CS 150 Quantum Computer Science

Lecture notes on error correcting codes

Spring 2024

Lecturer: Saeed Mehraban *Scribe: Preliminary notes*

These notes are closely aligned with the topics presented in Nielsen Chuang and the quantum computer science course by David Mermin. For more details please read relevant chapters from those references.

1 Classical repetition code

Suppose we want to encode a bit 0 or 1 assuming a model of noise where each bit has a probability p of being flipped. If we encode information using naive encoding, then with probability p, information gets corrupted; this error channel is known as the binary symmetric channel. Such a way of encoding information is not reliable. The repetition code instead repeats each bit at the encoding. For instance, suppose we encode 0 with 000 and 1 with 111. We furthermore decode information by taking a majority vote. In this way, we can correct one bit getting flipped. However, if two or more bits are flipped, our decoding procedure does not give the correct answer. The probability of two or more bits getting flipped is $3p^2(1-p)+p^3 = O(p^2)$. As a result, we suppress the probability of error by a quadratic factor. More generally, if we use the encoding $0 \to 0^k$ and $1 \to 1^k$ and decode using majority vote, then assuming $p < 1/2$, the probability of error decays exponentially fast in k.

Exercise: Prove this.

2 Correcting quantum bit flips

Initially, it may appear that error correction is impossible when comparing classical and quantum information processing due to significant differences between the two frameworks. For instance, due to the no-cloning theorem, we may not copy information. Also, quantum measurements destroy quantum information. Furthermore, the space of errors is continuous and corresponds to a much larger space than the classical domain. For instance, a quantum state may experience Z error, X error, or an error as a linear combination of the two. It is an outstanding discovery that besides all these limitations, quantum error correction is still possible.

Suppose, for now, our noise model is that on each qubit, we have equal probability p of getting an unwanted bit flip X. Let's choose an encoding:

$$0 \to |\bar{0}\rangle := |000\rangle\,, \quad 1 \to |\bar{1}\rangle := |111\rangle\,.$$

we can perform this encoding using a CNOT between first and second and a CNOT between first and third qubits. The circuit implementing this operation is givne in Figure 1. A nice observation is that if we input an arbitrary quantum state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, we obtain $|\bar{\psi}\rangle = \alpha|\bar{0}\rangle + \beta|\bar{1}\rangle$. The main observation is that any quantum state like $|\bar{\psi}\rangle$ belongs to a two-dimensional subspace spanned by $|000\rangle$ and $|111\rangle$. Let's call this subspace, V_0, the error-free subspace. Suppose we apply a bit flip to the first qubit of $|\bar{\psi}\rangle$ and obtain $\alpha|100\rangle + \beta|011\rangle$. Any such quantum state belongs to the subspace of quantum states spanned by $|100\rangle$ and $|011\rangle$. Let's call this subspace V_1. Similarly, if we apply a bit flip operation to the second or third qubit, we obtain a quantum state in $V_2 = \text{span}\{|010\rangle, |101\rangle\}$ and $V_3 = \text{span}\{|001\rangle, |110\rangle\}$. It is crucial to observe that subspaces V_0, V_1, V_2, V_3 are mutually orthogonal. To correct the incident error, we need to measure the subspace V_i ($i = 0, 1, 2, 3$) where the quantum state belongs to. To do this, we use the following POVM

$$\Pi_0 = |000\rangle\langle 000| + |111\rangle\langle 111| \tag{1}$$
$$\Pi_1 = |100\rangle\langle 100| + |011\rangle\langle 011| \tag{2}$$
$$\Pi_2 = |010\rangle\langle 010| + |101\rangle\langle 101| \tag{3}$$
$$\Pi_2 = |001\rangle\langle 001| + |110\rangle\langle 110|. \tag{4}$$

To see why this set of operators corresponds to a POVM, we note that $\Pi_i \geq 0$, and $\Pi_0 + \Pi_1 + \Pi_2 + \Pi_3 = I$. Once we perform this POVM measurement and obtain a label i, we can correct the error. If $i = 0$, we don't need to do anything. But if $i \in \{1, 2, 3\}$, then we can correct the quantum state by applying X_i (X on qubit i).

