Quantum Information and Quantum Computing, Solutions 12

Assistants : sara.alvesdossantos@epfl.ch, david.linteau@epfl.ch, shao.chiew@epfl.ch

Problem 1 : Correctable and non correctable errors in Shor's 9-qubit code

1. We know that the Shor's code encodes a generic single-qubit state to three sets of three qubits in order to detect and correct single-qubit errors. This is done by encoding the two basis state

$$|0_L\rangle \longrightarrow \frac{1}{\sqrt{8}}(|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \tag{1}$$

$$|1_L\rangle \longrightarrow \frac{1}{\sqrt{8}}(|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \tag{2}$$

where $|i_L\rangle$ means that we encoded the logical $|i\rangle$ state of a single qubit. This code can correct one bit flip error in each set of three qubits, plus one phase error. Also, two phase errors in the same set of three qubits act the same on the codewords, so do nothing to the state (the product is in the stabilizer). For this reason, the code can correct X_2X_7 , X_5Z_6 and Z_5Z_6. The two-qubits error X_1X_3 cannot be corrected because it involves two bit flip errors in the same set of three, and Y_2Z_8 cannot be corrected because it involves two phase flip error on different set of three (the bit flip part of Y_2 can be corrected, however).

2. For X_1X_3 , the error correction procedure notes that qubit number 2 is the misfit, and "corrects" it by performing the bit flip operation X_2 . Thus, the net effect is to flip all of the first three qubits. Thus, the encoded $|0_L\rangle$ state does not change (as $|000\rangle + |111\rangle$ becomes $|111\rangle + |000\rangle$), but the encoded $|1_L\rangle$ state becomes $-|1_L\rangle$ (as $|000\rangle - |111\rangle$ becomes $|111\rangle - |000\rangle$). That is, $\alpha|0_L\rangle + \beta|1_L\rangle$ becomes $\alpha|0_L\rangle - \beta|1_L\rangle$; the logical operation is an encoded Z.

   For Y_2Z_8 , we can write $Y_2 = iX_2Z_2$. The code can correct X_2 , but the residual phase error Z_2Z_8 cannot be corrected. (The factor i is an overall phase, which has no physical significance and does not count as an error.) The correction procedure notes that the phase on the middle block of three is different, and tries to fix it with a Z_5 , say, making the overall error a $Z_2Z_5Z_8$. Thus, $|000\rangle + |111\rangle$ becomes $|000\rangle - |111\rangle$ on all three blocks and vice-versa, changing $|0_L\rangle$ into $|1_L\rangle$. This is a logical X operation: $\alpha|0_L\rangle + \beta|1_L\rangle$ becomes $\alpha|1_L\rangle + \beta|0_L\rangle$.

Problem 2 : Combining stabilizer codes

1. Clearly the $M_i \otimes I_{n_2}$ commute with each other, since the M_i 's do, and the $I_{n_1} \otimes N_j$ commute with each other similarly. Moreover, $M_i \otimes I_{n_2}$ also commutes with $I_{n_1} \otimes N_j$, since they are trivial (identity) where the others are not, so we have an Abelian group and a stabilizer code $S = S_1 \oplus S_2$ using $n = n_1 + n_2$ qubits.

2. There are $n_1 - k_1$ M generators and $n_2 - k_2$ N generators, for a total of $r = n_1 + n_2 - (k_1 + k_2)$, so the number of encoded qubits is $k = n - r = k_1 + k_2$.

3. Suppose E is an error which is not detected by the code S_1 . That is, E commutes with all generators of S_1 , but is not in S_1 . Then, clearly, $E \otimes I_{n_2}$ commutes with the generators of S, but is not in S, and is therefore not detected by S either. Conversely, if E is detected by S_1 , either $E \in S_1$, in which case $E \otimes I_{n_2} \in S$, or E anticommutes with some generator M_i of S_1 , in which case $E \otimes I_{n_2}$ also anticommutes with the generator $M_i \otimes I_{n_2}$ of S. Similarly, if F is or is not detected by S_2 , then $I_{n_1} \otimes F$ is or is not detected by S as well.

