Quantum Information Science Notes

Andy Lutomirski

Spring 2007-08

Contents

1 Alternative QC models

The previous circuit model of QC involved a sequence of unitary gates. Classical computation models include:

1. Circuits (i.e. AND, OR, NOT) – not reversible
2. Turing machines
3. Memory-based computers (close to what we actually use)
4. Wolfram's cellular automata
5. Strange things such as Conway's fraction-based machine.

These are mostly equivalent up to polynomial factors.

In QC, there are bunch of such things as well.

1. Quantum Turing machines (QTM)
2. Quantum RAM (QRAM) – iffy to implement
3. Adiabatic QC
4. Approximating Jones polynomials at 5th root of unity (i.e. $p\left(e^{\frac{2i\pi}{5}}\right)$) (BCQ-complete, but the polynomial order is huge).
5. Topological QFT (TQFT) – computing with anyons (to be discussed in the next two lectures)
6. Probably more

BQP is the class of problems solvable on a QC with bounded error in polynomial time. Scott Aaronson will teach more about that later.

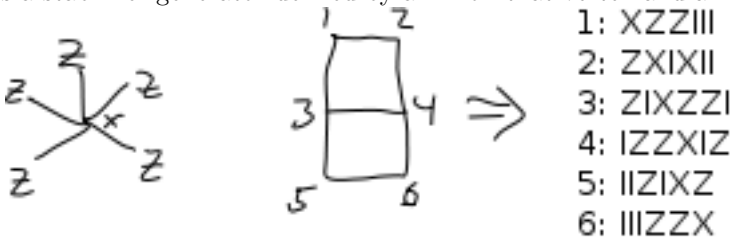
1.1 Measurement-based computation

Otherwise known as one-way QC or cluster state QC.

We start with a bunch of qubits and we measure them one at a time in bases determined by program and/or previous measurements. (Question: how powerful is the classical computer? It turns out that if you're willing to restrict yourself to Clifford states you get a Clifford QC and can do it entirely in parallel. But how much post-processing is needed and are the classical problems parallelizable?)

1.1.1 Cluster states

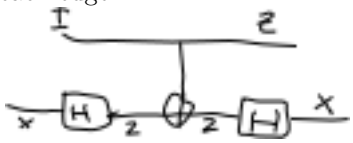
A cluster state is a stabilizer state defined on any graph G. There is qubit on each vertex and each vertex gets a stabilizer generator defined by an X on that vertex and a Z on adjacent vertices.



They commute because for two non-adjacent vertices, there can't be any overlap except for two X's in the same place, and, for two adjacent vertices, there are two XZ pairs which contribute $(-1)^2$. They are independent because each has an X in one unique spot.

To generate these states, we measure all the generators of the stabilizer group. Then apply Z to each vertex with the wrong sign. The single-vertex Z operators commute with the rest of the vertices, so all of the generator operators are forced simultaneously into the $+1$ eigenstate.

Alternatively, start in the state $|+\rangle^{\otimes|G|}$ (eigenstate of X) and apply $\begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & -1 \end{pmatrix}$ (controlled-Z) to each edge.



This Clifford group circuit works because $|0?\rangle \to |0?\rangle$ trivially, $|10\rangle \to |1\rangle(|0\rangle + |1\rangle) \to |10\rangle$, and $|11\rangle$ is the same except that the CNOT generates a sign change. Applied to a stabilizer $IIIIXI$ (with the X at a vertex), it turns the adjacent vertices into Z's. These all compute because they are all diagonal.

Yet another way is based on

$$\begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & -1 \end{pmatrix} = \exp\left[i\pi \begin{pmatrix} 0 & & & \\ & 0 & & \\ & & 0 & \\ & & & 1 \end{pmatrix} \right]$$

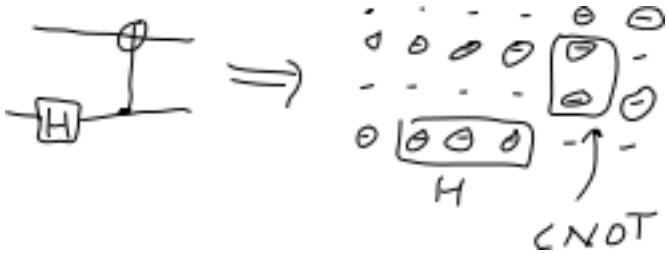
and using the appropriate Hamiltonian to apply these operators.

