Stabilizer codes + Fault-tolerance

Stabilizer codes are a prominent class of quantum codes that generalize, contain, classical linear codes.

What is a classical linear code?

it encodes k bits into n bits via a binary generator matrix G $(n \times k)$

write k-bit message as a column vector x. Then encoded message is $Gx = y$

Generator matrix for repetition code is $\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ so that $G[0] = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

$G[1] = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

Classical linear codes have a compact specification:

kn bits to specify generator matrix.

generally we need $n2^k$ bits for an arbitrary code.

How to error correct?

The notion of parity check matrix is important.

It is an $n-k \times n$ matrix H such that

$Hy = 0$

that is, the codewords are in the null space of H.

Equivalently,

$HG = 0$

so the check matrix & generator matrix are dual to each other

This makes sense. The $n-k$ rows of H are chosen to be linearly independent, & so its null space has k vectors, which become the columns of G.

Parity check matrix of repetition code is

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Parity check matrix simplifies error correction procedure.

Encode k-bit message x as

$$Gx = y$$

Send y over channel & get

$$y' = y + e$$

then compute

$$Hy' = H(y+e)$$
$$= Hy + He$$
$$= He$$

↑ this is the $n-k$ bit error syndrome that we use for error correction

The distance of a code is the minimum Hamming distance between any two codewords:

$$d(C) \equiv \min_{x,y \in C, x \neq y} d(x,y)$$

$$d(x,y) = wt(x+y)$$

Since code is linear, $x+y$ is a codeword & then

$$d(C) = \min_{x \in C, x \neq 0} wt(x)$$

C is then an $[n, k, d]$.

Importance of minimum distance is through the use of minimum distance decoder.

If corrupted codeword is y', then min. distance codeword corrects t-bit errors by decoding y' as unique codeword y satisfying $d(y, y') \leq t$ & can do so uniquely if $2t+1 = d$

Interesting code is Hamming code w/ check matrix $[7, 4, 3]$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Code can correct a single-bit error.

Why?

syndrome is a unique binary representation of which error occurred

the Hamming code is self-dual, in the sense that the binary inner product of one row w/ another is equal to zero. Helpful for constructing a quantum code.

Stabilizer codes

Let G_n denote the phase-free Pauli group.

Let S be a subgroup of G_n.

Define V_S to be a vector space stabilized by S.

of n-qubit states

I.e., for $|\psi\rangle \in V_S$ & $g \in S$, we have that

$$g|\psi\rangle = |\psi\rangle$$

S is called stabilizer of V_S

in applications, V_S is codespace

& $g_1, \ldots, g_e \in S$ play role

of parity check matrix,

for repetition codes, we have

$$V_S = \text{span}\{|000\rangle, |111\rangle\}$$

$$\Downarrow \quad S = \{I, Z_1Z_2, Z_2Z_3, Z_1Z_3\}$$

In general, to have a stabilizer we require the generators to commute.

Suppose that g_1 & g_2 are Paulis that anticommute: $g_1 g_2 = -g_2 g_1$

Then if g_1 & g_2 stabilize $|\psi\rangle$, $|\psi\rangle$ must be trivial.

Consider that

$$|\psi\rangle = g_1|\psi\rangle = g_2 g_1|\psi\rangle$$

$$= -g_1 g_2|\psi\rangle$$

$$= -g_1|\psi\rangle$$

$$= -|\psi\rangle \Rightarrow |\psi\rangle = 0$$

7-qubit Steane code is a nice example of a stabilizer code

I	I	I	X	X	X	X
I	X	X	I	I	X	X
X	I	X	I	X	I	X
I	I	I	Z	Z	Z	Z
I	Z	Z	I	I	Z	Z
Z	I	Z	I	Z	I	Z

by representing $I = [0|0]$

$X = [1|0]$

$Z = [0|1]$

$Y = [1|1]$

we can write as

$$\left[\begin{array}{c|c} H & 0 \\ 0 & H \end{array}\right]$$

where H is parity check matrix for classical Hamming code.

