Quantum Coding Theory **(UC Berkeley CS294, Spring 2024)**

Lecture 23: The Clifford group

April 17, 2024

Lecturer: John Wright *Scribe: Thilo Scharnhorst*

Last time we finished our series on increasingly good, but also increasingly difficult to understand, quantum error correction codes. And then we saw almost the full proof that quantum Tanner codes give good qLDPC codes.
Today we are going to move on to another main topic of the course, which is Fault Tolerance. So now the setting is that we have all these good quantum codes and we ask ourselves how we use them for computation so that it is resilient to noise.
This series of lectures follows the video lectures by Daniel Gottesman online [DG18]. His lecture notes are also available online [Got09].

We start by looking at a set of unitary gates that is very important for fault tolerance and is know as the Clifford group.

1 The Clifford Group

Recall:

- $\text{Pauli}_n = \{\pm P_1 \otimes \ldots \otimes P_n, \pm iP_1 \otimes \ldots \otimes P_n, P_i \in \{I, X, Y, Z\}\}$
- $S = \langle g_1, \ldots, g_{n-k} \rangle$ is a stabilizer group under the conditions that the g_i's are independent and commute and $-I \notin S$. Then S gives a stabilizer code of dimension k.
- A stabilizer state is the unique state (up to phase) stabilized by

$$S = \langle g_1, \cdots, g_n \rangle \text{ i.e. } g_i|\chi\rangle = |\chi\rangle \quad \forall i$$

  E.g. $|00\ldots0\rangle$ has the stabilizers $S = \{Z_1, \ldots, Z_n\}$ (Consisting of all Pauli Z's).
  The state is also the unique state in the stabilizer space (up to phase, for instance -$|00\ldots0\rangle$ is also stabilized by these). If we have n Pauli gates and they all commute and are independent then there is always one state that is specified by them.

The main topic of today's lecture is going to be the Clifford group.

Definition 1.1 (The Clifford Group)**.**
The Clifford group Cliff_n is

$$\text{Cliff}_n = \{U : UPU^\dagger \in \text{Pauli}_n, \forall P \in \text{Pauli}_n\}$$

The following gates are all in the Clifford group:

- $\text{Pauli}_n \subseteq \text{Cliff}_n$:
  $P \in \text{Pauli}_n : PQP^\dagger = PQP = \pm PPQ = \pm Q$
- $e^{i\theta} \cdot I \in \text{Cliff}_n$
- Hadamard H:
  $HXH = Z$, $HZH = X$, $HYH = HiXZH = i(HXH)(HZH) = iZX = -Y$
  (it would actually suffice to show this just for X and Z, as Y is a product of both)
- Phase gate $S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$:
  $SZS^\dagger = SS^\dagger Z = Z$
  $SXS^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ i & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = Y$
  $SYS^\dagger = -X$ (we could also deduce it from X and Z again)

- CNOT gate:
  It suffices to look at the following 4 Pauli strings as any Pauli string can be written as a product of them:

  1. CNOT $(X \otimes I)$CNOT $= X \otimes X$
     (we see this via seeing how it acts on a state $|a, b\rangle$:
     $|a, b\rangle \to \text{CNOT}\,|a, a + b\rangle \to X \otimes I\,|a + 1, a + b\rangle \to \text{CNOT}\,|a + 1, b + 1\rangle$)
  2. CNOT $(I \otimes X)$CNOT $= I \otimes X$
  3. CNOT $(Z \otimes I)$CNOT $= Z \otimes I$
  4. CNOT $(I \otimes Z)$CNOT $= Z \otimes Z$
     (we again see this via seeing how it acts on a state $|a, b\rangle$:
     $|a, b\rangle \to \text{CNOT}\,|a, a + b\rangle \to I \otimes Z(-1)^{a+b}\,|a, a + b\rangle \to \text{CNOT}(-1)^{a+b}\,|a, b\rangle = Z \otimes Z\,|a, b\rangle$)

