Lecture 16: Tues March 21

Guest Lecture by Tom Wong

Last time we addressed a few conceptual points about quantum computing. Today we cover two more:

3. What is the role of interference in quantum computing?

Since quantum amplitudes can cancel out (unlike classical probabilities), we can construct scenarios where the amplitudes for incorrect solutions cancel out with each other, leaving only amplitudes representing the correct solution.

4. What is the role of entanglement in quantum computing?

We can write a pure state of n qubits as the product,

$$(\alpha_1|0\rangle + \beta_1|1\rangle) \otimes (\alpha_2|0\rangle + \beta_2|1\rangle) \otimes \ldots \otimes (\alpha_n|0\rangle + \beta_n|1\rangle)$$

This requires keeping track of only 2n amplitudes, so we can store it efficiently. But an entangled state of n qubits $\sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$ requires 2^n amplitudes, which quickly becomes intractable.

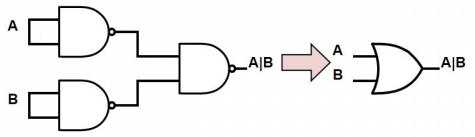
With 300 qubits you'd need more atoms than are available in the universe.

So arbitrarily entangled states can't be simulated well classically. This task requires a quantum computer.

In order to start talking about the construction of quantum computers through quantum gates, we need to cover…

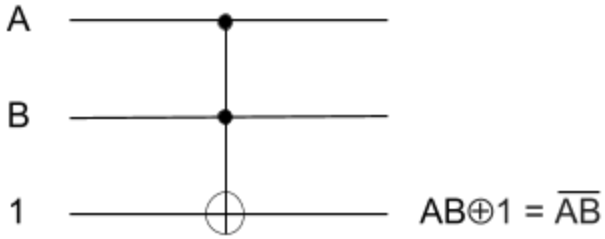
Universal Gate Sets

Classically, you're probably familiar with all of the standard gates (AND, OR, NOT, NAND, etc). A (classical) universal gate set is a grouping of such gates from which you can construct all of the others.



For example, NAND by itself is universal. The diagram on the right shows how you'd construct an OR gate out of NANDs, and the others can all be worked out too.

Similarly, the **Toffoli Gate** universal. The Toffoli Gate, also known as the controlled-controlled-NOT is a three bit gate where if A and B are 1, you flip C. To show that Toffoli is universal, we construct a NAND gate out of one (in the diagram on the right). If a Toffoli can create a universal gate set it must, too, be universal.



By making input C always 1, the output of C is 0 only if both A and B were 1 and the bit was flipped. If A or B or both were 0, the bit is not flipped and the gate returns 1. Thus, we've got a NAND gate.

It's worth noting that since Toffoli is reversible—given the outputs A, B, AB⊕C we can recover inputs A, B, C—which means we can use it as a quantum gate too. Thus you can see that a quantum computer can do anything a classical computer can do, because one can implement a classical universal gate set.

Now let's talk about *Quantum* Universal Gate Sets:

which we define as a set of gates that allows you to approximate any unitary to any desired precision. An important theorem on the subject is the…

Sololvay-Kitaev Theorem

which says that with any universal gate set, we can approximate a unitary on n qubits to precision using $O(2^n \text{ polylog}(1/\varepsilon))$ gates.

There are plenty of ways that a gate set can fail to be universal.

1. Your gate set doesn't create interference/superposition

   Ex: {cNOT} can only flip between $|0\rangle$ and $|1\rangle$. It can maintain superposition, but it can't create any.

2. Your gate set has superposition, but is missing entanglement

   Ex: {Hadamard} can create superposition, but it should be obvious that the gate can't create entanglement since it only acts on one qubit.

3. Your gate set only has real gates

   Ex: {cNOT, Hadamard} is getting closer, but neither can reach positions with non-real values.

4. Your gate set is "only a stabilizer set"

   We're not going to go in depth with the concept of stabilizer sets. What's important to know is that a set like $\{\text{cNOT, Hadamard, } P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}\}$ fails because it's efficiently simulated by a classical computer (by the **Gottesman-Knill Theorem**). This property prevents it from getting speedups relative to a classical computer.

Are there any other ways to fail to be universal?

That's an open question!

So what *is* universal?

It turns out that if you replace Hadamard in the above example with almost anything else, the set becomes universal.

So $\{\text{cNOT}, R_{\pi/8} = \begin{pmatrix} \cos(\pi/8) & -\sin(\pi/8) \\ \sin(\pi/8) & \cos(\pi/8) \end{pmatrix}, P\}$ is universal.