[7, 1, 3] quantum code

So this corrects an arbitrary single qubit error.

logical operators are

$\bar{X} = XXXXXXX$

$\bar{Z} = ZZZZZZZ$

Pauli operators commute iff the symplectic product of their binary representations is equal to zero.

where

$v(g)\,\Lambda\, v(g')^T$

where $\Lambda = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}$

the normalizer of a stabilizer code consists of all U such that

$$Ug = gU \quad \forall\, g \in S$$

A generator for the normalizer consists of the $n-k$ stabilizer generators & the $2k$ logical operators, & so has $n+k$ generators.

error correction conditions for stabilizer codes:

Suppose that $\{E_j\}$ is a set of Pauli errors such that

$E_j^\dagger E_k \notin N(S) - S \quad \forall j,k$ then

$\{E_j\}_j$ is a correctable set of errors.

If $E_j^\dagger E_k \notin N(S) - S$ then either

1) $E_j^\dagger E_k \in S$ or 2) $E_j^\dagger E_k \in G_n - N(S)$

Suppose 1)

projector onto code space is

$$P = \prod_{\ell=1}^{n-k} \frac{(I + g_\ell)}{2^{n-k}}$$

then $P E_j^\dagger E_k P = P$ b/c projection is invariant under multiplication by stabilizer.

Suppose $E_j^\dagger E_k \in G_n - N(S)$.

Then $E_j^\dagger E_k$ anticommutes w/ at least one element of S. Call it g_1.

Then

$$E_j^\dagger E_k P = (I - g_1) E_j^\dagger E_k \frac{\prod_{\ell=2}^{n-k} I + g_\ell}{2^{n-k}}$$

& so

$$P E_j^\dagger E_k P = 0 \quad \text{b/c}$$

$$P(I - g_1) = 0 \quad \text{given that}$$

$$(I + g_1)(I - g_1) = 0$$

Thus we have that

$$P E_j^\dagger E_k P = P \quad \text{if}$$

$$E_j^\dagger E_k \in S \quad \&$$

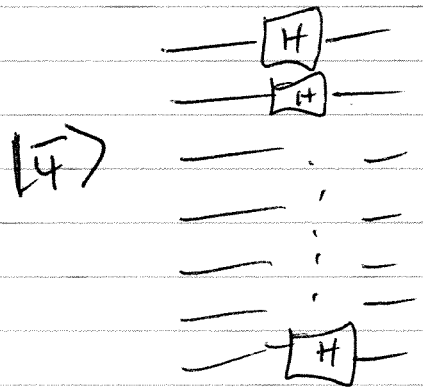
$$P E_j^\dagger E_k P = 0 \quad \text{when } E_j^\dagger E_k \in G_n - N(S)$$

So QEC conditions are satisfied.

Fault-tolerant constructions of gates w/ Steane code.

it has the transversal property for Hadamard & CNOT.

