QI Lecture 5

Universality and Deutsch-Jozsa

Universal Quantum Gates

Theorem (Construction of Arbitrary Unitary Operators)**.** *Any unitary operation on n qubits can be expressed as the product of single qubit operators and CNOT gates.*

We can define the error when we apply the unitary operator V in place of U as

$$E(U,V) \equiv \max_{|\psi\rangle} \|(U-V)|\psi\rangle\|$$

If P_U is the probability of a given measurement assuming U is applied and P_V is the probability of the same outcome assuming V is applied, we find that

$$|P_U - P_V| \leq 2E(U,V)$$

In other words, if we can *approximate* a given unitary operation to a high enough precision, the results at the other end will have the same statistics. This is important because if we only have access to a finite set of gate operations, then arbitrary unitary operations are out of reach. Instead we need to know if we can usefully approximate any unitary operation using only combinations of operations from our finite set.

The "**standard set of universal gates**" is the Hadamard (H), phase (S), controlled not (CNOT), and $\frac{\pi}{8}$ (T)[1] gates:

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \qquad CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}$$

A set of gates is universal if it can be used to approximate any unitary operator to arbitrary precision. That doesn't mean that it's easy, merely possible. This particular set is actually more than we need, since $T^2 = S$.

[1] The "$\frac{\pi}{8}$ gate" name is historical. T can also be written $T = e^{i\frac{\pi}{8}}\begin{bmatrix} e^{-i\frac{\pi}{8}} & 0 \\ 0 & e^{i\frac{\pi}{8}} \end{bmatrix}$.

Proving Universality

Using only T and H it's possible to come arbitrarily close to any rotation on the Bloch sphere. The central idea is that as long as you can rotate by *approximately* any amount around two different axes, $\hat{m}$ and $\hat{n}$, then you can approximate any unitary operation on one qubit, up to an ignorable global phase, using $U = R_{\hat{n}}(\beta)R_{\hat{m}}(\gamma)R_{\hat{n}}(\delta)$. The idea is similar (but slightly generalized) to the $Z-Y$ decomposition for a single qubit.

By direct calculation (and ignoring global phase)

$$T = R_z\left(\frac{\pi}{4}\right) = \cos\left(\frac{\pi}{8}\right)I - \sin\left(\frac{\pi}{8}\right)Z$$

$$HTH = R_x\left(\frac{\pi}{4}\right) = \cos\left(\frac{\pi}{8}\right)I - \sin\left(\frac{\pi}{8}\right)X$$

doing one then the other, and remembering that $ZX = iY$[2]

$$\begin{aligned}
THTH &= R_z\left(\tfrac{\pi}{4}\right)R_x\left(\tfrac{\pi}{4}\right) \\
&= \left[\cos\left(\tfrac{\pi}{8}\right)I - i\sin\left(\tfrac{\pi}{8}\right)Z\right]\left[\cos\left(\tfrac{\pi}{8}\right)I - i\sin\left(\tfrac{\pi}{8}\right)X\right] \\
&= \cos^2\left(\tfrac{\pi}{8}\right)I - i\sin\left(\tfrac{\pi}{8}\right)\cos\left(\tfrac{\pi}{8}\right)X - i\sin\left(\tfrac{\pi}{8}\right)\cos\left(\tfrac{\pi}{8}\right)Z - \sin^2\left(\tfrac{\pi}{8}\right)ZX \\
&= \cos^2\left(\tfrac{\pi}{8}\right)I - i\sin\left(\tfrac{\pi}{8}\right)\cos\left(\tfrac{\pi}{8}\right)X - i\sin\left(\tfrac{\pi}{8}\right)\cos\left(\tfrac{\pi}{8}\right)Z - i\sin^2\left(\tfrac{\pi}{8}\right)Y \\
&= \cos^2\left(\tfrac{\pi}{8}\right)I - i\sin\left(\tfrac{\pi}{8}\right)\left[\cos\left(\tfrac{\pi}{8}\right)X + \sin\left(\tfrac{\pi}{8}\right)Y + \cos\left(\tfrac{\pi}{8}\right)Z\right]
\end{aligned}$$