Also, {Toffoli, Hadamard, P} is universal.

Any two unitaries picked at random will likely be universal.

Quantum Complexity

There's two major ways we look at the complexity of quantum algorithms

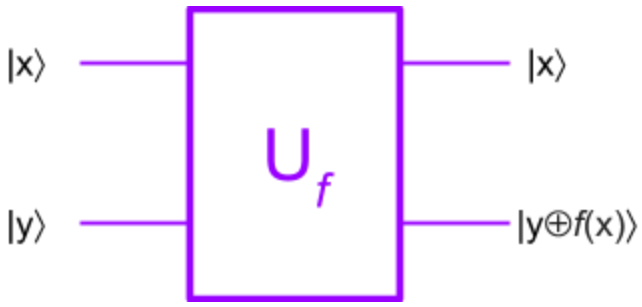
The circuit complexity of a unitary is the size of the smallest circuit that implements it. We like unitaries with polynomial circuit complexity. This can be difficult to find: it's a gate-set-dependent measure. At best we usually only get upper/lower bounds, so instead we tend to use…

Query complexity, the number of calls the algorithm makes to an oracle (or black box function). The idea is that your oracle takes a bit and outputs a bit $f : \{0,1\} \rightarrow \{0,1\}$. Classically you'd have a bit go $x \rightarrow f(x)$, but we replace this with quantum states $|x\rangle \rightarrow |f(x)\rangle$.

Or rather, we *want* to replace it with quantum states, but we run into a bit of trouble because such a transformation is not unitary. What we have to do instead is use an extra answer/target qubit.

So we give the black box two qubits: x, which stays the same, and y, which receives the answer.

$$|x,y\rangle \rightarrow |x, y \oplus f(x)\rangle$$



A lot of times we ignore the answer qubit by moving the phases around. So let's say we prepare the answer qubit as |-⟩.

We start with $|x,-\rangle = \frac{1}{\sqrt{2}}(|x,0\rangle - |x,1\rangle)$

Applying U_f gets us $\frac{1}{\sqrt{2}}(|x,0 \oplus f(x)\rangle - |x,1 \oplus f(x)\rangle)$

Which equals $\{ \frac{1}{\sqrt{2}}(|x,0\rangle - |x,1\rangle)$ if $f(x) = 0$

$\{ \frac{1}{\sqrt{2}}(|x,1\rangle - |x,0\rangle)$ $f(x) = 1$

Which we can rewrite as $(-1)^{f(x)} |x,-\rangle$

This lets us avoid dealing with the answer qubit and just use the "phase oracle".

$$|x\rangle \rightarrow (-1)^{f(x)}|x\rangle$$

Now we're finally ready to tackle our first quantum algorithm.

Deutsch's Algorithm

We're given two unknown bits, b_0 and b_1.

Given an index x ϵ {0,1}, our oracle returns the bit. i.e. $f(x) = b_x$

What we want to know is, "What is the parity of these bits?"

Parity is whether the bits have different values, so $b_0 + b_1 \pmod 2$ or $b_0 \oplus b_1$

Classically, this would take two queries since we need to know both bits.

Quantumly, Deutsch's Algorithm can do it in one.

Start with a qubit at |0⟩, Hadamard it, then do a query which applies a phase change to each part depending on the value of the function.

$$|0\rangle \longrightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \longrightarrow \frac{1}{\sqrt{2}}((-1)^{f(x)}|0\rangle + (-1)^{f(x)}|1\rangle)$$

We can substitute in the bits.

$$= \frac{1}{\sqrt{2}}((-1)^{b_0}|0\rangle + (-1)^{b_1}|1\rangle)$$

Then drag out b_0.

$= \frac{1}{\sqrt{2}}(-1)^{b_0} \ (|0\rangle + (-1)^{b_1 - b_0}|1\rangle)$

So now if we have $b_0 = b_1$ we get $(-1)^{b_0} \ \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$

$b_0 \neq b_1$ $\quad\quad$ $(-1)^{b_0} \ \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$

We can ignore the phase out front since global phase doesn't affect measurement, and then Hadamard again to get our quantum states back in the $|0\rangle$, $|1\rangle$ basis.

Now the $b_0 = b_1$ case becomes $|0\rangle$

and the $b_0 \neq b_1$ case becomes $|1\rangle$

The complete quantum circuit is drawn to the right. If the bits had parity we measure 1, if they don't we measure 0.