The Clifford group is actually the group that is generated by this set of one- and two-qubit gates. If I just have access to these one- and two-qubit gates then I can generate any gate in the Clifford group. We will see in the next lecture why we care about the Clifford group. In short, when I want to make a computation fault tolerant, what I want to do is take my qubits and encode them into an error correcting code. But I don't want them to just be resistant to noise, I also want to be able to perform computations on my encoded qubits. Because you cannot simulate a circuit without being able to compute on the qubits of the circuit. So you might ask what sort of gates am I allowed to apply to my encoded qubits. It turns out that for many error correcting codes we will look at, the natural gates that you are able to apply correspond to Clifford operations. We are going to see this more next lecture, but if I have an encoded state and I can apply Pauli gates, Hadamard gates, Phase gates

and CNOT gates, the question is how powerful this is. Can I use these gates to compute any gates that I want, or is this a restricted gate set. And if this is a restricted gate set, how do I get access to more general gates?
Let's prove a little fact about the Clifford group (i.e. that it is indeed a group).

Fact 1.2. *Cliff_n is a group.*

Proof.

- $I \in \text{Cliff}_n$
- $U, V \in \text{Cliff}_n \Rightarrow UV \in \text{Cliff}_n$
  as $UVP(UV)^\dagger = U(VPV^\dagger)U^\dagger = UQU^\dagger = R \in \text{Pauli}_n$
- $U \in \text{Cliff}_n \Rightarrow U^{-1} \in \text{Cliff}_n$
  as $UPU^\dagger = Q \Rightarrow P = U^\dagger QU = U^{-1}Q(U^{-1})^\dagger$
  (So you just have to check that for every Pauli matrix P there is another Pauli matrix Q s.t. this is fulfilled. When P ranges over all the Pauli matrices, Q also ranges over all the Pauli matrices, so conjugating by U is just a permutation of Pauli matrices.)

□

For $U \in \text{Cliff}_n$:
$h(P) = UPU^\dagger$
$h(I) = I$ (as well as $h(-I) = -I$)
$h(P)h(Q) = UPU^\dagger UQU^\dagger = UPQU^\dagger = h(PQ)$
So h is a group homomorphism (i.e. respects the group's structure).
$= U(\pm QP)U^\dagger = \pm UQU^\dagger UPU^\dagger = \pm h(Q)h(P)$
where we get the same sign as $PQ = \pm QP$, so not only does h keep the multiplication structure, but also the commutation structure)

Let's mention one more fact that follows:

Fact 1.3. *It suffices to specify h on the generators $(X_1, \ldots, X_n, Z_1, \ldots, Z_n)$ of Pauli_n.*

So we only need to specify h on these Pauli matrices to specify U. Now we saw some properties that h has to satisfy, that essentially are some properties that our matrix U has to satisfy to be in the Clifford group. So a Clifford group element is going to permute the Pauli matrices correspondingly. Now the question is whether there are any further restrictions on what our Clifford group elements are allowed to do.

Theorem 1.4. *Let $h(\cdot)$ be a permutation of Pauli_n that preserves the group multiplication and commutation relations (i.e. h is a group homomorphism).*
Then: $\exists$ n-qubit unitary $U \in \text{Cliff}_n$ s.t. $h(P) = UPU^\dagger$ and U is unique up to global phase. (If h satisfies the given condition it corresponds to one exact U in Cliff_n)