Comparing this to

$$R_{\hat{n}}(\xi) = \cos\left(\frac{\xi}{2}\right)I - i\sin\left(\frac{\xi}{2}\right)(n_xX + n_yY + n_zZ)$$

we see that we're looking at a rotation around $\hat{n}$ by an angle ξ where

$$\hat{n} = \frac{1}{\sqrt{\cos^2\left(\frac{\pi}{8}\right)+1}}\left(\cos\left(\frac{\pi}{8}\right), \sin\left(\frac{\pi}{8}\right), \cos\left(\frac{\pi}{8}\right)\right) \qquad \cos\left(\frac{\xi}{2}\right) = \cos^2\left(\frac{\pi}{8}\right)$$

With a calculator, we can calculate these two

$$\hat{n} \approx (0.678598, 0.281085, 0.678598) \qquad \xi \approx 0.174443 \times 2\pi$$

You should feel comfortable with the fact that all of these are nice, messy, irrational numbers. If that angle were a rational fraction of 2π, $\xi = \frac{p}{q}2\pi$, then after q rotations by ξ

[2] This is a specific case of a more general property: $\sigma_a\sigma_b = \delta_{ab}I + i\epsilon_{abc}\sigma_c$. ϵ_{abc} is the "Levi-Civita symbol" which is the "sign of the permutation of $\{abc\}$". $\epsilon_{123} = \epsilon_{231} = \epsilon_{312} = 1$, $\epsilon_{321} = \epsilon_{213} = \epsilon_{132} = -1$, and doubled indices produce zeros (e.g., $\epsilon_{112} = 0$).

you'd have made p complete rotations, landing on every multiple of $\frac{2\pi}{q}$. But for irrational multiples of 2π, the rotation never lands on the same angle twice. Instead it produces a set of angles that are "**dense**" on $[0, 2\pi)$, meaning that for any angle $\theta \in [0, 2\pi)$ and any error ϵ, there's a k such that $|\theta - k\xi| < \epsilon$. In other words, for sufficiently large k, we can rotate by any angle around $\hat{n}$ with arbitrarily small error.

If you rotate by ξ a total of N times, then you produce N points on the interval $[0, 2\pi)$ and by the Pigeon Hole Principle[3] there must be some j and k such that $|k\xi - j\xi| \leq \frac{2\pi}{N}$. Therefore, $(k-j)\xi$ is an angle smaller than $\frac{2\pi}{N}$ (which is arbitrarily small) and we now have access to multiples of this arbitrarily small angle, $\ell(k-j)\xi$.

Finally, by direct calculation, we find that

$$HR_{\hat{n}}(\xi)H = R_{\hat{m}}(\xi)$$

where $\hat{m} \approx (0.678598, -0.281085, 0.678598)$. Notably, $\hat{m} \neq \hat{n}$ and we still have access to arbitrarily fine control over the angle of rotation. With these two rotations in hand, we can construct any arbitrary rotation on the Bloch sphere, and thus any arbitrary unitary operator (ignoring global phase).

How Many Gates?

You would be right to suspect, after the derivation above, that the number of universal gates you might need for your quantum computer, for even the simplest operations, is large. But how large? The Solvay-Kitaev theorem answers this question.

- Define $SU(2)$ to be the set of 2×2 unitary operators with determinant one[4] (this is another way to say "don't worry about the global phase").
- Define $\mathcal{G}$ be a finite set of unitary operators in $SU(2)$, such that the inverse of every element in $\mathcal{G}$ is also included in $\mathcal{G}$.
- Define $\langle \mathcal{G} \rangle$ as the set of all finite products of elements from $\mathcal{G}$. For example, if $a, b, c, d \in \mathcal{G}$, then $acdca^{-1}b^3c \in \langle \mathcal{G} \rangle$. Elements in $\langle \mathcal{G} \rangle$ are called "words" and the number of operations in a word is the "length", ℓ, of that word.

