that any braiding process on n particles induces the evaluation of some unitary representation $\beta \to U_\beta$ of the braid group B_n.
In order to make sure the braiding process is a unitary transformation of the state space we will assume one further constraint on our categorical model of computation: unitarity of the braids in the modular category in question.

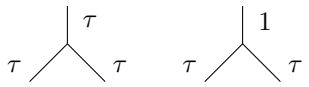
Definition 2 (UMC) *A unitary modular category (UMC) is a modular category where each component of the braiding is a unitary.*

Given a UMC $\mathcal{C}$, we have a finite set of data types I given by the isomorphism classes of simple objects in $\mathcal{C}$. The topological qudit is usually encoded in some fusion space as follows. We usually fix some data type a and consider the fusion of n copies of a. We then choose some output type b on which to post-select in order to obtain a fusion space $V := V^b_{a^{\otimes n}} = Hom(a^{\otimes n}, b)$ of dimension d. We can picture standard basis as labelled binary fusion trees with b at the root and a on the n leaves. Each binary tree shape (with n leaves) corresponds to a different bracketing of $a^{\otimes n}$ and usually yields distinct bases related to each other via F-moves. Braiding a pair of type a anyons before fusing is an R-move. All computational processes on V can be decomposed in sequences of F-moves (re-bracketing) and R-moves (braiding).

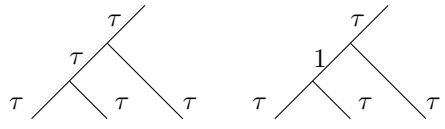
Example 1 (Fibonacci anyons) *We now look back at example [ref] on Fibonacci anyons and show how to compute in the model. Recall we have only two particle types: the vacuum sector* 1 *and the non-trivial* τ *such that:*

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

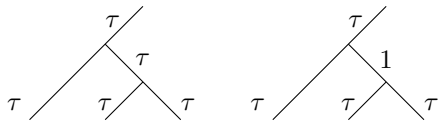
τ *and* 1 *are their own anti-particles so we don't need to distinguish particles and antiparticles by wrinting arrows on wires. We can write the basis of the two dimensional space* $\tau \otimes \tau$ *as:*



The fusion space $V^\tau_{\tau^{\otimes 3}}$ *is two dimensional, we will take it as our computational space. Practically, this corresponds to considering three* τ *particles with overall charge (type)* τ*. This space is our topological qubit and we write the computational basis as:*



Let's denote by $|0\rangle, |1\rangle$ *these basis states. Another basis is given by fusing the left-most two anyons first:*



And we denote them by $|+\rangle, |-\rangle$*. These two bases are linked by a unitary* 2×2 *transformation* $F := F^{\tau}_{\tau^{\otimes 3}}$ *given by the solution of the following system:*

$$|0\rangle = F_{0+} |+\rangle + F_{0-} |-\rangle$$

$$|1\rangle = F_{1+} |+\rangle + F_{1-} |-\rangle$$

To derive the form of the F-matrix we need to consider the pentagon axiom. It turns out that for the Fibonacci model the pentagon is enough to derive the F-matrix but it is not the case in general. The resulting F-matrix is [3]:

$$F^{\tau}_{\tau^{\otimes 3}} = \begin{bmatrix} \phi^{-1} & \phi^{-\frac{1}{2}} \\ \phi^{-\frac{1}{2}} & -\phi^{-1} \end{bmatrix} \tag{1}$$

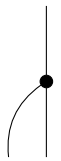
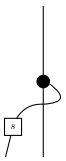
where $\phi = \frac{\sqrt{5}-1}{2}$*. Given the F-matrix and the two hexagon axioms for braided monoidal categories the possibilities for the R-matrix are few. In this case there is only one possibility and we obtain the R-matrix:*

$$R^{\tau}_{\tau^{\otimes 3}} = \begin{bmatrix} e^{-4\pi i/5} & 0 \\ 0 & -e^{-2\pi i/5} \end{bmatrix} \tag{2}$$

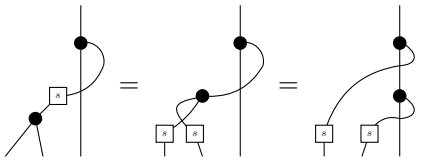
In [4] it is shown that the Fibonacci model allows universal quantum computation. This is done by first noting that polynomially many R and F matrices as above can approximate any unitary on one qubit, and then by constructing a CNOT gate on two topological qubits.