Proof. To find the unitary U corresponding to a map h, it suffices to specify the map for the Z_i matrices and also correctly computes the map for the X_i matrices, as follows:

$$\begin{aligned} P \to h(P): \quad & Z_i \to h(Z_i) \\ & X_i \to h(X_i) \end{aligned}$$

h is now specified on what it does on the Z_i and X_i. What the problem then reduces to is to find a unitary U that maps the $+1$ eigenbasis of Z_i to the $+1$ eigenbasis of $h(Z_i)$. This is because conjugation with U is never going to change the eigenvalues but just the eigenbasis. Now the all zeros vector is in the $+1$-eigenbasis of each Z_i

$$|00\ldots0\rangle = \text{ stabilized by } \langle Z_1, \ldots, Z_n \rangle$$

Then:

$$\begin{aligned} & U|00\ldots0\rangle \\ & = UZ_i|00\ldots0\rangle \\ & = \underbrace{UZ_iU^\dagger}_{h(Z_i)} U|00\ldots0\rangle \\ & = h(Z_i)U|00\ldots0\rangle = \text{ stabilized by } \langle h(Z_1), \ldots, h(Z_n) \rangle \end{aligned}$$

Now picking a phase specifies a unique vector $U|00\ldots0\rangle$ stabilized by $\langle h(Z_1), \ldots, h(Z_n) \rangle$

$$\begin{aligned} & U|10\ldots0\rangle \\ & = UX_1|00\ldots0\rangle \\ & = UX_i|00\ldots0\rangle \\ & = \underbrace{UX_1U^\dagger}_{h(X_1)} U|00\ldots0\rangle \\ & = h(X_1)U|00\ldots0\rangle \end{aligned}$$

And from this it follows:

$$U|b_1, \ldots, b_n\rangle = h(X_1^{b_1}, \ldots X_n^{b_n})U|00\ldots0\rangle$$

Now the only thing left to show is that for a specified choice (of phase) of $U|00\ldots0\rangle$, all the $U|b_1, \ldots, b_n\rangle$ for different b_i will be orthogonal and thus U unitary. Let's prove this, by showing $\langle 00\ldots0|\, U^\dagger U|10\ldots0\rangle = 0$ (all the other inner products work analogously):

$$\begin{aligned}
\langle 00\ldots0|\, U^\dagger U\, |10\ldots0\rangle &= \langle 00\ldots0|\, U^\dagger h(X_1) U\, |00\ldots0\rangle \\
&= \langle 00\ldots0|\, U^\dagger \underbrace{h(X_1)h(Z_1)}_{=-h(Z_1)h(X_1)} U\, |00\ldots0\rangle \\
&= -\langle 00\ldots0|\, U^\dagger h(Z_1)h(X_1) U\, |00\ldots0\rangle \\
&= -\langle 00\ldots0|\, U^\dagger \underbrace{h(X_1) U\, |00\ldots0\rangle}_{=U|10\ldots0\rangle} \\
&= -\langle 00\ldots0|\, U^\dagger U\, |10\ldots0\rangle \\
&= 0
\end{aligned}$$

This works for any b_i, as we can then just choose one Z_i, this shows that U is unitary. The next thing we have to show is that this unitary actually fulfills $h(Z_i) = UZ_iU^\dagger$ and is thus the right unitary!

$$\begin{aligned}
UZ_iU^\dagger U\, |b_1,\ldots,b_n\rangle &= UZ_i\, |b_1,\ldots,b_n\rangle \\
&= (-1)^{b_i} U\, |b_1,\ldots,b_n\rangle
\end{aligned}$$

Now let's compare this with what we get replacing $UZ_iU^\dagger$ by $h(Z_i)$ and we just need to check that we get the exact same thing:

$$\begin{aligned}
h(Z_i)U\, |b_1,\ldots,b_n\rangle &= h(Z_i)U\, |b_1,\ldots,b_n\rangle \\
&= h(Z_i)h(X_1^{b_1},\ldots,X_n^{b_n})U\, |00\ldots0\rangle \\
&= (-1)^{b_i} h(X_1^{b_1},\ldots,X_n^{b_n})\, \underbrace{h(Z_i)U\, |00\ldots0\rangle}_{U|00\ldots0\rangle} \\
&= (-1)^{b_i} U\, |b_1,\ldots,b_n\rangle
\end{aligned}$$