Theorem (Solvay-Kitaev)**.** *Assume that $\langle \mathcal{G} \rangle$ is "dense" in the set of $SU(2)$, in the sense that for any $U \in SU(2)$, there exists a $V \in \langle \mathcal{G} \rangle$ such that $E(U, V)$ is arbitrarily small.*

Then for any (very small) $\epsilon \in \mathbb{R}^+$, every $U \in SU(2)$ there is a "word" $V \in \langle \mathcal{G} \rangle$ with a length no longer than $\ell = O\left(\log^c\left(\frac{1}{\epsilon}\right)\right)$, where $c \approx 4$, such that $E(U, V) < \epsilon$.

[3] "If you have more pigeons than pigeon holes, then at least some pigeons will have a roommate." Why pigeons are the titular example, or why they're being kept in holes instead of nests or cubbies, is not worth worrying about.

[4] "$SU(n)$" means "special unitary $n \times n$ matrices", where "special" means "determinant 1".

Solvay-Kitaev promises that any single qubit unitary operation can be simulated with arbitrarily small error using only gates from a universal set of gates and that decreasing the error, ϵ, doesn't force us to use too many more gates.

Deutsch's Problem

Deutsch's Problem was among the first quantum algorithm discovered that is known to be exponentially faster than any known algorithm and is almost always the first algorithm taught. Being famously pointless, it's more proof-of-concept than useful. We'll look at a small version first.

There is a function $f : \{0, 1\} \to \{0, 1\}$ that is a "black box".[5] The only way to figure out what f does is to feed it an input and see what it produces as output. The question that the algorithm seeks to answer is "Is f balanced or constant?",[6] where "balanced" means that f is 0 exactly as often as it is 1.

In order to make this determination classically, we'd need to evaluate f twice. If $f(0) = f(1)$, it's constant and if $f(0) \neq f(1)$, it's balanced. We'll find that Deutsch-Jozsa can do this in one evaluation.

We enact $f(x)$ through a unitary operation, U_f,

$$U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$$

where $\oplus$ indicates addition modulo 2.[7] Notice that unlike the Hadamard operations, U_f is being applied to both qubits. Stepping through the circuit in figure 2 we can see how the algorithm works.

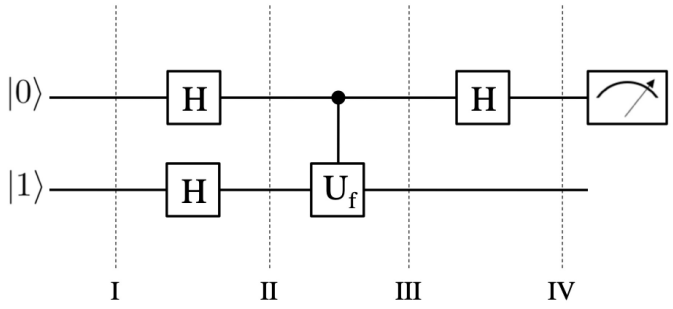

Figure 1: The circuit for the Deutsch-Jozsa Algorithm.

The initial state is

[5] This notation means that both the domain and range of f are $\{0, 1\}$.

[6] Notice that we're not answering "What is f?", a genuinely useful question.

[7] $0 \oplus 0 = 0$, $0 \oplus 1 = 1$, and $1 \oplus 1 = 0$.

$$I: \qquad |0\rangle|1\rangle$$

The Hadamard operations put both qubits into superpositions. $H\otimes H|0\rangle|1\rangle = \left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$

$$II: \qquad \frac{1}{2}\left(|0\rangle+|1\rangle\right)\left(|0\rangle-|1\rangle\right)$$