1.1.2 Kitaev's quantum double model

The models which we present here have first been introduced by Kitaev [1] and some special cases have been implemented experimentally on two dimensional materials at very low temperatures.
Suppose we have particles living in state space H where H is a Hopf algebra (with black multiplication, white comultiplication and antipode S as usual). We can define two canonical types of left H-module structures on H given by the right and left multiplication as follows:

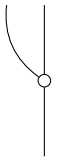
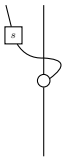
$L_+ =$  ; $L_- =$  (3)

Right multiplication defines a module by associativity, left multiplication also works because the antipode is an antialgebra morphism:



(4)

Similarly there are two canonical left H-comodule structures on H given by left and right comultiplication.

$T_+ =$  ; $T_- =$ 

(5)

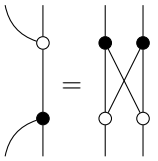
And the proofs that these are H-comodules are dual to the previous ones. Kitaev considers the case where $H = \mathbb{C}G$ for some group G. The above module and comdule structures then yield 4 types of linear operators on $\mathbb{C}G$: $L^g_\pm$, $T^h_\pm$ (using the notation from [1]), indexed by elements $g, h \in G$, and defined as follows:

$$\begin{aligned} L^g_+ |z\rangle &= |gz\rangle \quad L^g_- |z\rangle = |zg^{-1}\rangle \\ T^h_+ |z\rangle &= \delta_{h,z} |z\rangle \quad T^h_- |z\rangle = \delta_{h^{-1},z} |z\rangle \end{aligned} \tag{6}$$

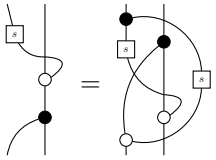
Note that L operators commute with each other and T operators too. The non-trivial commutation relations are the following:

$$\begin{aligned} L^g_+ T^h_+ &= T^{gh}_+ L^g_+ \quad L^g_+ T^h_- = T^{hg^{-1}}_- L^g_+ \\ L^g_- T^h_+ &= T^{hg^{-1}}_+ L^g_- \quad L^g_- T^h_- = T^{gh}_- L^g_- \end{aligned} \tag{7}$$

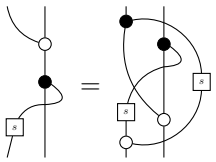
In the general Hopf algebra case those commutation relations correspond to the following equalities of diagrams, which are easily obtained using the bialgebra rule and the fact that the antipode is an anti-algebra and an anti-coalgebra morphism.



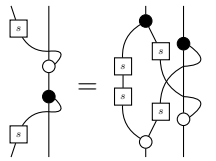
(8)



(9)



(10)



(11)

We then consider a lattice with oriented edges embedded in some $2D$ oriented manifold (e.g a sphere or a torus) with particles on the edges. For any vertex s and adjacent plaquette p, a site is defined as the pair $a = (s, p)$. Let $star(a)$ be the set of edges adjacent to s and $bound(p) = \{j_1, ..j_k\}$ the ordered set of edges adjacent to plaquette p starting and ending at vertex s. Every edge $j \in star(s)$ on the lattice has an orientation and in [1] is defined $L^g(j, s)$ to be L^g_- applied to vertex j when s is the origin of j and L^g_+ otherwise. Similarly $T^h(j, p)$ is defined to be T^h_+ (repectively T^h_-) if j is on the right (resp. on the left) of j. The following operators are then defined at each site a of the lattice:

$$\begin{aligned} A_g(a) &= \prod_{j \in star(s)} L^g(j, s) \\ P_h(a) &= \sum_{h_1 ... h_k = h} \prod_{m=1}^{k} T^{h_m}(j_m, p) \end{aligned} \tag{12}$$

We will write the general diagrammatic form of these operators in the next section. For the moment let us go through Kitaev's reasoning [1].
From a physical point of view P_h operators can be understood as measuring the magnetic flux of the system at some site and A_g are local symmetry transformations on the charge. Flux measurements are projection $P_h \in \mathbb{C}G^*$ onto flux sector h. The allowed residual global symmetry transformations are then implemented via A_g for $g \in N(h)$.
The projectors form a Von Neumann family and satisfy

$$P_h P_{h'} = \delta_{h,h'} P_h.$$

Operators A_g are global symmetry transformation

$$A_g A_h = A_g h$$

and affect the fluxes via conjugation:

$$A_g P_h = P_{ghg^{-1}} A_g \tag{13}$$

(this was shown was shown by Kitaev [1]). Operators A_g and P_h generate the algebra DG. So the quantum double construction allows to capture both global symmetry transformations and projective measurements in one algebraic structure. It is easy to check, rewriting the definition, that the following is true.

