Part II : Fault-tolerance

Lecture I: DEFINITION OF FAULT TOLERANCE

1

So far we have only considered error models where data qubits are affected by errors.

But what if the error correction circuits themselves are also noisy?

This is the case in current (and most likely) future quantum hardware.

How do we run long computations when all parts of the circuits are noisy?

[Classical hardware is so reliable that we don't need to worry about this issue]

Fault tolerant error correction

Like putting out a fire with a fire extinguisher that is also on fire!

Definition of fault tolerance

Def : circuit noise
error model

Given a circuit, break it up into locations, where a location is a gate (1 qubit, 2 qubit, maybe 3 qubit), a measurement, a state preparation (generally $|0\rangle$), or a storage/wait location.

Assume that classical computations 'of modest size' are perfect and istantaneous.

what this means depends on context

For a location ℓ assume that with prob. $1-p_\ell$ the location functions as intended.

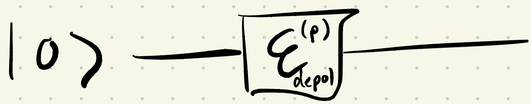
And with probability p_ℓ the location ℓ is replaced by an unknown quantum channel $\mathcal{E}_\ell$. We usually assume that $\mathcal{E}_\ell$ maps qubits to qubits and that each error channel is independant. Commonly $\mathcal{E}_\ell$ just depends on the type of location.

We often assume

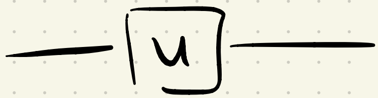
depolarizing channel w/ prob p

- state prep $|0\rangle$ —

↓

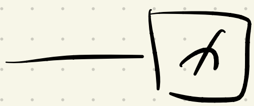


- gate

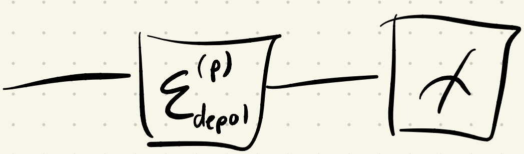


↓

- measurement



↓



Sometimes we use different error probabilities for different types of location e.g. 2-qubit gates are usually more error-prone than 1-qubit gates.

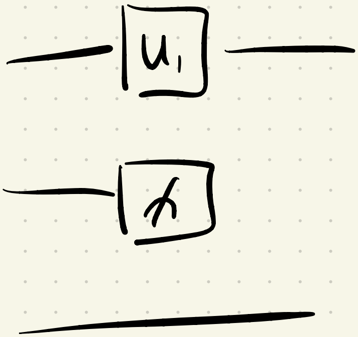
This is by no means the most general error model!

⌜In a later lecture we will discuss extensions⌟

Fault-tolerance is a surprisingly slippery concept to define.

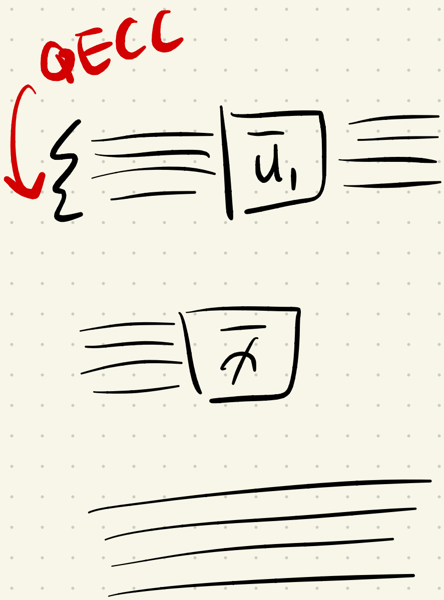
The basic idea is that we encode the qubits of the circuit in a quantum error-correcting code and we replace each physical location with a corresponding logical location. We want the logical locations to not spread errors 'too much'. We also periodically do error correction to prevent the build-up of errors.