A clever thing happens when the second input to U_f is $|-\rangle$.
If $f(x)=0$, then $U_f|x\rangle\left(|0\rangle-|1\rangle\right) = |x\rangle\left(|0\oplus 0\rangle-|1\oplus 0\rangle\right) = |x\rangle\left(|0\rangle-|1\rangle\right)$.
If $f(x)=1$, then $U_f|x\rangle\left(|0\rangle-|1\rangle\right) = |x\rangle\left(|0\oplus 1\rangle-|1\oplus 0\rangle\right) = |x\rangle\left(|1\rangle-|0\rangle\right)$.
We can sum this up with a single equation:

$$U_f|x\rangle\left(|0\rangle-|1\rangle\right) = (-1)^{f(x)}|x\rangle\left(|0\rangle-|1\rangle\right)$$

With this trick in hand, we can cleanly apply U_f:

$$\begin{aligned}
& U_f\tfrac{1}{2}\left(|0\rangle+|1\rangle\right)\left(|0\rangle-|1\rangle\right) \\
= \;& \tfrac{1}{2}\left[U_f|0\rangle\left(|0\rangle-|1\rangle\right)+U_f|1\rangle\left(|0\rangle-|1\rangle\right)\right] \\
= \;& \tfrac{1}{2}\left[(-1)^{f(0)}|0\rangle\left(|0\rangle-|1\rangle\right)+(-1)^{f(1)}|1\rangle\left(|0\rangle-|1\rangle\right)\right] \\
= \;& \tfrac{1}{2}\left[(-1)^{f(0)}|0\rangle+(-1)^{f(1)}|1\rangle\right]\left(|0\rangle-|1\rangle\right)
\end{aligned}$$

$$III: \qquad \frac{1}{2}\left[(-1)^{f(0)}|0\rangle+(-1)^{f(1)}|1\rangle\right]\left(|0\rangle-|1\rangle\right)$$

Finally, a second Hadamard operation is applied to the first qubit, $H\otimes I$. Again, be very careful to note the where the operators are applied and which symbols are taking about the first and second qubits.

$$\begin{aligned}
& H\otimes I\tfrac{1}{2}\left[(-1)^{f(0)}|0\rangle+(-1)^{f(1)}|1\rangle\right]\left(|0\rangle-|1\rangle\right) \\
= \;& \tfrac{1}{2}\left[(-1)^{f(0)}H|0\rangle+(-1)^{f(1)}H|1\rangle\right]\left(|0\rangle-|1\rangle\right) \\
= \;& \tfrac{1}{2}\left[(-1)^{f(0)}\left(\tfrac{|0\rangle+|1\rangle}{\sqrt{2}}\right)+(-1)^{f(1)}\left(\tfrac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\right]\left(|0\rangle-|1\rangle\right) \\
= \;& \tfrac{1}{2\sqrt{2}}\left[\left((-1)^{f(0)}+(-1)^{f(1)}\right)|0\rangle+\left((-1)^{f(0)}-(-1)^{f(1)}\right)|1\rangle\right]\left(|0\rangle-|1\rangle\right) \\
= \;& \frac{\left((-1)^{f(0)}+(-1)^{f(1)}\right)|0\rangle+\left((-1)^{f(0)}-(-1)^{f(1)}\right)|1\rangle}{2}\left(\tfrac{|0\rangle-|1\rangle}{\sqrt{2}}\right)
\end{aligned}$$

So the final state is

$$IV: \qquad \frac{\left((-1)^{f(0)}+(-1)^{f(1)}\right)|0\rangle+\left((-1)^{f(0)}-(-1)^{f(1)}\right)|1\rangle}{2}\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

We don't need to bother measuring the second qubit, because we already know what the result will be. It's $|-\rangle$. The state of the first qubit depends on $f(x)$:

If f is constant, then the first qubit is $\pm|0\rangle$ and if f is balanced, then the first qubit is $\pm|1\rangle$. Therefore, with only one evaluation of f, we can determine with certainty if f is balanced or not.

The Deutsch-Jozsa Algorithm

Answering a silly question (balanced vs. constant) by evaluating a function once instead of twice isn't terribly impressive. So we'll look at a very similar algorithm where we try to answer the same question for $f : \{0,1\}^N \to \{0,1\}$. That is, f outputs a 0 or 1 for every N bit string (every number from 0 to $2^N - 1$) and is either constant or balanced, in the sense that half the outputs are 0 and half are 1. The circuit should look familiar.