Proposition 1 *For any finite group G, its quantum double $D(G)$ is the algebra generated by $\{P_h A_g\}_{h,g\in G}$ with multiplication induced by (13), comultiplication and antipode as defined in [first section].*

$D(G)$ has a natural quasi-triangular structure witnessed by the universal R-matrix $R = \sum_{g,h\in G} P_h e \otimes P_h g$, making *RepDG* braided.
Kitaev then builds a Hamiltonian for the system and shows that the ground state of the Hamiltonian is an irreducible representations of DG. Here we will skip this part of the reasoning and rely on the intuition that the operators A_g and P_h correspond to the symmetries of the system, i.e the dynamics which are 'constantly being applied'. So the allowed processes of the systems are processes that commute with all of those operators, i.e the system lives in a representation of DG. The ground state of the Hamiltonian has degeneracy 4^g where g is the genus of the surface in which we embedded the lattice.
In the case of a sphere, $g = 0$, so there is no degeneracy and the overall system lives in a one dimensional (trivial) representation of DG, the vacuum sector. An excitation can arise at some site on the lattice when the constraints given by the Hamiltonian are violated. In representation theoretic terms this corresponds to the creation of a state of some higher dimensional irreducible representation of DG. Those excitations (or quasi-particles) are anyons and can only be created in pairs (particle-antiparticle pairs). When the lattice is 'layered' enough (i.e contains many particles) we can move those excitations on the lattice (practically this is done by applying charge and flux operators at given sites on the lattice). All the excitations can then be fused pairwise to end back in the vacuum sector and obtain fusion results. There are different possible types of excitations (anyon flavours) we can create on the lattice, corresponding to different possible violations of the constraints. These are precisely indexed by the irreducible representations of DG. We can see that we have obtained a physical setting giving rise to anyons whose behaviour is modeled by the modular category $Rep(DG)$.
In order to understand the possible anyon types in the model, we must study the irreducible representations of the quantum double finite group algebra DG. This has been done by Gould [5], who showed that irreducible representations of DG are obtained in the following way.
Let $\{C_i\}_{i=1}^n$ be the distinct conjugacy classes in G. To each of those conjugacy classes corresponds a centralizer subgroup N_i (two choices of representatives for C_i yield isomorphic centralizer subgroups). Then for any irreducible representation (α, V_α^i) of N_i with basis elements v_j^α, let $V_{i,\alpha} = \mathbb{C}C_i \otimes V_\alpha^i$, this has basis $\{|k, v_j^\alpha\rangle\}_{j=1,\dots,dim\alpha}^{k\in C_i}$ and forms an irreducible representation of $D(G)$ under the action

$$P_h g\,|k, v_j^\alpha\rangle = \delta_{h,gkg^{-1}}\,|h, \alpha(h^{-1}gk)v_j^\alpha\rangle \tag{14}$$

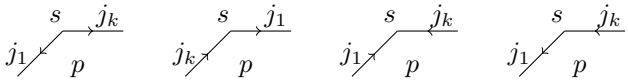
and the $\{V_{i,\alpha}\}$ is the complete set of irreducible representations. When G is abelian, all irreducible representations are one-dimensional and we only have abelian anyons which are very unlikely to be universal for quantum computation. It was shown by Kitaev that if $G = S_5$ the model is universal for quantum computation. In [cite Lahtinen] the $G = S_3$ model was considered but not shown to be universal.