This usually looks something like



physical circuit

↓



logical circuit

bar denotes logical location

Defn: FT QEC

Let $\mathcal{C}$ be an $[[n,k,d]]$ stabilizer code & let $t = \lfloor \frac{d-1}{2} \rfloor$. An error correction protocol for $\mathcal{C}$ is FT if:

1. For an input codeword $|\psi\rangle$ with error of weight s_1, if s_2 faults occur during

the protocol w/ $s_1 + s_2 \leq t$ 13

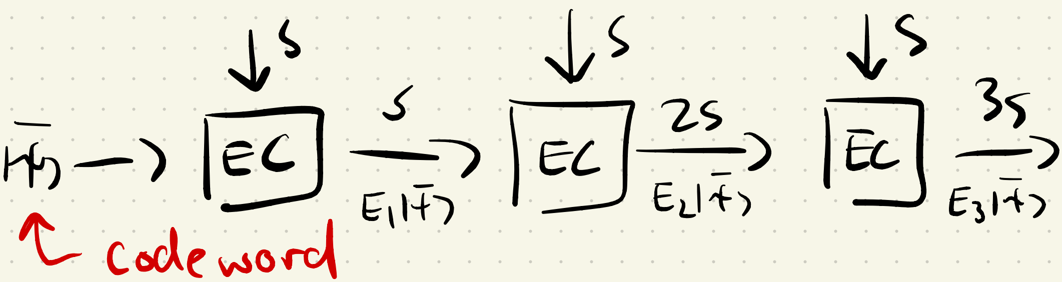
then perfectly decoding the output state gives $|\psi\rangle$.

2 For $s \leq t$ faults ocurring during the protocol for an arbitrary input state the output state differs from a codeward by an error of weight $\leq s$.

1 Ensures that correctable errors don't spread to uncorrectable errors during the course of the protocol.

To understand why 2 is necessary let $\frac{t}{n} < s < \frac{2t+1}{3}$ where $n \in \mathbb{Z}^+$, e consider a QEC protocol where r input errors and s errors during the protocol result in an output with at most $r+s$ errors. 1 is satisfied.

Now suppose we apply the protocol j times



When $j > n$ the input state to EC will have $ns > t$ errors! Failure after linear number of steps. :(

But if 2 also holds

Input $|\bar{\psi}\rangle$

After EC output is $E_1|\bar{\psi}\rangle$

where $wt(E_1) \leq s$

After 2nd EC output is

$E_2|\bar{\psi}\rangle$, where $wt(E_2) \leq 2s$

But by 2 output is also

$E_2'|\bar{\phi}\rangle$ where $|\bar{\phi}\rangle$ is a codeword and $wt(E_2') \leq s$

$E_2'|\bar{\phi}\rangle = E_2|\bar{\psi}\rangle$

$$|\bar{\phi}\rangle = E_2'^{\dagger} E_2 |\bar{\psi}\rangle$$

$$\mathrm{wt}(E_2'^{\dagger} E_2) \leq 3s$$

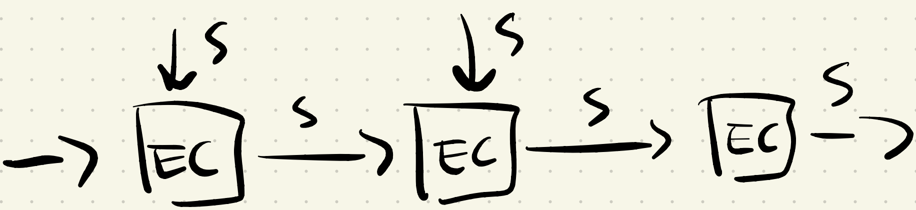
as $\mathrm{wt}(E_2) \leq 2s$ & $\mathrm{wt}(E_2') \leq s$

By assumption $3s < \underbrace{2t+1}_{d}$ code dist.

$$\Rightarrow |\bar{\psi}\rangle = |\bar{\phi}\rangle$$

$$\mathrm{wt}(E_2) = \mathrm{wt}(E_2') \leq s$$

ie



We can write similar defns for all location types e.g. for a logical gate if the input has s_1 errors & s_2 errors occur during the gate where $s_1 + s_2 \leq t$ then ideally decoding the output gives the same thing as ideally decoding the input after applying the gate with no errors.

Upshot: to construct a FT circuit we need to construct

1. FT error correction
2. FT state prep
3. FT measurement

Lecture 2

4. FT gates Lecture 3 + 4

<u>Aside</u> : the defn of fault-tolerance we just discussed is perhaps too stringent e.g. surface code error correction fails to satisfy this defn. However it is the right defn for proving threshold thm w/ concatenated codes,

as we will see in

Lecture 5.

Post script

For an 'operational' defn of fault tolerance see

arXiv.org/abs/1610.03507

For a discussion of the defn of fault tolerance see

https://youtu.be/FMXFNCIaF3k