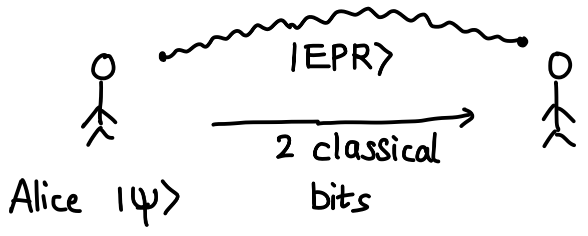
LECTURE 9 February 18th, 2025

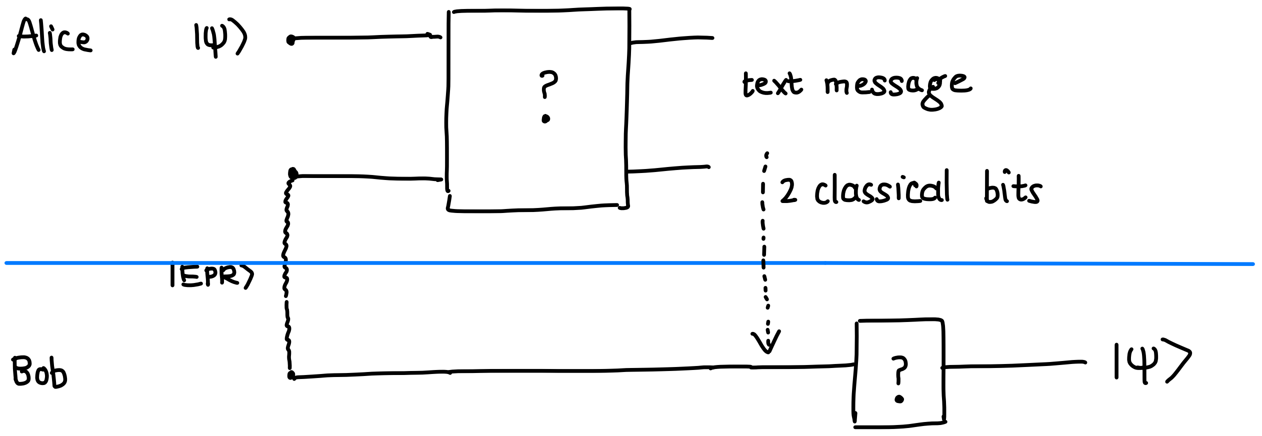
Today: Exchanging Quantum Information (contd)
└ Quantum Teleportation & Holevo's Theorem

PART II: Fundamental Quantum Algorithms
└ Basics of Quantum Computing

RECAP Quantum Teleportation



Alice has $|\psi\rangle$
Alice & Bob share an EPR pair
They can exchange classical messages



Moral "1 ebit + 2 classical bits $\geq$ 1 qubit"

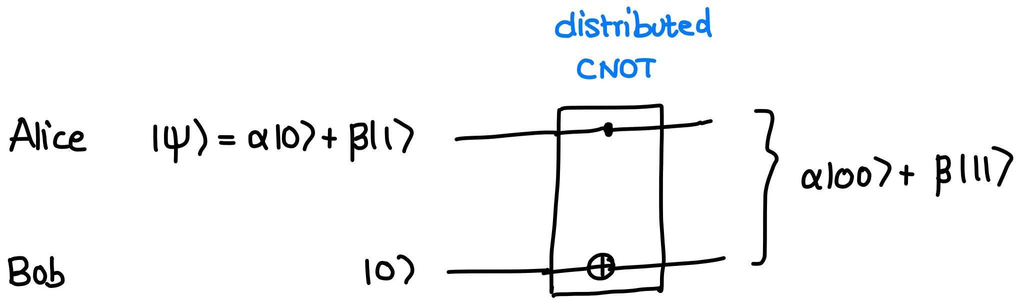
Even if Alice knew the description of $|\psi\rangle$ you would think she needs to send many bits to describe the amplitudes, but here she only sends two bits & Bob gets a perfect copy of $|\psi\rangle$

How does this work ?

Suppose that Alice and Bob could do a distributed CNOT gate where Alice has the control qubit & Bob has the target

We will describe how they can do this in a bit but let's proceed assuming this

Why CNOT? It's the thing that's most similar to copying


distributed CNOT
Alice
$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$
Bob
$|0\rangle$
$\alpha|00\rangle + \beta|11\rangle$

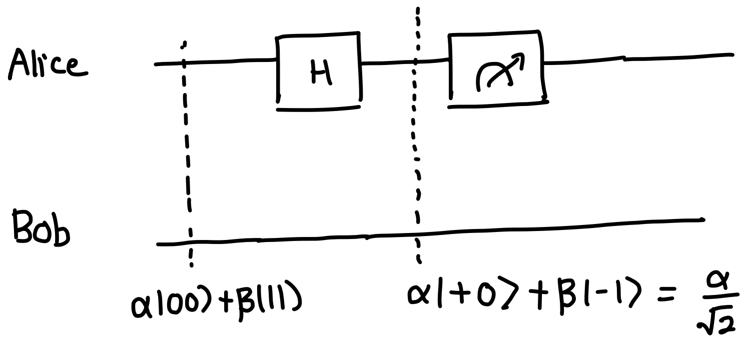
At the end Bob is supposed to have $|\psi\rangle$ and Alice's copy is supposed to be destroyed

What can she do ? Measure her qubit ?