If we measure a cluster state vertex in the Z basis, then we anticommute with exactly one stabilizer generator (the one belonging to that vertex) and we end up with a cluster state for the graph minus that vertex.

1.1.2 Computing with cluster states

We assume that all qubits start in $|0\rangle$.

The frist step is to lay out "qubit paths" such that the paths only touch where we want operations to happen. We need three vertices in a row to do a one-qubit gate.



(We're drawing these on lattices, but that's not a requirement. It just happens to be handy because we might be able to build these on optical lattices.)

The next step is to measure all the qubits outside the logical path in the Z basis. All that's left now is a cluster state on the qubit paths, with possibly some signs wrong, so we'll need to remember that and apply virtual Pauli gates at the end of the computation to correct it.

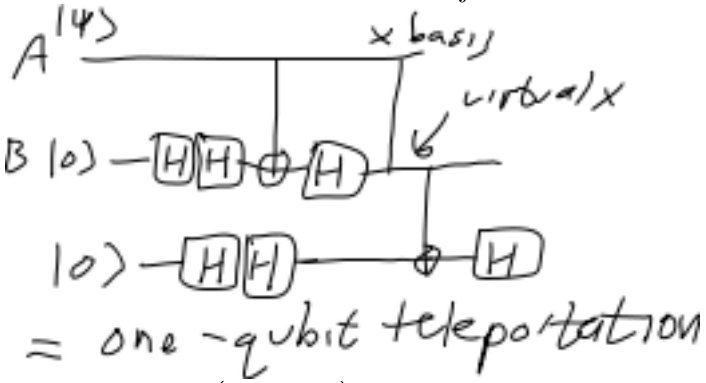
If we were to start with the left side of a wire in $|\psi\rangle$ and the rest of the wire vertices in $|+\rangle$ (this is *not* a cluster state) and apply CZ to each edge (in any order), then we end up with what we call the "logical state $|\psi\rangle$" or $|\psi_L\rangle$. (This is not how we'll actually generate $|\psi_L\rangle$.

To "move the logical state" means to move the position of the $|\psi\rangle$ (prior to the CZ's) one to the right and the remove the previous position from the cluster by measuring it. We measure the vertex with the $|\psi\rangle$ in the X basis. This only fails to commute with the single closest CZ, so we can consider only the two-qubit case. Algebraically:

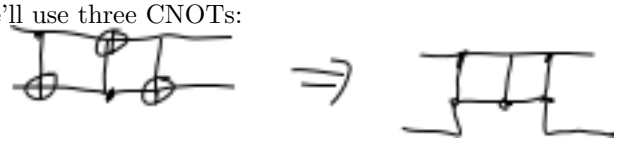
$$\begin{aligned}(\alpha|0\rangle+\beta|1\rangle)(|0\rangle+|1\rangle) &\xrightarrow[CZ]{} (\alpha|00\rangle+\beta|10\rangle+\alpha|01\rangle-\beta|11\rangle)\\ &\xrightarrow{\text{measure } x_1} \begin{cases}(\alpha+\beta)|0\rangle+(\alpha-\beta)|1\rangle, & \text{if } 0\\ (\alpha-\beta)|0\rangle+(\alpha+\beta)|1\rangle, & \text{if } 1\end{cases}\end{aligned}$$

Now, if you get a + outcome, then you've applied H. If you get a - outcome, then you've applied XH. We can't undo the X, but we can keep track of it and, in the future, apply GX instead of G. We end up accumulating a Pauli error, but we can keep track of it.

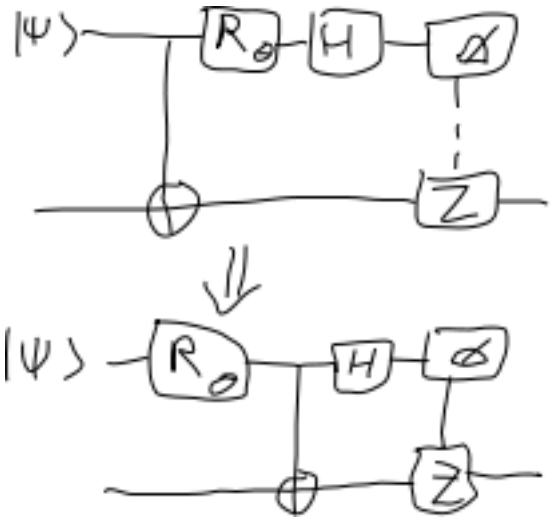
Another way to see this is to imagine we've started with all bits in $|0\rangle$ and that we've taken advantage of the fact that CZ commutes with non-adjacent measurements to reorder everything:



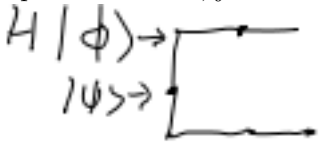
The gates $R_\theta = \begin{pmatrix} 1 & \\ & e^{-i\theta} \end{pmatrix}$, CNOT, and $HR_\theta H$ are universal for QC. If we imagine the grid to be checkerboard-colored, we'll put R_θ on white squares and $HR_{theta}H$ on black squares. To get wires to cross, we'll use three CNOTs:



To get R_θ gates, we'll measure in the $|0\rangle \pm e^{i\theta}|1\rangle$ basis, which is equivalent to applying R_θ and measuring in the x basis (with a possible sign error).



To implement CNOT, just let two qubits intersect. The whole process is automatic.



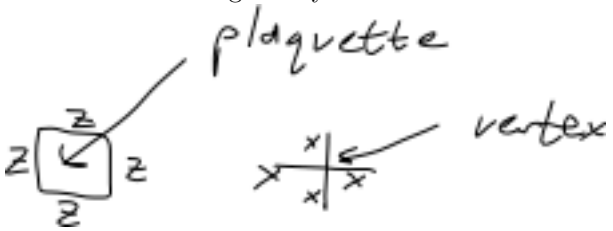
To commute all the errors to the end, we may need change signs on the thetas as we go through, because $R_\theta Z = Z R_\theta$ and $R_\theta X = X R_{-\theta}$. (Open problem: if we only have $\theta = \frac{\pi}{2}$, can we measure everything at once and then just do classical computation. Can we parallelize the classical part as well? Can we do anything that's not in NC1 (???) this way.)

1.2 Anyons and toric codes

1.2.1 Toric codes (and their extensions to other graphs)

Toric codes are also referred to as Abelian anyons. They live on grids where the qubits are edges. The squares in the grid are known as plaquettes, presumably because John Preskill used to work on lattice gauge theory. The lattice points are called vertices.

The stabilizers are given by:

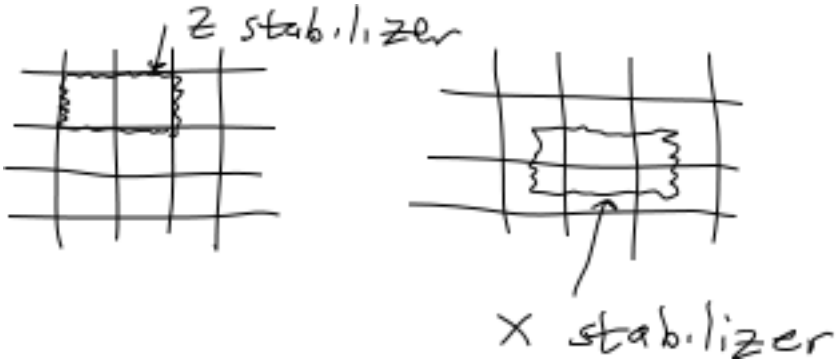


On a $k \times l$ grid, there $(k+1)\,l + (l+1)\,k = 2kl + k + l$ qubits, with $(k+1)\,(l+1) - 1$ independent generators (the product of all the vertex generators is $I^{\otimes ?}$), so we have dimension 0.

On a $k \times l$ torus, there are extra edges wrapping around each side. Now there are $(k+1)\,(l+1)$ edges in each direction, for a total of $2\,(k+1)\,(l+1)$ qubits. There are $(k+1)\,(l+1) - 1$ independent plaquettes

and the same number of vertex generators, for a total of $2(k+1)(l+1)-2$ generators and two qubits left.

The normalizers in the Pauli group The stabilizer contains all unions of simple closed curves of Z's on the torus, by multiplying all the plaquettes contained inside. For the X elements of the stabilizer, look at the dual lattice (turning vertices into faces and vice versa – in this case, the same lattice offset by $(\frac{1}{2}, \frac{1}{2})$).