Another way to understand the error measurement is through the framework of observable measurements. Consider two observables $O_1 = Z_1 Z_2$ and $O_2 = Z_2 Z_3$. If $|w\rangle \in V_0$ then $O_1|w\rangle = |w\rangle$ and $O_2|w\rangle = |w\rangle$, if $|w\rangle \in V_1$ then $O_1|w\rangle = -|w\rangle$ and $O_2|w\rangle = |w\rangle$, if $|w\rangle \in V_2$ then $O_1|w\rangle = -|w\rangle$ and $O_2|w\rangle = -|w\rangle$ and if $|w\rangle \in V_3$ then $O_1|w\rangle = |w\rangle$ and $O_2|w\rangle = -|w\rangle$. In other words, if we measure O_1, O_2 and obtain $x, y \in \pm$, then $+, +$ corresponds to no error, $-, +$ corresponds to bit flip on the first qubit, $-, -$ corresponds to an error on the second qubit and $+, -$ corresponds to an error on the third qubit. We can correct each error correspondingly. This step is known as syndrome measurement.

To implement this measurement using the circuit model, we can use two extra ancillary qubits, both initialized at 0. We perform CNOT between the first qubit and the first ancilla qubit and another CNOT from the second qubit to the first ancillary qubits. This way, we store the parity between the first two qubits in the first ancillary qubit. Similarly, we apply CNOT from the second qubit onto the second ancillary qubit and another CNOT from the third qubit onto the second ancillary qubit. As a result, we obtain the parity between the second and third qubits in the second ancillary qubits. We then measure the ancillary qubits. If we obtain 00 we apply nothing I. If we obtain 10 we apply X_1. If we obtain 11 we apply X_2 and if we obtain 01, we apply X_3. We can implement this step using the SELECT operation we discussed before (using Toffoli gates). See Figure 2 for the implementation.

Exercise: Analyze an explain what happens to this error correcting code if two errors occur.

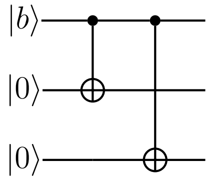

Figure 1: The encoding map which maps $|b\rangle$ to $|\bar{b}\rangle = |bbb\rangle$ for $b \in \{0,1\}$. We note that the same map can be used for decoding, i.e., it transforms the quantum state $\alpha\,|000\rangle + \beta\,|111\rangle$ to $(\alpha\,|0\rangle + \beta\,|1\rangle)\,|00\rangle$.

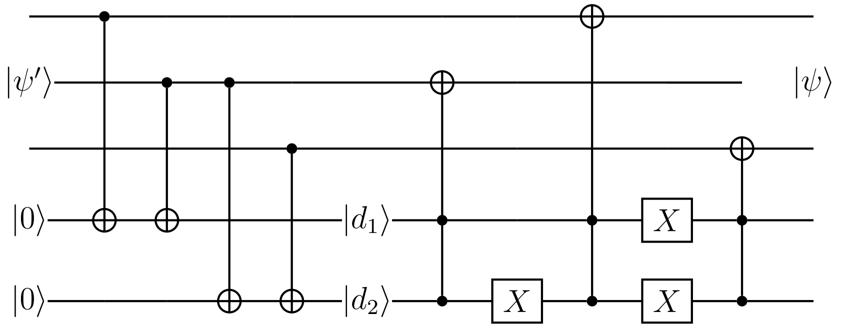

Figure 2: The error correcting map. The input is the quantum state $|\psi'\rangle$ which is equal to $|\psi\rangle$ after going through the noise channel. Assuming at most one bit flip has been applied we can correct this error. If $d_1 = d_2 = 0$ then no correction is needed. If $d_1 = 1, d_2 = 0$ we correct by applying an X operator on the first qubit of $|\psi\rangle$. If $d_1 = 1, d_2 = 1$ we correct by applying an X operator on the second qubit of $|\psi\rangle$. If $d_1 = 0, d_2 = 0$ we correct by applying an X operator on the third qubit of $|\psi\rangle$.

