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 (particles) and processes are quantum transformations of the system. A computation consists of the preparation of some quantum states, their manipulation and measurement. 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 denote by $\mathbb{C}^2$
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 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 (it is in the complexity class $\#P$) .
The problem of building a topological quantum computer has been addressed by various authors [1] [cite freedman]. The difficulty arises in the two-dimensional nature of anyons. Indeed those are only known to arise as quasi-particle excitations on two-dimensional fluids at low temperature, an experimentally difficult setup which was never achieved for non-abelian excitations.
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

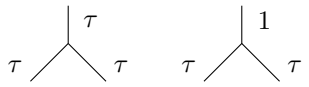
vacuum sector is a link, formed by the particle trajectories in space-time (i.e the briading process). The time evolution of the system V must be a unitary operator, so that any braiding process on n particles induces the evaluation of some representation $\beta \to U_\beta$ of the braid group B_n.
In order to make sure the braiding process is a unitary transformation of the state space we will impose one further constraint on our categorical model of computation: unitarity of the braids in the modular category in question.
Take the fusion space to be our computational hilbert space.
Topological qudits are usually encoded as fusion tree basis elements
Topologial gates: braids (can express as action of the braid group) + measurements (=fusions and associators).

1.1.2 Fibonacci anyons

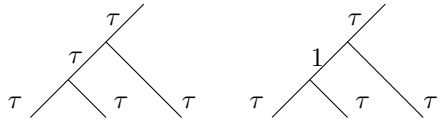
We now look back at example [ref] on Fibonacci anyons and show how to compute in the model. Recall we have only two particle types: the vacuum sector 1 and the non-trivial τ such that:

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

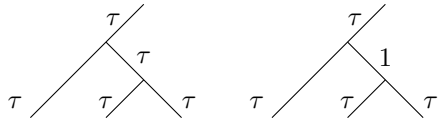
τ and 1 are their own anti-particles so we don't need to distinguish particles and antiparticles by wrinting arrows on wires. We can write the basis of the two dimensional space $\tau \otimes \tau$ as:



The fusion space $V^\tau_{\tau\otimes 3}$ is two dimensional, we will take it as our computational space and 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]:

$$\begin{bmatrix} \phi^{-1} & \phi^{-\frac{1}{2}} \\ \phi^{-\frac{1}{2}} & -\phi^{-1} \end{bmatrix}$$

where $\phi = \frac{\sqrt{5}-1}{2}$.

1.1.3 Kitaev's quantum double model

Fix a group G, and suppose we have particles living in state space $\mathbb{C}G$ (the group agebra). We will build a special case of Kitaev's quantum double model [1], in order to see directly the connection with the category $Rep(DG)$.
First of all let us consider 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:

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

Now, consider the trivial lattice on a torus with particles on the edges, i.e a square with opposite edges identified as follows:

For the moment take this lattice as our system, note that states are generated by pairs of elements of G, each labelling one edge of the lattice. Let's say that $L^g_\pm(0)$, $L^g_\pm(1)$ and $T^h_\pm(0)$ $T^h_\pm(1)$ are $L_\pm$ and $T_\pm$ operators acting on the first and second particle. Note that in this configuration, we have only one plaquette and one vertex, we will only need to define two types of operators on the lattice, indexed by elements of G. For $g, h \in G$ define:

$$\begin{aligned} A_g &= L^g_+(0)L^g_+(1)L^g_-(0)L^g_-(1) \\ P_h &= \sum_{h_1h_2h_3h_4=h} T^{h_1}_+(0)T^{h_2}_+(1)T^{h_3}_-(0)T^{h_4}_-(1) \end{aligned} \tag{2}$$

From a physical point of view P_h operators can be understood as measuring the magnetic flux of the system and A_g are 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 $g \in N(h)$.
Naturally 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:

$$A_g P_h = P_{ghg^{-1}} A_g \tag{3}$$

(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.

Definition 2 *For any finite group G, its quantum double $D(G)$ is the algebra generated by $\{P_h g\}_{h,g\in G}$ with multiplication induced by 3. $D(G) \simeq \mathbb{C}G^* \otimes \mathbb{C}G$ and inherits their hopf algebra structure (comultiplication and antipode are given by tensoring).*

$D(G)$ 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 sectors of this Hamiltonian are precisely the 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 and the allowed processes are intertwiners. We obtain the process theory $Rep(DG)$. Note that because we chose our lattice to be trivial, in the model as we described it the overall dimension of the system is at most $|G|^2$ and we cannot obtain higher dimensional representations. This issue was directly avoided by Kitaev in his presentation [1], by allowing the lattice to be arbitrairy (embedded in some manifold) and defining A_g and P_h operators on sites (i.e vertices together with a plaquette). For each site $a = (s, p)$ where s is a vertex and p a plaquette, we have operators $A_g(a)$ and $P_h(a)$ which generate the quantum double algebra. An excitation is then a state of some non-trivial irreducible representation of DG being created at some site a on the lattice. Those excitations have anyonic behaviour. In what follows we will assume the lattice was 'layered' enough so that we can create more than one excitation and we can move around without fusing them. If the lattice is big enough we have obtained a practical implementation of anyons whose behaviour is described by the modular category $Rep(DG)$. When G is abelian, 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.
In order to understand the possible anyon types in the model induced by finite group G, we must study the irreducible representations of the quantum double finite group algebra DG. This has been done by Gould [4], 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^i_α) of N_i with basis elements v^α_j, let $V_{i,\alpha} = \mathbb{C}C_i \otimes V^i_\alpha$, this has basis $\{|k, v^\alpha_j\rangle\}^{k\in C_i}_{j=1,...,dim\alpha}$ and forms an irreducible representation of $D(G)$ under the action

$$P_h g\,|k, v^\alpha_j\rangle = \delta_{h,gkg^{-1}}\,|h, \alpha(h^{-1}gk)v^\alpha_j\rangle \tag{4}$$

and the $\{V_{i,\alpha}\}$ is the complete set of irreducible representations.

Example 1 (Kitaev's Toric code) *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 $y \mapsto i$.*

1.2 Permutational Quantum Computing

This section is about a model of quantum computation introduced by Jordan [5]. We will first introduce the model as it appears in [5] 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:

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

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:

$$S^2 |J,\, M\rangle = J(J+1)\, |J,\, M\rangle\,, Z\, |J,\, M\rangle = M\, |J,\, M\rangle\,, \tag{5}$$

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, it is easy to see that the operators S^2_A and S^2_B on sets A, B commute if and only if A and B are disjoint or one is subset of the other. We can therefore give a complete set of commuting operators on I:

$$S^2_{\{12\}},\, S^2_{\{123\}},\, \ldots S^2,\, Z \tag{6}$$

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 (6). 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 clearly there are 2^n of 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 [5], 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)

Theorem 1 (Biedenharn-Louck [?]) *Let* A, B, C *be disjoint sets of qubits and use the shorthand* $AB := A \cup B$. *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 2 (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 3 *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 4 *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 2 (Permutational quantum computation in $Rep(S_3)$)

Example 3 (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?)*

1.3 A braided representation of quantum computation

For an abelian group G, we have seen that $Rep(DG)$ doesn't contain enough states to model preparations and measurements. But we will see that for suitably chosen G, $Rep(DG)$ contains a universal set of gates as braids. In this section we use functorial boxes [6] to map the braided pictures in $Rep(DG)$ down to $Hilb$ and obtain a braided representation of quantum gates. This will allow us to to describe preparations and measurements in $Hilb$ and quantum gates as boxed braids from $Rep(DG)$.
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 and X G-coaction.

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

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

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