Example 2 (Quantum memory) *The case where $G \simeq \mathbb{Z}_2$ gives rise to Kitaev's toric code. Note that $D(\mathbb{Z}_2) \simeq \mathbb{C}(\mathbb{Z}_2 \times \mathbb{Z}_2^\star)$, so that there are 4 irreducible representations, all of which are 1-dimensional. Each of those corresponds to a different type of excitation. Let x and y be the generators of the group. The trivial representation is the trivial excitation (or 'no excitation'). The other irreducible representations are obtained by mapping x and y to order 2 elements of $\mathbb{C}$. We obtain two bosons, when both get sent to -1 or i and one fermions when $x \mapsto -1$ and $y \mapsto i$.*
If we implement the construction on a lattice embedded on a torus, we obtain a model for a topologically protected quantum memory. Consider a (layered enough) lattice on a torus with spins on the edges. Let C_1 and C_2 be two cycles. States of the system are generated by labellings of the lattice with elements of $\mathbb{Z}_2$. As shown by Kitaev, the ground state degeneracy has dimension 4, so that we can think of the system as storing two qubits of information. If a particle-antiparticle pair is created at some site on the lattice it will re annihilate at some other site. The world-lines will form a loop on the torus and we have three possible behaviours. If the loop can be shrinked to a point, this won't affect the underlying information otherwise we obtain two non-trivial operations T_1 and T_2 affecting the ground state when the world-lines loop around cycle C_1 and C_2. If we initialise the lattice in some ground state it will remain in that state unless a T_1 or T_2 operation is implemented. If the lattice is layered enough, it is very unlikely that such processes occur spontaneously, and therefore the quantum information is protected.

1.1.3 Generalising the model

We will now try to generalize the above construction to the cases where the Hopf algebra H is not a group algebra. We will wait before 'bending cables' (i.e using the dual hopf algebra) to see how far we can go without making too many assumptions on H. This will provide an interesting illustration of the proof given in the previous chapter [ref section]. Interpreted in this physical context, the Drinfeld construction can be understood as imposing global (or topological) dependencies on the particles under consideration (e.g in the form of a Hamiltonian as the one considered by Kitaev) giving rise to anyonic behaviour.
As above, we have an oriented lattice on a $2D$ oriented manifold with particles on the edges taking values in H. For simplicity we will assume the manifold is a sphere and that the lattice has no loops. Let $\mathcal{L}_a$ be the state space of particles at some site $a = (s, p)$ (i.e the particles on edges adjacent to s or p with some order that we give below). We will define a left H-module and a left H-comodule structure on $\mathcal{L}_a$ and show those satisfy the left-left Yetter-Drinfeld module compatibility condition.
Analogously to [1] we first define an H-module structure L, for $j \in star(s) \cup bound(p)$ given by L_- from (3) if s is the origin of $j \in star(s)$, by L_+ if j is not the origin and $j \in star(s)$ and the trivial H-module otherwise. The H-comodule structure is given by the T_+ action from (5) if p is on the right of j, by T_- if it is on the left and by the trivial H-comodule otherwise. In his model, Kitaev

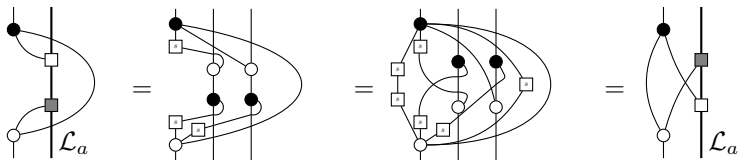
only needed to order the edges in $bound(p)$ because the comultiplication of $\mathbb{C}G$ is just the copy map, here we will need some more conventions on the ordering of the edges.
We have $bound(p) = \{j_1, j_2, ..., j_k\}$ starting and ending at vertex s, then order $star(s) = \{i_1, i_2, ..., i_n\}$ where $i_1 = j_1$ and $i_n = j_k$ also note that we have 4 possible configurations of vertex s adjacent to plaquette p and we can choose which of the edges is j_1 and which is j_k. We choose as follows:



Then we can define $\mathcal{L}_a = H_{j_1} \otimes ... \otimes H_{j_{k-1}} \otimes H_{i_2} \otimes ... \otimes H_{i_n}$ where H_m is the copy of H corresponding to edge m. Each of the H_m's carries a left H-module and left H-comodule structure as defined above so that $\mathcal{L}_a$ inherits the tensor product H-module (given by using the comultiplication of H) and tensor product H-comodule structure (using the multiplication). We obtain the following result.