3 Correcting quantum phase flips

In the previous section we showed how we can correct one phase error. What happens if we consider phase flip errors i.e. receiving an unwanted Z error on one of the qubits. As we have seen before the Z operator can be obtained from X by the Hadamard change of basis: $Z = HXH$. Using this observation we consider the following encoding

$$0 \to |\bar{0}\rangle := |+++\rangle\,, \quad 1 \to |\bar{1}\rangle := |---\rangle\,.$$

we can implement this map using the circuit in Figure 5

The decoding map is according to the inverse of this map (and tossing out the two right-most ancillary qubits). The following figure captures this map:

We show that we can correct up to one Z error using this encoding. To see this we observe that the dencoding from previous seciton works exactly the same way if we replace 0 with $+$ and 1 with $-$. For instance, if we start with a quantum state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, after the encoding step we get $|\bar{\psi}\rangle = \alpha\,|+++\rangle + \beta\,|---\rangle$. If we apply a Z gate to the first qubit we get $Z_1\,|\bar{\psi}\rangle = \alpha\,|-++\rangle + \beta\,|+--\rangle$. So, the subspace corresponding to no error is $V_0^Z = \text{Span}\{|+++\rangle\,, |---\rangle\}$, the

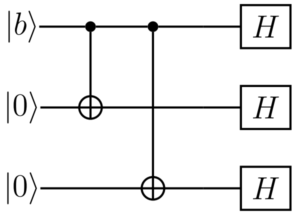

Figure 3: The encoding map which maps $|0\rangle$ to $|\bar{0}\rangle = |+++\rangle$ and $|1\rangle$ to $|\bar{1}\rangle = |---\rangle$.

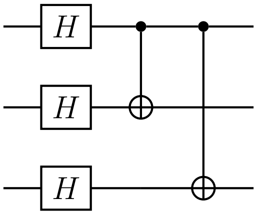

Figure 4: The dencoding map which maps $|0\rangle$ to $|\bar{0}\rangle = |+++\rangle$ and $|1\rangle$ to $|\bar{1}\rangle = |---\rangle$.

subspace corresponding to Z_1 error is $V_1^Z = \text{Span}\{|-++\rangle, |+--\rangle\}$, the subspace corresponding to Z_2 error is $V_2^Z = \text{Span}\{|+-+\rangle, |-+-\rangle\}$, the subspace corresponding to Z_3 error is $V_3^Z = \text{Span}\{|++-\rangle, |--+\rangle\}$. Similar to before, we can detect this error using the following $POVM$:

$$\Pi_0^Z = |+++\rangle\langle+++| + |---\rangle\langle---| \tag{5}$$

$$\Pi_1^Z = |-++\rangle\langle-++| + |+--\rangle\langle+--| \tag{6}$$

$$\Pi_2^Z = |+-+\rangle\langle+-+| + |-+-\rangle\langle-+-| \tag{7}$$

$$\Pi_2^Z = |++-\rangle\langle++-| + |--+\rangle\langle--+|. \tag{8}$$

Also similar to before, to detect this error we can measure the operators $O_1^Z = X_1X_2$ and $O_2^Z = X_2X_3$. We can see that the subspaces V_i^Z correspond to specific eigenspaces of O_1^Z and O_2^Z. For V_0^Z we obtain $+, +$ for the eigenvalues of O_1^Z and O_2^Z, respectively. For V_1^Z we obtain $-, +$, for V_2^Z we obtain $+, +$, and for V_3^Z we obtain $+, -$. The error correcting map is similar to Figure 2 except that we have to apply Hadamard gates in the begining and in the end to all qubits of $|\psi'\rangle$.

4 Shor's 9-qubit code

We already saw how to correct errors in two (incompatible) basis. How can we design a quantum code that corrects both X and Z errors at the same time? The Shor's 9-qubit code achieves this objective. As expected, the code is a combination of blocks involving phase flip correcting code and bit flip correcting code. The code is obtained by first mapping $|0\rangle \rightarrow |+++\rangle$ and $|1\rangle \rightarrow |---\rangle$ and then mapping each of the three qubits according to the bit flip encoding map and obtain

$$|0_L\rangle = (\frac{|000\rangle + |111\rangle}{\sqrt{2}})(\frac{|000\rangle + |111\rangle}{\sqrt{2}})(\frac{|000\rangle + |111\rangle}{\sqrt{2}})$$

and

$$|1_L\rangle = (\frac{|000\rangle - |111\rangle}{\sqrt{2}})(\frac{|000\rangle - |111\rangle}{\sqrt{2}})(\frac{|000\rangle - |111\rangle}{\sqrt{2}})$$

The main observation is that if we apply one X or Z gate we move to mutually orthogonal subspaces. This allows us to detect and correct error.