The normalizer contains sets of X's that have 0, 2, or 4 edges incoming on any plaquette. Euler's theorem says that these decompose into closed curves, so the X normalizer elements outside the stabilizer are closed curves of odd winding number in either direction. The Z normalizer elements are the same things on vertices. In other words, the normalizer is the same thing as the stabilizer with the requirement that the curves have 0 winding number removed. This gives four extra generators ($\bar{X}_1$ is X winding horizontally acting on vertical qubits, $\bar{X}_2$ is X winding vertically acting on horizontal qubits, $\bar{Z}_1$ is Z winding vertically acting on vertical qubits, and $\bar{Z}_2$ is Z winding horizontally acting on horizontal qubits). It's easy to check the (anti-)commutation relations.

To correct errors, we measure the syndromes at each vertex and each plaquette. If we get -1 for a pair of "stars", then the error is some path between them. It doesn't matter which path we use to correct it so long as they are topologically equivalent. Pick the shortest. If we have 4 -1's, then we want the shortest matching (this is min-cost matching, which can be solved in polynomial time, although the proof is a little bit nontrivial).

We know that this is a CSS code b/c the generators are all X's or all Z's. If we defined it on an arbitrary graph on a manifold (i.e. a graph plus a definition of what a face is), then #generators = #faces $-1+$ #vertices -1. The #encoded qubits $= E - F - V + 2 = 2 \cdot$ Genus. The genus is the number of holes, so it is 1 for a torus. (Another way to see it is that there are 2^{genus} ways to thread through the holes, and there are both X lines and Z lines.

1.2.2 The Hamiltonian version and Abelian anyons

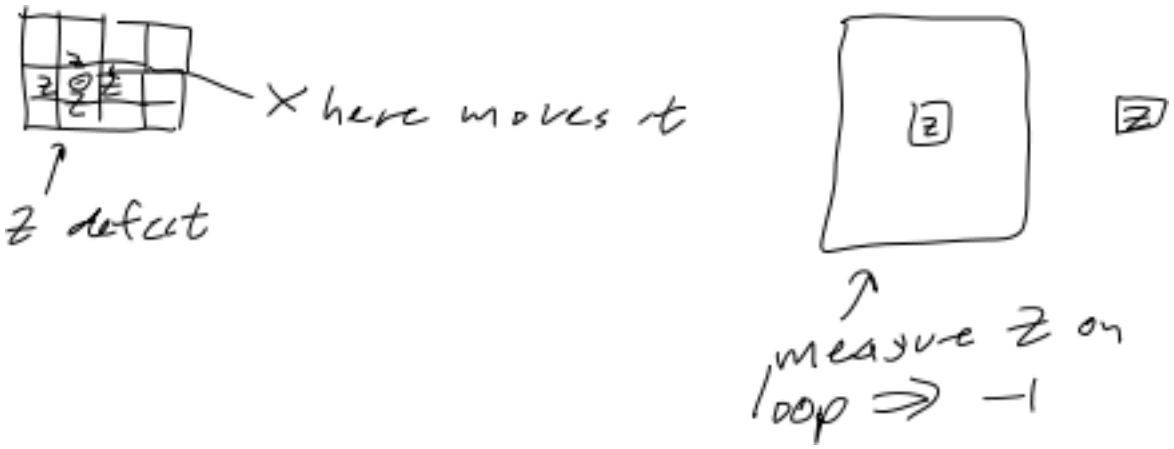
Start with the Hamiltonian

$$H = \sum_i \left(\frac{I - g_i}{2} \right).$$

The ground state is then the state where all of the stabilizers are in the +1 eigenstate. Then the ground state is the code space.

There can't be just one -1 eigenvalue in a different state because multiplying all the other stabilizers recovers that one. But there can be 2. As before, though, if we have two -1 eigenstates (say on the X stabilizer elements, i.e. vertices), then the error could be corrected by applying Z over the shortest path.

The Z operator moves one of these "X_{-1} particles, " and, if two collide, they annihilate. Similarly, X moves a "Z_{-1} particle."

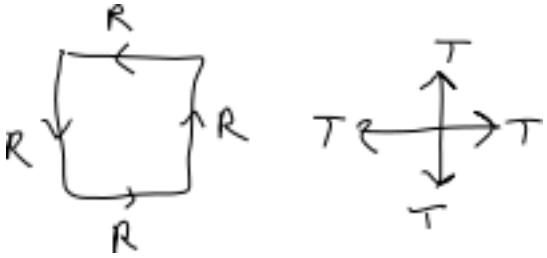


If we move a Z defect around the X defect, then we'll cross the "string of $X's$" coming out of the Z defect once, and we'll pick up a phase of -1. This is a representation of $\mathbb{Z}_2$.