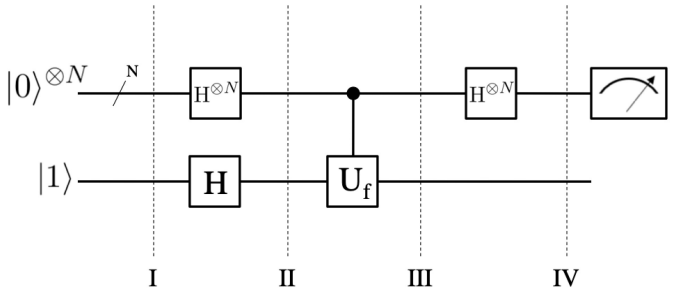

Figure 2: In this version, the top "wire" is actually N qubits.

We introduce some new notation to handle N-qubit strings.

$$|0\rangle^{\otimes N} = \underbrace{|0\rangle|0\rangle\ldots|0\rangle}_{N} = |00\ldots0\rangle \qquad\qquad H^{\otimes N} = \underbrace{H \otimes H \otimes \ldots \otimes H}_{N}$$

The initial state is:

$$I: \qquad |0\rangle^{\otimes N}|1\rangle$$

After the first bank of Hadamard gates the state is:

$$\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)^{\otimes N}\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

That state on the left is an equal superposition of every N-qubit string: $|0\ldots00\rangle$, $|0\ldots01\rangle$, $|0\ldots10\rangle$, up to $|1\ldots11\rangle$. If we think of these strings as the binary representation of a number, we can save a little room by just writing that number. For example, $|6\rangle = |0\ldots0110\rangle$.

In other words, $|x\rangle = |x_1x_2\ldots x_N\rangle$, where x_j is the jth binary digit of x.

$$II: \qquad \frac{1}{2^{N/2}} \sum_{x=0}^{2^N-1} |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

This is *much* easier than writing the sum over every binary digit individually.

$$\frac{1}{2^{N/2}} \sum_{x=0}^{2^N-1} |x\rangle = \frac{1}{2^{N/2}} \sum_{x_1=0}^{1} \sum_{x_2=0}^{1} \cdots \sum_{x_N=0}^{1} |x_1 x_2 \ldots x_N\rangle$$

Just like in the simpler algorithm above, $U_f|x\rangle\left(|0\rangle - |1\rangle\right) = (-1)^{f(x)}|x\rangle\left(|0\rangle - |1\rangle\right)$. All that's important here is that $f(x) \in \{0,1\}$ and the fact that x now runs from 0 to $2^N - 1$ doesn't change that. Applying the (now much more complicated) U_f:

$$\begin{aligned} & U_f \tfrac{1}{2^{N/2}} \textstyle\sum_{x=0}^{2^N-1} |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \\ = \; & \tfrac{1}{2^{N/2}} \textstyle\sum_{x=0}^{2^N-1} U_f|x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \\ = \; & \tfrac{1}{2^{N/2}} \textstyle\sum_{x=0}^{2^N-1} (-1)^{f(x)}|x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \end{aligned}$$

So the state of the system after the function has been applied is

$$III: \qquad \frac{1}{2^{N/2}} \sum_{x=0}^{2^N-1} (-1)^{f(x)}|x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

For the last bank of Hadamard gates, it helps to write the affect of H on each qubit like this:

$$H|j\rangle = \frac{1}{\sqrt{2}} \sum_{k=0}^{1} (-1)^{xy}|k\rangle$$

Take a minute to explicitly write this out for $j = 0, 1$ (it's a lot simpler than it looks). Applying the same operation to every binary digit we get

$$H|x\rangle = \frac{1}{2^{N/2}} \sum_{y=0}^{2^N-1} (-1)^{x \cdot y}|y\rangle$$

where $x \cdot y \equiv x_1 y_1 + x_2 y_2 + \ldots + x_N y_N$. Again, seriously, take a minute to untangle what just happened there.