   The distance of S_1 is d_1 and the distance of S_2 is d_2 . That means that there exist errors E and F of weight d_1 and d_2 , respectively, which are not detected by S_1 and S_2 . Thus, there is a Pauli operator of weight $d = \min(d_1, d_2)$ which is not detected by S.

   Now, we have to prove that this is exactly the distance we are looking for. To do this, we have to demonstrate that S can detect any error of weight less than d. Any operator of weight less than d can be written as $E \otimes F$, where E has weight less than d_1 (and is therefore detected by S_1) and F has weight less than d_2 (and is therefore detected S_2).

   We have four cases:

   - E and F anticommute with some generator of S_1 and S_2 , respectively. In this case, clearly the product is detected by both the M generators and the N generators, and is therefore detected by S.
   - $E \in S_1$ and F anticommutes with a generator of S_2 . In this case, $E \otimes F$ is detected by an N generator, so is still detected by S
   - Same situation if conversely $F \in S_2$ and E anticommutes with a generator of S_1, so still detected by S
   - $E \in S_1$ and $F \in S_2$. In this case, $E \otimes I_{n_2}$ and $I_{n_1} \otimes F$ are both in S, which is closed under multiplication, so $E \otimes F \in S$ as well.

   In all 4 cases, S detects the error $E \otimes F$. That is, S detects all errors of weight less than d and fails to correct at least one error of weight d. The distance of the code is thus exactly $d = \min(d_1, d_2)$.

Problem 3 : Logical codewords of the 5-qubit stabilizer code

1. We are considering tensor products of Pauli operators. This means that, if we want to see if an operator commutes with another, we have to look at pairwise commutation relationships. In this case we have that Z commutes with I and Z, but $XZ = -ZX$, therefore for a 5 qubit operator to commute with Z_L it has to contain an even number of X operators. This is the case of the four five-qubit operators indicated in the text, so they all commute with Z_L. Given that Z_L does affect the logical qubits and commutes with the generators, we can use it as a logical operator, given that it will not bring us out of the code space.

2. Since Z_L represents the logical Z gate applied on the logical qubit encoded by the code, we want that $Z_L|0_L\rangle = |0_L\rangle$ and $Z_L|1_L\rangle = -|1_L\rangle$, namely each term in the definition of $|0_L\rangle$ has to contain an even number of $|1\rangle$ and each term in the definition of $|1_L\rangle$ has to contain an odd number of them. We can also use the Z_L operator, together with the four generators of the whole code, in order to create a projector that will help us to detect the $|0_L\rangle$ and $|1_L\rangle$ state. We therefore define the operator

$$P = \sum M \tag{3}$$

where the sum runs over the four generators of the stabilizer plus the logical Z_L. It is intuitively clear that this operator is a projector on the state $|0_L\rangle$, as per definition there are no operators M other than Z_L and the four generators, such that $M|0_L\rangle = |0_L\rangle$. We can then determine the $|0_L\rangle$ state by finding a state which is not annihilated by the projector (such as $|00000\rangle$) and just see what we get when the projector acts on it:

$$\begin{aligned}|0_L\rangle = \frac{1}{4}(&|00000\rangle + |10010\rangle + |01001\rangle - |11011\rangle + |10100\rangle - |00110\rangle - |11101\rangle - |01111\rangle \\ &+ |01010\rangle - |11000\rangle - |00011\rangle - |10001\rangle - |11110\rangle - |01100\rangle - |10111\rangle + |00101\rangle)\end{aligned}$$

We can then apply a similar procedure to the state $|11111\rangle$, but in this case the projector P must include the operator $-Z_L$ instead of Z_L. We get

$$\begin{aligned}|1_L\rangle = \frac{1}{4}(&|11111\rangle + |01101\rangle + |10110\rangle - |00100\rangle + |01011\rangle - |11001\rangle - |00010\rangle - |10000\rangle \\ &+|10101\rangle - |00111\rangle - |11100\rangle - |01110\rangle - |00001\rangle - |10011\rangle - |01000\rangle + |11010\rangle)\,.\end{aligned}$$