QDCS : Diagrammatic Calculus and Error Correction

Renaud Vilmart

TD 6

1 Stabilisers and Normalisers

Question 1. Show that any subgroup of $\mathcal{P}_n$ that does not contain $-I \otimes ... \otimes I$ is abelian.
Hint: First show that any element in the subgroup is involutive.

Answer: Let $\mathcal{G}$ be a subgroup of $\mathcal{P}_n$ that does not contain $-I \otimes ... \otimes I$. Let $A = i^k P_1 \otimes ... \otimes P_n \in \mathcal{G}$. Since Pauli matrices are involutive, $A^2 = i^{2k} P_1^2 \otimes ... \otimes P_n^2 = i^{2k} I \otimes ... \otimes I$. Since $-I \otimes ... \otimes I$ is not in the group, but A^2 is, $k \in \{0, 2\}$. Hence, A is involutive. Let $A, B \in \mathcal{G}$. Suppose A and B don't commute. Since they are Pauli strings, they anticommute: $AB = -BA$. Then $ABAB = -AABB = -I \otimes ... \otimes I$ which is not possible by hypothesis. Hence A and B commute. This is true for every pair of elements in $\mathcal{G}$, so it is abelian.

Question 2. Let $S \in \mathcal{N}(\mathcal{G})$ and $|\psi\rangle \in \mathcal{C}(\mathcal{G})$. Show that $S|\psi\rangle \in \mathcal{C}(\mathcal{G})$.

Answer: For all $F \in \mathcal{G}$, $S|\psi\rangle$ is stabilised by F: $F(S|\psi\rangle) = SF|\psi\rangle = S|\psi\rangle$. Hence $S|\psi\rangle \in \mathcal{C}(\mathcal{G})$.

Question 3. Let $S \in \mathcal{P}_n \setminus \mathcal{N}(\mathcal{G})$ and $|\psi\rangle \in \mathcal{C}(\mathcal{G})$. Show that $S|\psi\rangle \in \mathcal{C}(\mathcal{G})^\perp$.

Answer: Since $S \notin \mathcal{N}(\mathcal{G})$, there exists $F \in \mathcal{G}$ such that $SF = -FS$. We can already get: $F(S|\psi\rangle) = -SF|\psi\rangle = -S|\psi\rangle$ so $(S|\psi\rangle) \notin \mathcal{C}(\mathcal{G})$.

Moreover, for all $y \in \mathcal{C}(\mathcal{G})$: $\langle y|S\psi\rangle = \langle Fy|S\psi\rangle = \langle y|FS\psi\rangle = -\langle y|SF\psi\rangle = -\langle y|S\psi\rangle$ which implies $\langle y|S\psi\rangle = 0$. Hence, $S|\psi\rangle \in \mathcal{C}(\mathcal{G})^\perp$.

2 Syndrome Decoding

Let $\mathcal{G}$ be an admissible subgroup of $\mathcal{P}_n$, and $S \in \mathcal{G}$. We define $S_1 := \frac{I+S}{2}$ and $S_{-1} := \frac{I-S}{2}$.

Question 1. Show that $S = S_1 - S_{-1}$ defines a projective measurement.

Question 2. Show that for all $|\psi\rangle \in \mathcal{C}(\mathcal{G})$: $S_1|\psi\rangle = |\psi\rangle$ and $S_{-1}|\psi\rangle = 0$.

Suppose an error $E \in \mathcal{P}_n$ occurs on the codestate $|\psi\rangle \in \mathcal{C}(\mathcal{G})$, and we apply the above projective measurement on the system.

Question 3. Show that if E commutes with S then outcome -1 has probability 0. If instead E does not commute with S, show that outcome 1 has probability 0.

Answer: Seen in class.

3 A Small Stabiliser Code

Let $\mathcal{G} := \langle ZXX, XZX, XXZ\rangle$.

Question 1. Is $\mathcal{G}$ admissible? What are its length and dimension?

Answer: It is admissible, as all Pauli strings pairwise commute. The length of the code is $n = 3$. It can be checked that the tree Pauli strings are independent, so $k = 3 - 3 = 0$.

Question 2. Build a ZX encoder for the state that spans $\mathcal{C}(\mathcal{G})$.

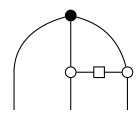
Answer: $\mathcal{G}$ is encoded as the following binary matrix, which can be put in the desired form as follows (with the X-part on the left and the Z-part on the right):

$$\left(\begin{array}{ccc|ccc} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right) \equiv \left(\begin{array}{ccc|ccc} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right) \equiv \left(\begin{array}{ccc|ccc} 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right) \equiv \left(\begin{array}{ccc|ccc} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right)$$

Applying a CZ on the 2 rightmost qubits yields the following code:

$$\left(\begin{array}{ccc|ccc} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right)$$

which can now be seen as a CSS code. We can then apply the construction of CSS codes using either of the two parity check matrices, say H_Z here, to finally get:



Suppose we had instead $\mathcal{G} := \langle ZXX, XZX \rangle$.