Qutrits and $\mathbb{Z}_3$ anyons Introduce operators

$$\begin{aligned} T &= \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \\ R &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \omega & 0 \\ 0 & 0 & \omega^2 \end{pmatrix} \\ \omega &= e^{2\pi i/3} \end{aligned}$$

Now we have operators $\overset{T}{\rightarrow}=\overset{T^2}{\leftarrow}$ and $\overset{R}{\rightarrow}=\overset{R^2}{\leftarrow}$. Flipping a qubit is just a basis change $|1\rangle \leftrightarrow |2\rangle$. The stabilizer generators are



To check that the generators commute, there's only one case (say the T is at the top-right corner of the R):

$$R_1 R_2 T_1 T_2^2 = \omega^3 T_1 T_2^2 R_1 R_2.$$

Moving a T defect around an R defect gives a phase of ω or ω^2 depending on direction.

1.2.3 The group S_3

S_3 is the group of permutations of three elements a, b, and c. The group elements can be written with a cycle representation, where, for example, (12) means interchanging the first two elements. So (23)(12) = (132) (you can draw a picture, but it's not very important for what we're currently doing).

1.2.4 Non-ableian groups

Label the basis states of our Hilbert space by group elements (e.g. $|g_1\rangle, \cdots, |g_m\rangle$). (So, if we used S_3, we'd have 6-ary qudits, with basis states $|(12)\rangle$, $|(13)\rangle$, $|(23)\rangle$, $|(123)\rangle$, $|(132)\rangle$, $|I\rangle$.)

We'll define directions on edges, where $\overset{|z\rangle}{\rightarrow}=\overset{|z^{-1}\rangle}{\leftarrow}$. For each group element g, we have two unitary operators $L_+^g|z\rangle = |gz\rangle$ and $L_-^g|z\rangle = |zg^{-1}\rangle$. This way, L_- on a left-facing qudit looks like L_+ on a right-facing qudit.

We have nonunitary operators $T_+^h|z\rangle = \delta_{h,z}|z\rangle$ and $T_-^h|z\rangle = \delta_{h,z^{-1}}|z\rangle$. Any operator at all can be constructed from these operators and scalars. The commutation relations are

$$\begin{aligned} L_+^g T_+^h &= T_+^{gh} L_+^g \\ L_-^g T_+^h &= T_+^{hg^{-1}} L_-^g. \end{aligned}$$

The star relation gives $A_g(s) = \prod_{j \in s} L_{j,s}^g(j)$, where we use + if j points into s and - otherwise. The plaquette relations gives $B_h(p,s) = \sum_{h_1 \cdots h_4 = h} \prod_{i<4} T^{h_i}(j_i)$, where s is the starting point. This means that $\sum_{h \in G} B_h(p,s) = I$ because you select every possibility.

The stabilizer generators are $A(s) = \frac{1}{N} \sum_{g \in G} S_g(s)$ and $B(p) = B_I(s,p)$ (the starting vertex doesn't matter because if $h_1 h_2 h_3 h_4 = I$ then $h_2 h_3 h_4 h_1 = I$). These are both projectors and we want the null space. (To prove that they A is a projector, we note that this sort of symmetrization over group elements has no effect if it's already been done, so $A^2 = A$. B is straightforward.)

Considering a star operator on a vertex of a plaquette, there are two edges of overlap. That piece of the B operator looks like $\sum_h \sum_{h_1 h_2 = h} T^{h_1}(j_1) T^{h_2}(j_2)$. That piece of the A operator looks like

$$\begin{aligned} A &= \frac{1}{N} \sum_g L^g(j_1) L^g(j_2) \\ &= \sum_{h = h_1 h_2} \sum_g L_+^g(j_1) L_-^g(j_2) T_+^{h_1}(j_1) T_+^{h_2}(j_2) \\ &= \sum_{h_1 h_2 = h} \sum_g T_+^{g h_1}(j_1) T_+^{h_2 g^{-1}}(j_2) L_+^g(j_1) L_-^g(j_2) \end{aligned}$$

We'd like to show that these commute independently of h. Something is backwards in this proof.

On a torus we get a degenerate ground state space.