Notes 8.370/18.435 Fall 2021

Lecture 9 Prof. Peter Shor

For the last couple of lectures, we've been talking about classical gates and classical circuits. Today, we'll start talking about quantum gates.

Recall that we can build any Boolean circuit out of AND, OR, and NOT gates. These are one- and two-bit gates. We will show that the same thing is true for quantum circuits——we can build quantum circuits out of one-qubit gates and a small set of two-qubit gates. A one-qubit gate is a 2×2 unitary matrix, and a two-qubit gate is a 4×4 unitary matrix.

We've discussed a nubmer of specific quantum gates before. For example, we've seen the Pauli matrices

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \sigma_z = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$$

These satisfy the commutation relations:

$$\sigma_x^2 = \sigma_y^2 = \sigma_z^2 = I$$
$$\sigma_z \sigma_x = -\sigma_x \sigma_z = i\sigma_y$$

as well as the relations obtained from this one by cyclic permutations of x, y, and z.

$$\sigma_x \sigma_y = -\sigma_y \sigma_x = i\sigma_z$$
$$\sigma_y \sigma_z = -\sigma_z \sigma_y = i\sigma_x.$$

These are often abbreviated by X, Y, and Z, especially in quantum circuit diagrams.

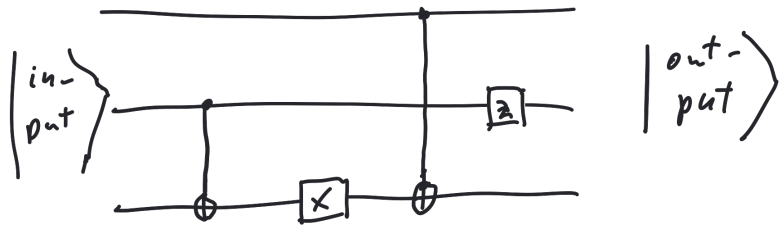
We also saw the rotations of the Bloch sphere areound the x-, y-, and $z-$axes by an angle θ:

$$R_x(\theta) = \begin{pmatrix} \cos\theta & -i\sin\theta \\ -i\sin\theta & \cos\theta \end{pmatrix},$$
$$R_y(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix},$$
$$R_z(\theta) = \begin{pmatrix} e^{-i\theta} & 0 \\ 0 & e^{i\theta} \end{pmatrix}$$

We have $R_z(\pi) = \begin{pmatrix} -i & 0 \\ 0 & i \end{pmatrix} = -i\sigma_z$ and similarly $R_x(\pi) = -i\sigma_x$ and $R_y(\pi) = -i\sigma_y$. Recall that a global phase does not affect the actual quantum state, so $R_w(\pi)$ is essentially the same transformation as σ_w, where w is x, y, or z.

Finally, there is the Hadamard gate, $H - \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, which rotates the Bloch sphere around the point $(\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}})$, switching the x- and z=axes. So we have $HXH = Z$ and $HYH = -Y$.

A quantum circuit is a set of quantum gates connecting qubits. We represent each qubit as a horizontal wire, and we draw the gates connecting two qubits as a line between two wires. Time proceeds from left to right. For example, a simple quantum circuit is given in Fig. 1.



Quantum Circuit

Here, there are two CNOT gates (represented by a dot on the control qubit and an XOR symbol $\oplus$ on the target qubit, a σ_x gate (represented by X and a σ_z gate (represented by a Z. The CNOT gate, for example, is the 4×4 matrix

$$\mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

But the state space of three qubits is an 8-dimensional state space. How do we apply a 4×4 matrix to it? We take the tensor product. For example, in the state space of three qubits, a CNOT with qubit 2 as the control and qubit 1 as the target is:

$$\mathrm{CNOT}_{2\to 3} = \begin{pmatrix} 1 & 0 & 0 & 0 & & & & \\ 0 & 1 & 0 & 0 & & & & \\ 0 & 0 & 0 & 1 & & & & \\ 0 & 0 & 1 & 0 & & & & \\ & & & & 1 & 0 & 0 & 0 \\ & & & & 0 & 1 & 0 & 0 \\ & & & & 0 & 0 & 0 & 1 \\ & & & & 0 & 0 & 1 & 0 \end{pmatrix},$$

and a CNOT with qubit 2 as the control and qubit 3 as the target would be:

$$\mathrm{CNOT}_{1\to 2} = \begin{pmatrix} 1 & 0 & & & & & & \\ 0 & 1 & & & & & & \\ & & 1 & 0 & & & & \\ & & 0 & 1 & & & & \\ & & & & & & 1 & 0 \\ & & & & & & 0 & 1 \\ & & & & 1 & 0 & & \\ & & & & 0 & 1 & & \end{pmatrix}.$$