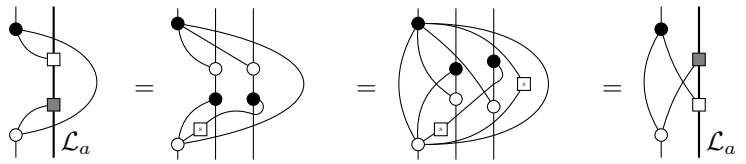
Theorem 2 *If the antipode of H is involutive (i.e $S \circ S = id_H$), then $\mathcal{L}_a$ is a left-left Yetter Drinfeld module.*

Proof We need to check the compatibility condition. First note that for all components of $\mathcal{L}_a$ except the first and last one, the H-action and H-coaction commute (as one of them is trivial). In order to keep our diagrams tidy we will only prove this for the case where $\mathcal{L}_a = H_{j_1} \otimes H_{j_k}$ (i.e $j = n = 2$), but it is easy to generalize the proof as all other components would trivially commute.
For the first configuration we have:



(15)

Where the last step also uses the fact that S is an involution. For the second configuration gives:



(16)

For a proof of the remaining two cases flip the teo proofs above and interchange white with black.

■

Note that this doesn't require the Hopf algebra to be finite dimensional. If $H = \mathbb{C}G$ then we recover the DG-module structure defined by Kitaev from the equivalence seen in the section on the Drinfeld center. [Is there a categorical description of the Hamiltonian formalism that we could use here, in order to obtain the more concrete formulation of Kitaev [1]]

1.2 Permutational Quantum Computing

Permutational quantum computing is a model of quantum computation introduced by Jordan [6]. This section was developed in collaboration with Vojtech Havlicek, (1.2.1) is based on [7] and [6]. We will first introduce the model and then give a categorical presentation not present in the literature which will allow us to generalize the model and compare it to other computational models.

1.2.1 Jordan's model

Let $\mathcal{L}$ be an n-qubit quantum system. Basis states of an n-qubit quantum systems are often specified by listing eigenvalues of Pauli-Z operators applied to each qubit, which is known as computational basis. Permutational quantum computing (PQC) works with another choice of basis states: eigenstates of complete set of commuting spin measurements on qubit subsets. Let us fix a finite set $I = \{1, 2, 3..., n\}$ indexing the qubits. With a convention that $\hbar = 1$, the spin of the k-th qubit is defined by a triple:

$$\vec{S}_k = \frac{1}{2}\left(X_k,\, Y_k,\, Z_k\right),$$

where X_k, Y_k and Z_k denote the Pauli X,Y and Z operators on the k-th qubit. The total spin operator of a qubit subset A is given by:

$$S_A^2 = \left(\sum_{k\in A} \vec{S}_k\right) \cdot \left(\sum_{k\in A} \vec{S}_k\right),$$

and we will use S^2 to denote the spin operator on the set of all qubits. Let

$$Z_A = \frac{1}{2}\sum_{k\in A} Z_k$$

denote the total Z-spin operator on qubit subset A and we label by Z the total Z-operator applied to all qubits (i.e $Z = Z_I$). Z and S^2 commute and stabilize an eigenspaces labeled by quantum numbers J and M:

$$S^2\left|J,\, M\right\rangle = J(J+1)\left|J,\, M\right\rangle, Z\left|J,\, M\right\rangle = M\left|J,\, M\right\rangle, \tag{17}$$

where J is the total spin of all qubits and M takes values $-J \leq M \leq J$ in an integer steps. There are therefore $2J+1$ Z-operator eigenstates for each J and we will refer to this degeneracy as M-degeneracy.

Now, the operators S_A^2 and S_B^2 on sets A, B commute if and only if A and B are disjoint or one is subset of the other. We can then give a complete set of commuting operators on I:

$$S^2_{\{12\}},\, S^2_{\{123\}},\, \ldots S^2,\, Z \tag{18}$$

In practice, this means that if we have n qubits, measuring each of those operators yields a a sequence of outcomes $j_{12}, j_{123}, ..., J, M$ (the eigenvalues of each operator) which tests for some state of $\mathcal{J}$. Dually, allowing superselection on the outcomes of each measurement we have also defined a preparation recipe. This choice of basis states is known as *sequential coupling.*
The j-quantum numbers on sets of qubits A, B combine according to the angular addition rules [**?**]:

$$|j_A - j_B| \leq j_{A\cup B} \leq j_A + j_B,$$
$$j_{A\cup B} + j_A + j_B \in \mathbb{Z},$$