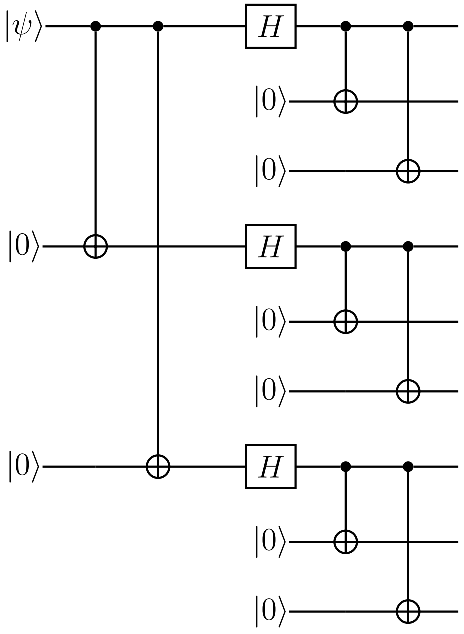

Figure 5: The encoding map for the Shor's code.

To describe the error detection and correction step we use the syndrome measurement language. Now suppose a bit flip occurs on the first qubit. We can detect this error by measuring the syndrome observables Z_1Z_2 and Z_2Z_3 and verifying that we get -1 on the first observable and $+1$ on the second. Similarly to detect bit flip error on the second block by measuring Z_4Z_5, Z_5Z_6, and on the third block by measuring Z_7Z_8, Z_8Z_9. We can correct this error by applying X operator to the faulty bit.

Now we analyze what happens if a phase flip error Z gets applied to one of the qubits. We claim by measuring the syndromes $X_1X_2X_3X_4X_5X_6$ and $X_4X_5X_6X_7X_8X_9$ we can detect and correct phase flip error.

Exercise: Explain how and why this syndrome measurement works. Give a procedure to correct this indecent error.

Exercise: Suppose a Y operation is applied to the first qubit. Analyze the Shor's code and explain how we can detect and correct this error. (Hint: $Y = iXZ$.)

Exercise: Suppose we apply a $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$ to the first qubit. Show that Shor's code is capable of correcting this error.

It turns out that any general quantum operation that involves affecting only one qubit can be corrected using Shor's code. For instance, if we apply a unitary operation $U = \sum_i \alpha_i A_i$ where A_i is a single qubit operation, we can correct this error using Shor's code. The code can correct more general forms of (non-unitary) errors, which is beyond the scope of this course.

5 Stabilizer codes

In this section, we introduce "stabilizer codes" as an important framework for error correction that formally generalizes the three codes discussed so far. Recall the insights we obtained so far. The code space corresponds to a subspace in the Hilbert space. Once an error occurs, the quantum state in the error-free subspace gets mapped to an error subspace, which is orthogonal to the error-free subspace, and hence, they can be perfectly distinguished from each other. Furthermore, we would like each Z or X error on each qubit to map the subspace to subspaces that are mutually orthogonal to each other. Let's do a simple evaluation of how many qubits one needs to correct arbitrary single qubit gates. We saw that this is possible using the 9-qubit Shor's code. Can we do better? There are $3n$ single qubit operators (X, Y, Z on each qubit), so we need $3n + 1$ two-dimensional subspaces. Hence, $2^n \geq 2(3n + 1)$. We can see that to satisfy this criterion, we need $n \geq 5$. We will indeed give a five-qubit error-correcting code.

Before we get there, let's make a few simple observations about the bit flip code. Recall that the syndrome observables for this code are $Z_1 Z_2$ and $Z_2 Z_3$. We observe the following features: (1) these syndromes are tensor products of Pauli operators; hence they have eigenvalues ± 1, (2) they commute with each other (hence allow mutual eigenbasis), (3) they stabilize the code space (i.e., any quantum state of the form $\alpha |000\rangle + \beta |111\rangle$, (4) at least one of the syndrome operators anti-commutes with each of the bit flip errors which they can correct (they, however, commute with the Z errors and they are not able to correct these errors).