$$
\begin{aligned}
& \left(H^{\otimes N} \otimes I\right) \tfrac{1}{2^{N/2}} \textstyle\sum_{x=0}^{2^N-1} (-1)^{f(x)} |x\rangle \left(\tfrac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \\
= & \tfrac{1}{2^{N/2}} \textstyle\sum_{x=0}^{2^N-1} (-1)^{f(x)} H^{\otimes N} |x\rangle \left(\tfrac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \\
= & \tfrac{1}{2^{N/2}} \textstyle\sum_{x=0}^{2^N-1} (-1)^{f(x)} \left(\tfrac{1}{2^{N/2}} \textstyle\sum_{y=0}^{2^N-1} (-1)^{x \cdot y} |y\rangle\right) \left(\tfrac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \\
= & \tfrac{1}{2^N} \textstyle\sum_{x=0}^{2^N-1} \sum_{y=0}^{2^N-1} (-1)^{x \cdot y + f(x)} |y\rangle \left(\tfrac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \\
= & \tfrac{1}{2^N} \textstyle\sum_{y=0}^{2^N-1} \left(\sum_{x=0}^{2^N-1} (-1)^{x \cdot y + f(x)}\right) |y\rangle \left(\tfrac{|0\rangle - |1\rangle}{\sqrt{2}}\right)
\end{aligned}
$$

The final state is therefore

$$
IV: \qquad \frac{1}{2^N} \sum_{y=0}^{2^N-1} \left(\sum_{x=0}^{2^N-1} (-1)^{x \cdot y + f(x)} \right) |y\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)
$$

Once again, the last qubit is always in the state $|-\rangle$. The state of the first N qubits is $\frac{1}{2^N} \sum_{y=0}^{2^N-1} \left(\sum_{x=0}^{2^N-1} (-1)^{x \cdot y + f(x)}\right) |y\rangle$. Now consider the amplitude of just $|0\rangle$.

$$
\frac{1}{2^N} \left(\sum_{x=0}^{2^N-1} (-1)^{x \cdot 0 + f(x)} \right) |0\rangle = \frac{1}{2^N} \left(\sum_{x=0}^{2^N-1} (-1)^{f(x)} \right) |0\rangle
$$

If f is constant, then the total of the sum is $\pm 2^N$. Therefore the amplitude of $|0\rangle$ is ± 1, meaning that $P(0) = |\pm 1|^2 = 1$. We can stop here, because the probability of any other result must be zero.[8]

On the other hand, if f is balanced, then half of the terms in the sum will be -1, the other half will be 1, and the total will be zero. Therefore, $P(0) = 0$.

So we have a measurement that tell us whether f is constant or balanced after a single evaluation of f:

$$
|y\rangle = |0\rangle \Rightarrow \text{constant}
$$

$$
|y\rangle \neq |0\rangle \Rightarrow \text{balanced}
$$

[8] If that bothers you, you'll love the homework!

Exercises

1) Breaking stuff to see how it works.

For Deutsch's Problem, what changes if the second qubit is initially equal to $|0\rangle$ instead of $|1\rangle$? To figure this out, step through the entire circuit.

2) Really? The probability of any other result is zero?

The state of the first N qubits at the end of the Deutsch-Jozsa Algorithm is

$$\frac{1}{2^N}\sum_{y=0}^{2^N-1}\left(\sum_{x=0}^{2^N-1}(-1)^{x\cdot y+f(x)}\right)|y\rangle$$

If f is constant, the probability of measuring $|0\rangle$ is 1, and therefore the probability of measuring any other result must be zero. That "therefore" is a little unsatisfying.

By explicit calculation, show that when f is constant the probability amplitude for any non-zero $|y\rangle$ is zero.

Hint: First try $|y\rangle = |0\ldots01\rangle$, to get a sense of what's going on, then prove it more generally. My goal is to trick you into staring at the "binary number notation" until it makes sense.