For example if $n = 3$, there are two ways to obtain $J = \frac{1}{2}$ eigenstate of three spins - either by adding a qubit to a two-qubit singlet ($J = 0$) state, or by adding a qubit to a triplet ($J = 1$) [**?**]. We can picture those states as labeled binary trees with n leaves, which we refer to as labeled recoupling diagrams. For instance, for $n = 3$ we have:

Note that the shape of those binary trees is induced by the choices (18). Every rooted binary tree shape with n leaves (which we will refer to as recoupling diagram) yields a different choice of complete set of commuting observables, and therefore a different choice of basis for $\mathcal{L}$. And there are 2^n labelled recoupling diagrams for every recoupling diagram, one for each basis state. A computation in PQC is given by the following procedure:

Definition 3 (PQC) *Given a permutation π:*

1. *Prepare a simultaneous eigenstate $|\lambda\rangle = |j_{12}, j_{123}, \ldots, J, M\rangle$ of $S^2_{12},\, S^2_{123},\, \ldots S^2, Z$. Such basis (ie. the sequentially coupled basis) plays the role of computational basis .*

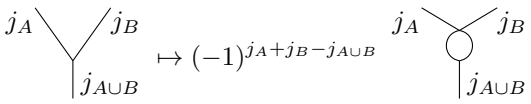
2. *Measure the following set of observables: $S^2_{\pi(1)\pi(2)},\, S^2_{\pi(1)\pi(2)\pi(3)},\, \ldots S^2, Z$. This is equivalent to applying a sequence of SWAP gates U_π in the quantum circuit model and measuring a J-spin eigenstate $|x\rangle = |j'_{12}, j'_{123}, \ldots, J', M'\rangle$ in the sequentially coupled basis.*

3. *The computing result is obtained by repeating steps 1 and 2 polynomially many times to yield an approximation of the probability distribution $P_\pi(x|\lambda) = |\langle x| U_\pi |\lambda\rangle|^2$.*

In his paper [6], Jordan shows that PQC can approximate the irreducible representations of the symmetric group in polynomial tyme. This is a relatively surprising result as this problem no classical polynomial time algorithm is known that solves the same problem. This hints that although the the PQC model

seems trivial in comparison with other quantum computation models it is still superior to classical computation. Any PQC computation (3), corresponds to a sequence of phase and racah moves.

Definition 4 (Phase and Racah moves) • *A phase move is obtained by swapping adjacent particles, diagrammatically we picture it as:*



(19)

• *Racah moves (or F-moves):*

$$\mapsto \sum_{f=|j_A-j_B|}^{j_A+j_B} F^{j_A,j_B,f}_{j_C,j_{ABC},j_{BC}} \tag{20}$$

Theorem 3 (Biedenharn-Louck [?]) *Let A, B, C be disjoint sets of qubits. Any quantum state corresponding to a labelled recoupling diagram can be transformed to a superposition of sequentially coupled labelled recoupling diagram states using a poly(n) sequence of Racah and Phase moves.*

Those moves have a general categorical description as we will see.

1.2.2 Categorical PQC

The theory of permutational quantum computing is based on the following abstract ingredients:

1. A tensor product to model many-body quantum systems
2. A direct product to model superpositions of particle types.
3. A set of labels of particle types (with antiparticle for each type) generating all other systems together with fusion rules which account for coupling of those particle types.
4. A permutational structure, i.e the possibility to permute particle positions, i.e phase moves
5. The Racah or F moves which models changes of basis.
6. Underlying Hilbert spaces which account for the quantum mechanical nature of the model.

Let us build a class of categories which account for all those ingredients. As already argued in the previous section we need the structure of a tensor category in order to model many-body quantum systems together with superpositions. We then require the category to contain a simple object for each particle type

and to be semisimple so that we obtain fusion rules (see appendix). Note that we do not require there to be finitely many simple objects as in the anyonic case. Indeed note that if we want a theory to reproduce Jordan's model for any chosen number of particles (n), the theory must contain infinitely many particle types, one for each half-integer value (value of angular momentum). We must also require the category to be rigid so that for we have antiparticles for each particle type. A tensor category is monoidal so it comes with associators which precisely model the equivalent of the Racah moves. For the permutational structure we require the theory to have a symmetric structure. And finally, if we want to recover finite dimensional Hilbert spaces underlying the objects of our theory we can impose the existence of a forgetful functor to $FHilb \simeq FVect$. Putting it all together we have obtained a rigid semisimple symmetric tensor category $\mathcal{C}$ equipped with a fiber functor $F : \mathcal{C} \to FVect$. We will call those categories Tannakian for our purposes.
The following theorem shows that any group and supergroup induces a model for permutational quantum computation.