Let's study the syndromes of Shor's 9-qubit code $Z_1Z_2, Z_2Z_3, Z_4Z_5, Z_5Z_6, Z_7Z_8, Z_8Z_9$, for bit flip errors and $X_1X_2X_3X_4X_5X_6, X_4X_5X_6X_7X_8X_9$ for phase flip errors. We note that these syndrome operators satisfy all the mentioned criteria. This inspires us to define a more general family of codes, namely the stabilizer codes that generalize all the codes discussed so far.

5.1 The stabilizer formalism

We define the Pauli group $\mathcal{P}_n$ as the group of Pauli strings of size n with phases $\pm, \pm i$. For instance, $iX \otimes Y \in \mathcal{P}_2$ and $-X \otimes X \otimes Z \in \mathcal{P}_3$[1]. All elements in a Pauli group either commute or anti-commute with each other. A quantum state is a stabilizer state if a subgroup of Pauli exists that stabilizes it. In other words,

Definition 5.1. A quantum state $|\psi\rangle$ is a stabilizer state if there exists a subgroup $G \trianglelefteq \mathcal{P}_n$ s.t. $\forall g \in G, g\,|\psi\rangle = |\psi\rangle$. More generally, a subspace of the Hilbert space is a stabilizer subspace if a subgroup of Pauli stabilizes it.

To see why the set of vectors $V \subseteq \mathcal{H}$ stabilized by a subgroup of Pauli constitute a linear subspace, note that if $|\psi\rangle, |\phi\rangle \in \mathcal{H}$, for any $\alpha, \beta \in \mathbb{C}$, $\alpha\,|\psi\rangle + \beta\,|\phi\rangle \in V$. Note that if the subgroup contains $-I, -iI, iI$, elements that have eigenvalues other than ± 1 or elements that anti-commute with each other, then the stabilizer state is trivially the 0 state. Why? We saw before that the elements of the Pauli group either commute or anti-commute with each other. Therefore, nontrivial stabilizer subgroups are commutative. Conversely, we can define a stabilizer group for a linear subspace $V \subseteq \mathcal{H}$.

Definition 5.2. The stabilizer group G_V corresponding to the linear subspace of the Hilbert space $V \subseteq \mathbb{C}^n$, is defined as the largest subgroup $G \trianglelefteq \mathcal{P}_n$ that stabilizes V, i.e., for all $|v\rangle \in V$, $g \in G_V$, $g\,|v\rangle = |v\rangle$.

To see why the set of elements that stabilize a subspace correspond to a group, we note that if two elements g and h stabilize a subspace, so does their multiplication. Furthermore, if g stabilizes V, so does g^{-1}, and clearly, the identity element stabilizes any element. Any subspace of the Hilbert space has a stabilizer subgroup because I by itself is a subgroup of Pauli.

How large is the stabilizer subspace for a given subgroup of Pauli? Suppose a G_V subgroup of $\mathcal{P}_n$ has k generators[2] $g_1, \ldots, g_k$ and stabilizes the subspace $V \subseteq (\mathbb{C})^{\otimes n}$. To get a nontrivial subspace, furthermore assume that all elements of G_V commute with each other and furthermore $g^2 = I$, for all $g \in G_V$; furthermore, except for the identity element, all elements of G_V have zero traces. Our first observation is that the projector onto V is given by

$$\Pi_V = \frac{1}{2^k} \sum_{g \in G_V} g$$

To see this, for each $g \in G_V$, since $g^2 = I$ then $\frac{I+g}{2}$ is the projector onto the $+1$ eigenspace of g. (Similarly, $\frac{I-g}{2}$ corresponds to the -1 subspace.) Since all elements in G_V commute, $\Pi_V = \prod_{g \in G}(\frac{I+g}{2})$. We know that $\dim(V) = Tr(\Pi_V)$. Therefore,

Lemma 5.3. *If the stabilizer subgroup of* V *has* k *generators, then* $\dim(V) = 2^{n-k}$.

[1] Recall that a group is a collection of objects with a multiplication rule, which is (1) closed under multiplication and is associative, (2) has an identity element, and (3) has an inverse element

[2] By $\langle g_1, \ldots, g_k\rangle$, we mean the set of elements generated by compositions of $g_1, \ldots, g_k$; a generator is the smallest set of group elements that generates that group

Intuitively, what this lemma is saying is that each generator of G_V divides the 2^n-dimensional Hilbert space $\mathbb{C}^{\otimes n}$ into two halves, hence the $+1$ subspace of k generators has dimension 2^{n-k}.