$\boxed{\text{Try 1}}$ Alice measures in the standard basis → w/prob. $|\alpha|^2$ Bob has $|0\rangle$
$|\beta|^2$ Bob has $|1\rangle$ Not what we wanted

$\boxed{\text{Try 2}}$ Alice measures in the $|\pm\rangle$ basis

Let's simulate this with "Alice applies a Hadamard gate & measures in the std. basis"



$\alpha|+0\rangle + \beta|-1\rangle = \frac{\alpha}{\sqrt{2}}|00\rangle + \frac{\alpha}{\sqrt{2}}|10\rangle + \frac{\beta}{\sqrt{2}}|01\rangle - \frac{\beta}{\sqrt{2}}|11\rangle$

After Alice measures

$\mathbb{P}[\text{ outcome } 0] = \left|\frac{\alpha}{\sqrt{2}}\right|^2 + \left|\frac{\beta}{\sqrt{2}}\right|^2 = 1/2$

Bob's qubit becomes $\alpha|0\rangle + \beta|1\rangle$ which is $|\psi\rangle$ ☺

$\mathbb{P}[\text{outcome } 1] = \frac{1}{2}$

Bob's qubit becomes $\alpha|0\rangle - \beta|1\rangle$ → Almost $|\psi\rangle$ but not quite

Alice texts Bob her measurement outcomes

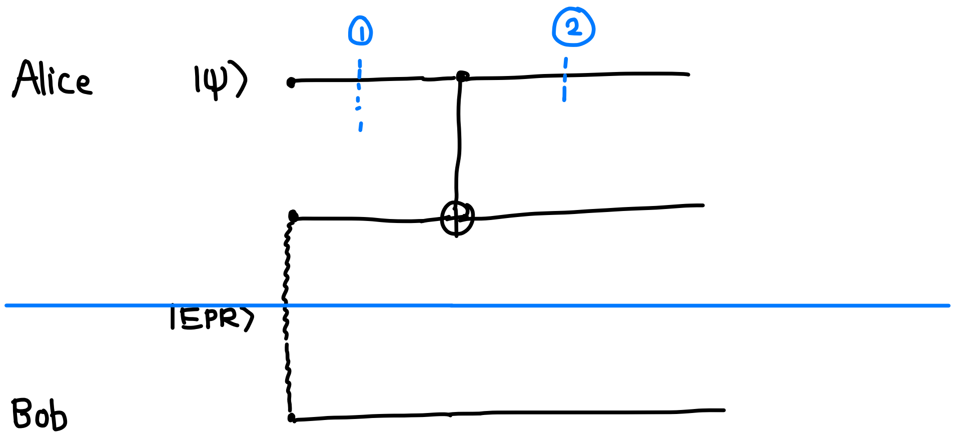
If 0 : Bob does nothing

If 1 : Bob applies the Z gate $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ $|0\rangle \to |0\rangle$
$|1\rangle \to -|1\rangle$

& Bob has $|\psi\rangle$ ☺

Note : Alice's qubit is destroyed after she measures

How do they perform a distributed CNOT without getting together ?



Let's say Alice first does a local CNOT

At time ① the state of all three particles is

$$(\alpha|0\rangle+\beta|1\rangle)\otimes\left(\frac{1}{\sqrt{2}}|00\rangle+\frac{1}{\sqrt{2}}|11\rangle\right) = \frac{\alpha}{\sqrt{2}}|\underset{\substack{\uparrow\\ \text{Control}}}{0}\overset{\substack{\text{Target}\\ \downarrow}}{0}0\rangle + \frac{\alpha}{\sqrt{2}}|011\rangle + \frac{\beta}{\sqrt{2}}|100\rangle + \frac{\beta}{\sqrt{2}}|111\rangle$$

At time ②, the state is $\frac{\alpha}{\sqrt{2}}|000\rangle + \frac{\alpha}{\sqrt{2}}|011\rangle + \frac{\beta}{\sqrt{2}}|110\rangle + \frac{\beta}{\sqrt{2}}|101\rangle$

Now what ? At the end Alice's wants to do a distributed CNOT with first qubit as control and Bob's qubit as target
She measures the second qubit

$$\frac{\alpha}{\sqrt{2}}|0\underline{0}0\rangle + \frac{\alpha}{\sqrt{2}}|0\underline{1}1\rangle + \frac{\beta}{\sqrt{2}}|1\underline{0}0\rangle + \frac{\beta}{\sqrt{2}}|1\underline{1}1\rangle$$

$\mathbb{P}[\text{outcome "0"}] = \left|\frac{\alpha}{\sqrt{2}}\right|^2 + \left|\frac{\beta}{\sqrt{2}}\right|^2 = \frac{1}{2}$

State of particles 1 & 3 is $\alpha|00\rangle + \beta|11\rangle$ → Distributed CNOT !!!