Theorem 4 (Doplicher-Roberts) *If $\mathcal{C}$ is a rigid semisimple symmetric tensor category equipped with a fiber functor to $Vect$ then $\mathcal{C}$ is symmetrically monoidally equivalent to $Rep(G)$ for G some group (if the twist is trivial) or some supergroup (if the twist is -1).*

And in fact we recognize Jordan's model as the theory of representations of the special unitary group.

Proposition 5 *Jordan's qubit model $\mathcal{J}_2$ is the category of representations of $SU(2)$.*

Proof Irreducible representations of $SU(2)$ are precisely indexed by half-integer values and the fusion rules given by angular addition rules [**?**].

■

We can easily see that defining $\mathcal{J}_d := Rep(SU(d))$ we obtain the corresponding qu*d*it model for permutational quantum computation. The permutational structure of the categories under observation, is tightly linked to the symmetric group S_n. In his model, Jordan builds an algorithm to compute representations of S_n, this can be done in any PQC category.

Proposition 6 *Any Tannakian category $\mathcal{C}$ induces representations of the symmetric group S_n for any $n \in \mathbb{N}$.*

Proof Fix $n \in \mathbb{N}$ and a simple object $a \in obj(\mathcal{C})$ then S_n acts on $a^{\otimes n}$ by permutations, and this clearly defines a module as we can consider a as a vector space using the fiber functor.

■

Example 3 (Permutational quantum computation in $Rep(S_3)$)

Example 4 (Approximation of Dijkgraaf-Witten link invariants) *The link invariant essentially counts homomorphisms from the fundamental group of the link complement to the group G. (cite Zhenghan?)*

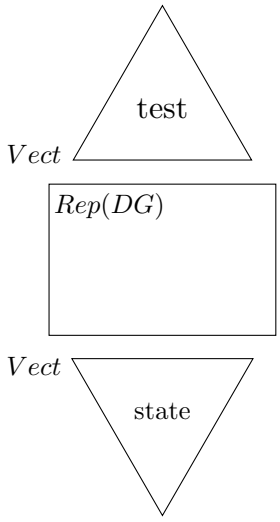
1.2.3 Can we boost PQC?

Note that if $SU(2)$ is an infinite dimensional hopf algebra with involutive antipode. In view of the generalization of Kitaev's model, it is a natural question to ask whether we can apply a similar reasoning here, to obtain a model induced by Jordan's model which exhibits topological dependencies. This means applying the Drinfeld center construction to J_2 or in other words pairing the $SU(2)$-modules of Jordan's model with an $SU(2)$ coaction to obtain a theory of $SU(2)$ Yetter-Drinfeld modules. It is not known to the reader whether such model is implementable in practice.
[If we want to repeat the same construction on the lattice as for Kitaev's model, then we are assuming that particles live in SU(2), I don't know if it makes any sense... if it does then the exact same reasoning as the section on generalising Kitaev can be applied]
[If that doesn't make sense then does it make sense to construct a lattice labelled by representations of SU(2) (say for instance they are all initialised in sector 1/2 as in the PQC framework) and define similar vertex operators (SU(2) action on the tensor) and plaquette operators which measure the the overall angular momentum of the plaquette(SU(2) coaction: fusing the particles on the plaquette). Then maybe we could show these form Yetter Drinfeld modules which give rise to a theory described by a non-symmetric braided monoidal category $Z(J_2)$ $\implies$ we have built a theory from Jordan's model which exhibits topological dependencies between particles]
[The previous reasoning gives rise to a theory with infinitely many particle types, but the assumption of TQC (and reasoning why we take modular categories as models) is that there can only be finitely many anyon types in nature, under that assumption the above theory can actually not be physically implemented, or maybe the category $Z(J_2)$ has only finitely many simple objects, which i doesn't seem to be the case...]