Let's work out a few examples. For the Hilbert space of one qubits, $\langle Z \rangle$ is a stabilizer subgroup with $k = 1$ generator. The dimension of the subspace stabilized by this group is $2^{1-1} = 1$ dimensional. We can see that this subspace is exactly the set of vectors spanned by $|0\rangle$. Similarly $\langle -Z \rangle$ stabilizes the subspaces $|1\rangle$. Now let us look at the quantum state $|0 \ldots 0\rangle$ (n zeros). What is the stabilizer group corresponding to this state? Clearly, the subspace is one-dimensional, so $2^{n-k} = 1$ only when $k = n$. Hence, we need to find n generators. It is easy to see that $\langle Z_1, \ldots, Z_n \rangle$ is the stabilizer group. Let us consider the subgroup $\langle XX \rangle$. The stabilizer subspace corresponding to this group has $2^{n-k} = 2$ elements. One of the elements will be $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$ and the other $\frac{|01\rangle+|10\rangle}{\sqrt{2}}$. Let us consider the subgroup $\langle XZ \rangle$. Like the previous example, its stabilizer subspace is a 2-dimensional subspace. This subgroup stabilizes $\frac{|00\rangle+|10\rangle}{\sqrt{2}}$ and $\frac{|01\rangle-|11\rangle}{\sqrt{2}}$. Next, consider the stabilizer subgroup $\langle XX, YY \rangle$. XX and YY commute with each other, and we have a 1-dimensional stabilizer subspace which is spanned by $\frac{|01\rangle+|10\rangle}{\sqrt{2}}$. Finally, consider the example $\langle Z_1Z_2, Z_2Z_3 \rangle$, which corresponds to the syndrome operators for the bit flip error. We expect a $2^{3-2} = 2$-dimensional subspace, which not surprisingly happens to be the code subspace for the bit flip error, i.e., the set of vectors spanned by $|000\rangle, |111\rangle$.

Lastly, we present a useful lemma in the stabilizer formalism:

Lemma 5.4. *Let G_V be the stabilizer subgroup corresponding to a linear subspace V, and let U be any unitary operator, then $G_{UV} = UG_VU^{-1}$.*

Here $UV = \{U|v\rangle : |v\rangle \in V\}$, and $UGU^{-1} = \{UgU^{-1} : g \in G\}$. We leave the proof as an exercise. For instance, $\langle Z_1, Z_2 \rangle$ is the stabilizer subspace of $|00\rangle$. Now let $U = CNOT_{12}H_1$. We know that $U|00\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. We also know that the stabilizer subgroup for $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is $\langle XX, ZZ \rangle$.

Exercise: Verify that $UZ_1U^2 = X_1X_2$, $UZ_2U^{-1} = Z_1Z_2$.

5.2 Stabilizer formalism for error correction

Let us get back to the bit flip code. As portrayed in the previous section, the code (no errors) subspace corresponds to the stabilizer subspace for the group $G_0 = \langle Z_1Z_2, Z_2Z_3 \rangle$. Now consider the error X_1 being applied to a quantum state $|\psi\rangle$ being initially stabilized by G_0. We will obtain $|\psi'\rangle = X_1|\psi\rangle$. Using Lemma 5.4 is now stabilized by $X_1Z_1Z_2X_1 = -Z_1Z_2$ and $X_1Z_2Z_3X_1 = Z_2Z_3$. That is why the syndrome Z_1Z_2 detects a -1 and Z_2Z_3 keeps detecting $+1$. More generally, the syndrome g detects $+1$ if the incident error commutes with g and -1 if it anti-commutes. We can deduce the pattern of $+1, -1$ in syndrome measurements for other bit-flip errors using this window of reasoning. Furthermore, we can understand why these syndromes cannot detect Z errors. That is because Z errors commute with the syndromes. Moreover, we can understand why these syndromes fail to detect X_1X_2 errors correctly. That is because, for instance, this error term commutes with Z_1Z_2 and anti-commutes with Z_2Z_3, so it incorrectly detects X_1 error. Let's look at the syndromes of Shor's code. Recall the syndromes are $Z_1Z_2, Z_2Z_3, Z_4Z_5, Z_5Z_6, Z_7Z_8, Z_8Z_9$, for bit-flip errors and $X_1X_2X_3X_4X_5X_6, X_4X_5X_6X_7X_8X_9$. As an exercise, show that if a single qubit X or Z error occurs, we measure -1 when the error anti-commutes with a syndrome and $+1$ when it commutes.