So we get the same thing for any b_i, which thus shows that indeed $h(Z_i) = UZ_iU^\dagger$, finalizing the proof. □

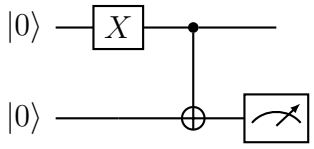
Now what is an upper bound on the size of the Clifford group? As seen above one can see a Clifford group element as a permutation fulfilling the specific commutation relations. This permutation is very restricted and is already specified by how it acts on the generators of the Pauli gates. There are $2n$ generators which can all get mapped to different elements in Pauli_n. And as they can only be mapped to the Pauli gates with ± 1 phases (not $\pm i$) as the eigenvalues have to stay the same, there are $2 \cdot 4^n$ Pauli gates which each of the generators can be mapped to. Thus we get:

Fact 1.5.

$$
\begin{aligned}
|Cliff_n| &\leq |Pauli_n|^{2n} && \textit{(only considering the Pauli gates with real eigenvalues)} \\
&= |2 \cdot 4^n|^{2n} \\
&= 2^{(2n+1)\cdot 2n} && \textit{(as there are } 2n \textit{ generators each of which has to map to different } P \in Pauli_n \\
& && \textit{where here we only care about the Pauli strings of eigenvalues real 1)}
\end{aligned}
$$

So, it takes $2n(2n+1)$ bits to specify a Clifford, while a general unitary takes $\geq 4^n$ bits to specify. That is a big difference. So Clifford gates are very easily specified and you can just write down any Clifford you want. Can they have the full power of Quantum Computation if there are so few of them?! If all I'm allowed is to implement Clifford gates it seems like we can't use that to simulate a general quantum computation.
We will formalise this intuition in the Gottesman-Knill Theorem, which says: If we only use Clifford gates we can simulate the circuit classically (starting with any stabilizer state, like a bit string, but we can also just think of $|00\ldots0\rangle$ for simplicity). Let me give an example of how we are going to do this, which will contain most of the ingredients of the actual full proof:

Example 1.6. *We would like to simulate the action of the following simple circuit, in a way that generalizes to more complicated Clifford circuits. We are going to do this by keeping track of the state at each step. As each of those are stabilizer states, they are described by their n Pauli stabilizers and can thus be specified efficiently.*



Let us specify the states at each step by calculating its stabilizers:

- *step 0:* $Z \otimes I$, $I \otimes Z$
- *step 1:* $XZX \otimes I = -Z \otimes I$, $XIX \otimes Z = I \otimes Z$
- *step 2:* $CNOT(-Z \otimes I)CNOT = -Z \otimes I$, $CNOT(I \otimes Z)CNOT = Z \otimes Z$

So, the final state $|\psi\rangle$ *is stabilized by* $-Z \otimes I$ *and* $Z \otimes Z$*, and thus also by their product* $-I \otimes Z$*. Therefore* $-I \otimes Z\,|\psi\rangle = |\psi\rangle$*, or equivalently* $|\psi\rangle$ *is in the* $+1$ *eigenspace of* $-I \otimes Z$*, which is given by* $-I \otimes |1\rangle\langle 1|$*. Thus we know we will always see outcome* 1*, which is exactly what we would expect.*

Theorem 1.7 (Gottesman-Knill Theorem)**.** *Consider a quantum circuit consisting of Clifford gates and Pauli measurements. It is possible to efficiently simulate its behaviour (output distribution) given the stabilizer state as input. (Even with gates conditioned on measurements)*

We will see the proof next time!

References

[DG18] Beni Yoshida Daniel Gottesman. Iqc - quantum error correction (lecture videos). https://pirsa.org/c17045, 2018. Accessed 05/06/24. (document)

[Got09] Daniel Gottesman. An introduction to quantum error correction and fault-tolerant quantum computation, 2009. (document)