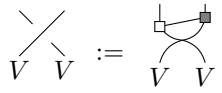
1.3 A braided representation of quantum computation

Recall our discussion on functorial semantics. We talked about categories representing syntax (such as $PROs$ and $PROPs$) and smenatic categories (such as $Vect$ or $Sets$). The category $Rep(DG)$ is a category which we filled with meaning and we have used it as a semantic category so far. Syntax and semantics are relative notions, in this section we forget all the meaning we associated to the category $Rep(DG)$ (e.g as a theory of anyons, as a model for Kitaev's lattice construction, as aboosting of PQC etc...) and we just see it as a syntax for diagrams which we will interpret in $Vect$.
We will use functorial boxes [8] to map the braided pictures in $\mathcal{C} := Rep(DG)$ down to $Vect$ and obtain a braided representation of quantum gates. We will

give a description of quantum computation in the ambient world of vector spaces, but using pictures borrowed from $\mathcal{C}$. Preparations and measurements will be performed in $Vect$ and quantum gates described by boxed braids from $\mathcal{C}$. The description looks like this:



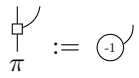
Where the center box corresponds to the forgetfull functor $U : Rep(DG) \to Vect$ applied to some diagram from $Rep(DG)$ (i.e some sequence of structural morphisms) and states and tests are prepared in $Vect$.
We know from [drinfeld center section] that $Rep(DG) \simeq \mathcal{D}_G^{lr}$, so specifying an object of $Rep(DG)$ just corresponds to choosing a vector space V with a left G-module structure and right G-comodule structure.
Let $G := \mathbb{Z}_2$, the reason why we chose this group will become apparent later on, many other choices are possible. Let us denote the standard basis of $\mathbb{C}G$ by $\{|0\rangle, |1\rangle\}$. This has natural Hopf algebra structure with multiplication given by $|i\rangle \otimes |j\rangle \mapsto |i+j\rangle$ (where $+$ is addition modulo 2) and comultiplication given by the copy map $|i\rangle \mapsto |i\rangle \otimes |i\rangle$.
We now choose a two-dimensional object of $Rep(DG)$ to serve as our qubit. Take $V = \mathbb{C}^2$ with the Z G-action (given by $1 \mapsto id$ and $-1 \mapsto Z$ the Pauli Z operator) and the X G-coaction (i.e X projective measurement). From [drinfeld section] we know $RepDG$ is braided, and the braid on $V \otimes V$ is:



(21)

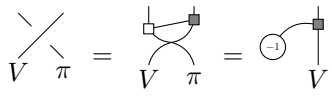
Consider the forgetfull functor $U : Rep(DG) \to Vect$. It just picks the underlying vector space of each object and the underlying linear map of each morphism. Under U, the braid is precisely a SWAP gate followed by a CNOT. We now want a way to apply phases on our system V, we will do this using ancillary systems

which will implement a phase when braiding around V. Define π as the one-dimensional representation with trivial action and the following coaction:



(22)

This is a well defined $\mathbb{C}\mathbb{Z}_2$-module structure as $-1 \in \mathbb{Z}_2$ and the comultiplication of $\mathbb{C}\mathbb{Z}_2$ is precisely the map that copies elements of $\mathbb{Z}_2$. Then braiding V with π gives the following:



(23)

Which, by definition of the action on V, is the Pauli Z operator applied to V, i.e the Z π phase. Having obtained a braided description of the CNOT and the π-phase this shows that any stabilizer quantum gate can be described via braids.

References

[1] A. Kitaev. Fault-tolerant quantum computation by anyons. *Annals Phys. 303*, pages 2–30, 2003.

[2] E. Rowell and W. Zhenghan. Mathematics of topological quantum computing. *eprint arXiv:1705.06206*, 2017.

[3] S. Simon. Toplogical quantum. http://oxfordtopquantum.tiddlyspot.com/, 2016.

[4] P. Panangaden and E. Paquette. *A categorical presentation of quantum computation with anyons*, volume 813 of *Lecture Notes in Physics*. Springer Berlin Heidelberg, 2011.

[5] M. Gould. Quantum double finite group algebras and their representations. *Bulletin of the Australian Mathematical Society*, 1993.

[6] S. P. Jordan. Permutational quantum computing. *Arxiv e-prints*, 2009.

[7] Vojtech Havlicek. Search for computational advantage in permutational quantum computing. unpublished (Term paper), April 2017.

[8] P. Mellies. Functorial boxes in string diagrams. In Springer Verlag, editor, *Computer Science Logic*, pages 1–30, 2006.