Suppose we measured the syndrome, and we are sure that not more than one error has been applied. How should we decide what correction circuit we should apply? We need to find the set of Pauli operators that anti-commute with the syndromes that are measured to be -1 and commute with those operators we measured to be $+1$. In the case of the example above, X_1 is specifically the operator that anti-commutes with Z_1Z_2 and commutes with Z_2Z_3.

How do we find the correction circuit systematically? We will do this next. But before that, let's define some notation. Let $a \in \mathbb{F}^n$ and $X^a = X_1^{a_1} \dots X_n^{a_n}$ (similarly for Z). Let's use the notation $P_{a,b} = i^{-a \cdot b} X^a Z^b$ to capture arbitrary Pauli strings, where $a \cdot b = a_1 b_1 + \ldots + a_n b_n$ is the usual inner product. We chose $c = i^{-a \cdot b}$ as the overall phase so that $P_{a,b}^2 = I$[3]. How do we capture Y using this notation? We leave it as an exercise. We can show that

$$P_{a,b} P_{a',b'} = (-1)^{a \cdot b' + a' \cdot b} P_{a',b'} P_{a,b}$$

Let

$$\Lambda := \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix}.$$

then $a \cdot b' + a' \cdot b = (a, b) \Lambda \begin{pmatrix} a' \\ b' \end{pmatrix}$. This is called the symplectic inner product. As a result $P_{(a,b)}$ commutes with $P_{(a',b')}$ iff (a, b) is orthogonal to (a', b') according to the Symplectic inner product. The question of finding a Pauli string that commutes with a given set of syndromes and anti-commutes with others can be, therefore, captured according to a linear algebra problem over $\mathbb{F}_2^n$, $Aa = s$ where s is the vector of syndromes (0 for $+1$ and 1 for -1).

5.3 The five qubit code

As promised, in this section, we describe an error-correcting code encoding one logical qubit and correcting single-qubit errors with five qubits. To encode a two-dimensional error-free subspace, we need to provide $k = 4$ syndrome measurements; to see this, recall 2^{n-k} is the dimension of the stabilized subspace, so $k = n - 1 = 4$ gives us a two-dimensional subspace). Consider the syndrome measurements:

$$\langle XZZXI, IXZZX, XIXZZ, ZXIXZ \rangle$$

Fortunately, you have all the tools to analyze this code. So we leave it as an exercise to you.

Exercise: Prove the following features

1. Prove that all syndrome elements commute.

[3] We equate $c^2 X^a Z^b X^a Z^b = c^2 (-1)^{a \cdot b} = I$, So $c = i^{-a \cdot b}$ works

2. Write an expression for the projector onto the code subspace.

3. Suppose there is a bit-flip error on one of the qubits; find the pattern of $\pm$ in syndrome measurements. Repeat the same with phase-flip errors.

4. Show that $XXXXX$ performs logical X and $ZZZZZ$ performs logical Z.

5. Find an expression for the logical 0 and 1. (Disclaimer: this may be a lengthy expression).

6 The Gottesman-Knill theorem

An important result in the theory of quantum computing is the Gottesman-Knill theorem, which introduces an important family of quantum computations, namely Clifford circuits, that generate large entanglement but can be simulated on a classical computer. We included this result in this handbook because it heavily builds on the stabilizer formalism we introduced in the previous sections.

Recall the Clifford gateset from previous lectures consisting of

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \quad \text{(Hadamard)}$$

$$S = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix} \quad \text{(Phase)}$$

$$CNOT = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix} \quad \text{(CNOT)}$$

While Clifford is not known to be universal If we add another gate $T = \begin{pmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{pmatrix}$, also known as the $\pi/4$ phase shift, to the Clifford gates, we obtain a universal gate set. What about the Clifford gateset? Can we perform universal quantum computation based on the Clifford gateset alone? The Gottesman-Knill theorem gives strong evidence that the answer should be "no", by giving a classical algorithm that simulates this gateset.