$\mathbb{P}[\text{outcome "1"}] = \frac{1}{2}$

State of particles 1 & 3 is $\alpha|01\rangle + \beta|10\rangle$ ☹ Not quite what we want

But Alice can text Bob what she measured :
if "0" : Bob does nothing
if "1" : Bob applies NOT gate to his qubit

Then, the state is $\alpha|00\rangle + \beta|11\rangle$ ← They have performed a distributed CNOT !

How much information can be encoded in qubits?

n-qubit state

$|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$ has 2^n complex amplitudes

On the surface it looks like it contains an exponential amount of information

In Quantum Computing, we want to harness this to our advantage

But as we have seen, we can only get information by measurements which changes the quantum state, so there is a delicate balance here

If we have n qubits, how much classical information can we store?

Can we use n qubits as a "quantum hard drive" to store much more than n classical bits?

Holevo's Theorem says that for information storage quantum bits are not much better than classical bits

Theorem

Alice has an m-bit string X that she wants to transmit to Bob. She wants to encode X in some n-qubit state $|\psi_X\rangle$ s.t. Bob can do local operations to try to decode X

Bob only gets X with high probability if $n \geqslant m$

$\Rightarrow \mathbb{P}[\text{Bob decodes } X \text{ correctly}] \leq 2^{n-m}$

More to this story

If Alice & Bob share an EPR pair she can send only $\frac{m}{2}$ qubits & Bob can recover with high probability.

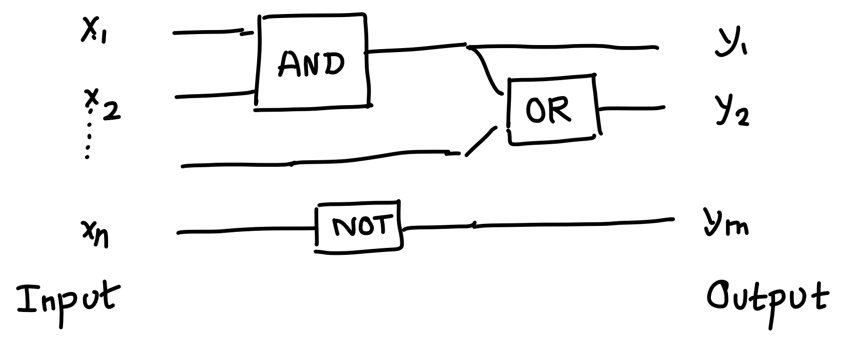
" 1 ebit + 1 qubit $\geqslant$ 2 classical bit "

This is called superdense coding & very similar to teleportation

Basics of Quantum Computing

First let us start with the basics of classical computing

Classical Circuit C



Computes a function $F: \underset{n\text{-bits}}{\{0,1\}^n} \longrightarrow \underset{m\text{-bits}}{\{0,1\}^m}$ ← Typically just consider $m=1$ since we can output bit by bit

E.g $x_1, \ldots x_n$ = bit representation of a large number
y_1 = if x is prime or not

Our focus will be on efficiency — design circuits with fewest number of gates in particular, how does the # gates scale with n ?
Is it n^2 or 2^n ?

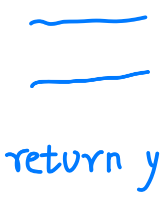
# Gates in a classical circuit roughly corresponds to number of time steps an algorithm takes

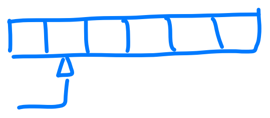
# Gates ≈ # time-steps

How would you implement this as a program or a Turing Machine ?

E.g. python def F(x):

OR Turing Machine





FACT: Given python code that computes F in T steps on length n-inputs, one can produce a circuit using {AND, OR, NOT} gates that computes F and has $c_{\text{pypthon}} \cdot T \log T$ gates ?

Shannon's Theorem (1937) Every $F : \{0,1\}^n \rightarrow \{0,1\}$ can be computed by a circuit with 2^n gates.

Also, almost all F's need $\geq \frac{2^n}{n}$ gates.

The functions we care about are special and in some cases we only need polynomial in n gates
E.g. shortest path

Probabilistic Computation · Add a FLIP gate

$$\underset{\text{no input}}{\boxed{\text{FLIP}}} \text{—} \begin{cases} 0 & \text{w.p. } 1/2 \\ 1 & \text{w.p. } 1/2 \end{cases} \quad \text{random output}$$

A probabilistic circuit C "computes" $F : \{0,1\}^n \rightarrow \{0,1\}$ if

$$\forall \text{ inputs } x, \quad \underset{\text{flips}}{\mathbb{P}}\left[C(x) \neq F(x) \right] \leq \text{small} \quad (\text{say } 1\%) \leftarrow$$ We can reduce the error by repeating

We strongly believe that probabilistic computation does not give exponential savings

NEXT TIME Quantum Circuits