I.e., to perform Hadamard do



$\bar{H}$ takes

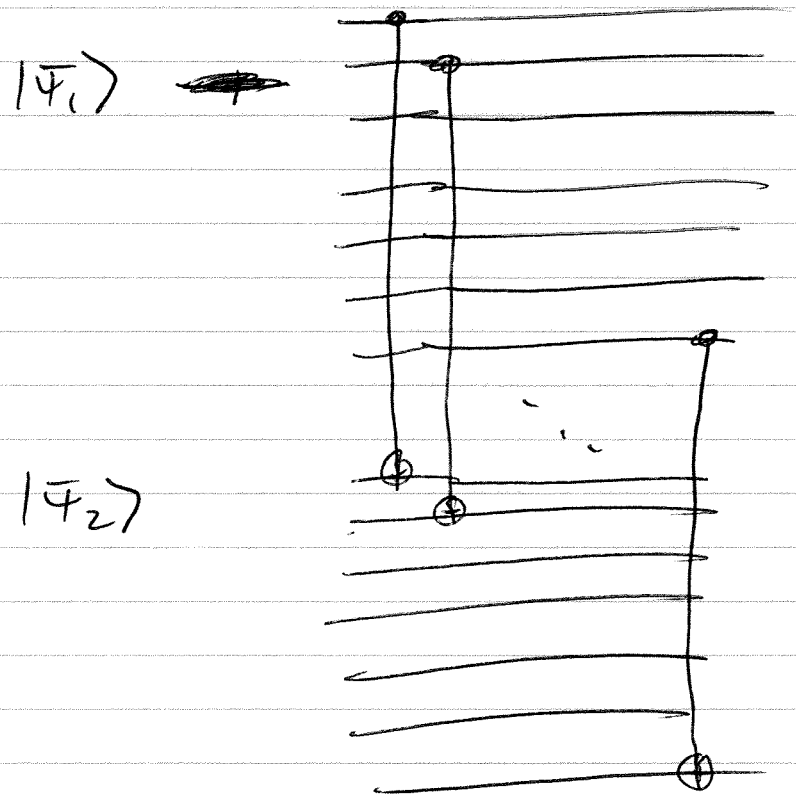
$\bar{H}\bar{X}\bar{H} = \bar{Z}$ &

$\bar{H}\bar{Z}\bar{H} = \bar{X}$

Why is this useful? Suppose that any individual component has failure probability p. Then probability that circuit introduces two or more errors is $O(p^2)$. We can correct a single error.

Same construction for phase gate & Pauli gates

For CNOT, we have transversal implementation as

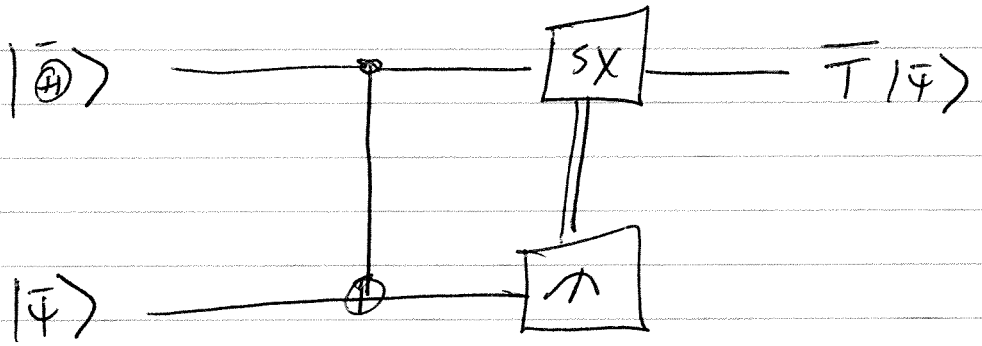


a failure of any single component causes an error in each block, but these are correctable, b/c they correspond to single-qubit errors in each block.

fault tolerant T gate:

prepare resource state $|\bar{\textcircled{T}}\rangle = \frac{|\bar{0}\rangle + e^{i\pi/4}|\bar{1}\rangle}{\sqrt{2}}$

offline & then do

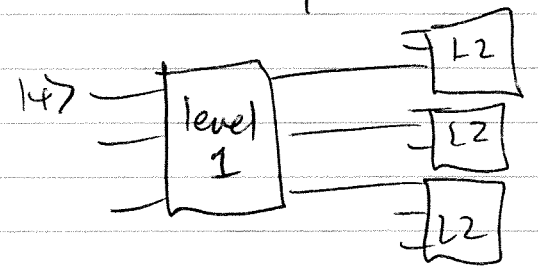


we won't cover it, but there is fault-tolerant procedure to prepare $|\bar{\textcircled{T}}\rangle$

Main idea behind threshold theorem

concatenation

recursively encode as



if failure prob. of each qubit is p, then after one level it becomes cp^2

2nd level $c(cp^2)^2$

Concatenating k times gives

failure probability $\frac{(cp)^{2^k}}{c}$ while size of simulating circuit goes as d^k where d is a constant depending on EC overhead for a single level.

If we want to simulate a circuit w/ $p(n)$ gates w/ accuracy ϵ, then we require k concatentions such that

$$\frac{(cp)^{2^k}}{c} \leq \frac{\epsilon}{p(n)}$$

w/ $p < \frac{1}{c}$, we can choose k such that the above is satisfied

then simulating ~~circuit~~ size is for each gate

$$d^k = \left\{ \frac{\log(p(n)/c\varepsilon)}{\log(1/pc)} \right\}^{\log d}$$

$$= O(\text{poly}(\log(p(n)/\varepsilon)))$$

$\Rightarrow$ # gates is

$$O\left(\text{poly}\left(\log\left(p(n)/\varepsilon\right)\right) p(n)\right)$$

only poly log larger than original circuit.