Theorem 6.1 (The Gottesman-Knill theorem)**.** *There is a classical algorithm that takes the description of a quantum circuit from the Clifford gateset C and samples from the output of $C|0\dots0\rangle$ in polynomial time.*

To set up the ground, let's study some basic features of the Clifford gateset. Among the operations this gateset generates include the Pauli strings. To see this note $S^2 = Z$, $HZH = X$ and $HXH = Z$. As a matter of fact, if we conjugate any Pauli string with any of the Clifford gates, we obtain other Pauli strings:

$$(CNOT)X_1(CNOT) = X_1X_2, \quad (CNOT)X_2(CNOT) = X_2, \tag{9}$$
$$(CNOT)Z_1(CNOT) = Z_1, \quad (CNOT)Z_2(CNOT) = Z_1Z_2, \tag{10}$$
$$SXS^{-1} = Y, \quad SZS^{-1} = Z \tag{11}$$

By induction, if we conjugate any Pauli string with any Clifford circuit C we obtain another Pauli string. Now recall the stabilizer formalism. Suppose we start with a subspace of quantum states V that are stabilized by a given set of Pauli stabilizer generations $\langle g_1, \ldots, g_k\rangle$. Let C be a Clifford circuit and let $CV = \{C|\psi\rangle : |\psi\rangle \in V\}$. Then CV is also a stabilizer subspace and is stabilized by $\langle Cg_1C^\dagger, \ldots, Cg_kC^\dagger\rangle$. This is the main idea behind the Gottesman-Knill algorithm. In particular, consider the quantum state $|0\ldots0\rangle$. We discussed that this is a 1-dimensional subspace stabilized by $\langle Z_1, \ldots, Z_n\rangle$. In order to store the description of the quantum state $C|0\ldots0\rangle$ we store $\langle CZ_1C^\dagger, \ldots, CZ_nC^\dagger\rangle$. Let C be a polynomial-size circuit in $CNOT, S$, and H. We can use Equations 9, 10, and 11 to update the set of generators at each step. Using the representation $P_{a,b} = i^{-a\cdot b}X^aZ^b$, for $a, b \in \mathbb{F}^n$, we can capture the Clifford operations using basic linear operations on over $\mathbb{F}^n$. For instance $(CNOT)X_1^{a_1}X_2^{a_2}(CNOT) = X_1^{a_1}X_2^{a_1\oplus a_2}$.

Exercise: Show that if C is a Clifford operation then $CP_{(a,b)}C^\dagger = (-1)^{g(a,b)}P_{f(a,b)}$ for suitable functions $f : \mathbb{F}^{2n} \to \mathbb{F}^{2n}$, $g : \mathbb{F}^{2n} \to \mathbb{F}$. Describe f and g for basic Clifford gates $X, Y, Z, CNOT, H, S$.

It remains to describe the measurement process. Suppose the set of generators before the measurement is $G = \langle g_1, \ldots, g_n|$, and we want to measure a specific Pauli element g. Let $|\psi\rangle$ be the state of the quantum computer. There are two cases. (1) If g commutes with all elements g_i. Therefore $g_ig|\psi\rangle = gg_i|\psi\rangle = g|\psi\rangle$. As a result, $g|\psi\rangle$ is itself in V_G. Therefore either $g|\psi\rangle = |\psi\rangle$ or $g|\psi\rangle = -|\psi\rangle$. So either g or $-g$ belongs to G. In the former case, we obtain $+1$, and in the latter case, we obtain -1. In either case, we don't have to update the stabilizer set. (2) If g anti-commutes with one or more generators. Without loss of generality, we can assume g anti-commutes with g_1 and commutes with the rest. The reason is that if it anticommutes with g_2, then we can replace g_2 with g_1g_2 which commutes with g. We claim that in this case, we obtain $+1$ half of the time and -1 half of the time. To see this note $Pr(+1) = \langle\psi|\frac{I+g}{2}||\psi\rangle\rangle = \langle\psi|\frac{I+g}{2}g_1||\psi\rangle\rangle = \langle\psi|g_1\frac{I-g}{2}||\psi\rangle\rangle = Pr(-1)$. We flip a coin. If it was heads, we sample $+1$ and replace g_1 with Z_1 and if it was tails, we sample -1 and replace g_1 with $-Z_1$.