Question 3. Is $\mathcal{G}$ still admissible? What are its length and dimension?

Answer: It is still admissible. Its length is still three, and now $k = 1$.

Question 4. Complete the stabiliser tableau as you see fit, then create a ZX encoder for this code.

Answer: We need $n + k$ stabilisers of length $n + k$, so we can choose to complete the tableau as follows:

$$\begin{pmatrix} Z & X & X & I \\ X & Z & X & I \\ I & I & X & X \\ Z & Z & Z & Z \end{pmatrix}$$

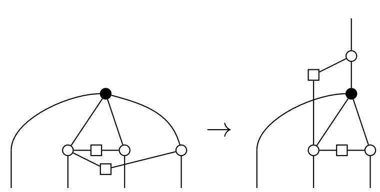
One can check that the resulting group is admissible. We obtain the following binary matrix put in the adequate form:

$$\left(\begin{array}{cccc|cccc} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right) \equiv \left(\begin{array}{cccc|cccc} 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right)$$

The top Z-block can be removed by applying a CZ between qubits 2 and 3; and between qubits 2 and 4. This yields matrix:

$$\left(\begin{array}{cccc|cccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right)$$

which is a CSS code. Combining the CSS code from the remaining H_Z and the above CZs, we end up with:



4 5-Qubit Code

Let $\mathcal{G}$ be the subgroup of $\mathcal{P}_5$ generated by the following tableau:

$$\begin{pmatrix} I & Z & X & X & Z \\ Z & I & Z & X & X \\ X & Z & I & Z & X \\ X & X & Z & I & Z \end{pmatrix}$$

Question 1. Is $\mathcal{G}$ admissible? What are its length and dimension?

Answer: It is admissible. $n = 5$ and $k = 5 - 4 = 1$ since all Pauli strings are independent.

Question 2. We can encode the logical Z operator as $ZZZZZ$ and the logical X operator as $XXXXX$. From this, can we deduce a way to complete the code into a maximal one?

Answer: Putting the input qubit on the right, the resulting tableau is the following:

$$\begin{pmatrix} I & Z & X & X & Z & I \\ Z & I & Z & X & X & I \\ X & Z & I & Z & X & I \\ X & X & Z & I & Z & I \\ X & X & X & X & X & X \\ Z & Z & Z & Z & Z & Z \end{pmatrix}$$

Question 3. Build a ZX encoder for the original code, using the above completion.

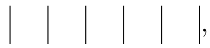
Answer: As done in the previous exercise:

$$\left(\begin{array}{cccccc|cccccc} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right) \equiv \left(\begin{array}{cccccc|cccccc} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right)$$

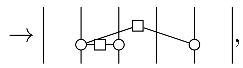
$$\equiv \left(\begin{array}{cccccc|cccccc} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right) \equiv \left(\begin{array}{cccccc|cccccc} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right)$$

$$\equiv \left(\begin{array}{cccccc|cccccc} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right) \equiv \left(\begin{array}{cccccc|cccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right)$$

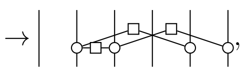
Let's now build the encoder step by step while keeping track of the effect of applications of CZs on the code:



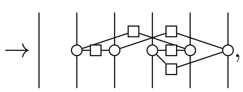
$$\left(\begin{array}{cccccc|cccccc}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}\right)$$



$$\left(\begin{array}{cccccc|cccccc}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}\right)$$

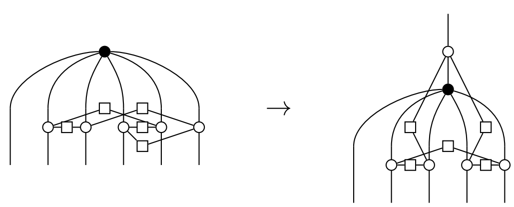


$$\left(\begin{array}{cccccc|cccccc}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}\right)$$



$$\left(\begin{array}{cccccc|cccccc}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}\right)$$

The remaining is a CSS code. Using the H_Z parity check matrix to build the corresponding state, we get:



Question 4. Show that the minimal distance of this code is 3.

Answer: First $XZXII$ is in the normaliser, but not the stabiliser, so the minimal distance is ≤ 3. It can be 1, that would mean there is an all-I column. It can't be 2 either, as there are no two columns that are identical up to a single Pauli. So the minimal distance has to be 3.