But how do we handle a CNOT from qubit 1 to qubit 3? What we need to do is tensor $\mathrm{CNOT}_{1\to 3}$ with I_2. This isn't representable in the standard Kronecker form that you've seen for $A \otimes B$, because the qubits the CNOT acts on aren't consecutive. So what do we do? One thing we can do is write down the formula for $\mathrm{CNOT}_{1\to 2}$ and then apply a change of basis matrix that interchanges qubits 2 and 3. This gives the formula:

$$\mathrm{CNOT}_{1\to 3} = \big(I_1 \otimes \mathrm{SWAP}_{2,3}\big)\big(\mathrm{CNOT}_{1\to 2} \otimes I_3\big)\big(I_1 \otimes \mathrm{SWAP}_{2,3}\big)$$

where

$$\mathrm{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

This gives the correct answer, as does

$$\mathrm{CNOT}_{1\to 3} = \big(\mathrm{SWAP}_{1,2} \otimes I_3\big)\big(I_1 \otimes \mathrm{CNOT}_{2\to 3}\big)\big(\mathrm{SWAP}_{1,2} \otimes I_3\big)$$

but it's rather cumbersome.

An easier way to do it is to break the CNOT into 2×2 blocks and tensor the identity on qubit 2 with each of the 2×2 blocks . This is shown below:

$$\mathrm{CNOT} = \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{array}\right) \qquad \mathrm{CNOT}_{1\to 3} = \left(\begin{array}{cccc|cccc} 1 & 0 & & & 0 & 0 & & \\ 0 & 1 & & & 0 & 0 & & \\ & & 1 & 0 & & & 0 & 0 \\ & & 0 & 1 & & & 0 & 0 \\ \hline 0 & 0 & & & 0 & 1 & & \\ 0 & 0 & & & 1 & 0 & & \\ & & 0 & 0 & & & 0 & 1 \\ & & 0 & 0 & & & 1 & 0 \end{array}\right)$$

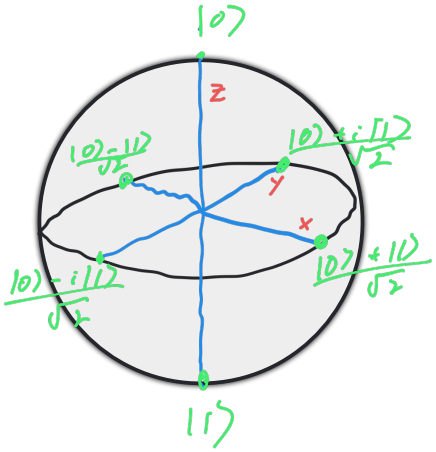
Since we aren't going to be representing gates on n qubits as $2^n \times 2^n$ matrices (for obvious reasons), we won't be seeing this kind of representation much. But it's good to know what's going on conceptually.

Recall that we showed that AND, OR, and NOT can produce any Boolean function, and TOFFOLI gates and NOT Gates can produce any reversible Boolean function.

So we can ask: is there a finite set of quantum gates that will produce any unitary transformation?

The answer to this question is "no", for a fairly straightforward reason: there are uncountably many unitary transformations, and only a countable number of quantum circuits using a finite gate set. The resolution to this is that we can approximate any unitary transformation arbitrarily well using a finite gate set, and that we don't even need an unreasonable number of gates to do so. However, this is the Solovay-Kitaev theorem, and the proof of this theorem is rather involved. (It's in an appendix to the textbook, if you want to look at it.) So what we will show is that any unitary transformation can be produced using CNOT gates and one-qubit gates (in particular, a limited set of one-qubit gates).

The first thing we will do is recall the Bloch sphere. Recall that $R_y(\theta)$ was a



Bloch Sphere

rotation of θ around the y axis, and similarly for $R_z(\theta)$. Further recall that we can use $R_y(\theta)$ and $R_z(\theta)$, to perform an arbitrary rotation of the Bloch sphere by applying

$$R_z(\theta_3)R_y(\theta_2)R_z(\theta_1).$$

This isn't quite an arbitrary unitary because $R_z(\theta)$ and $R_y(\theta)$ have determinant 1, so we only get determinant-1 unitaries this way. However, we can multilply any unitary by a global phase to get a determinant-1 unitary, and global phases have no effect on a quantum state, so this gives us an arbitrary quantum transformation on a one-qubit state.

Our next goal will be to show that we can get controlled $R_z(\theta)$ and controlled $R_y(\theta)$ gates. These are

$$\text{C}{-}R_y(\theta) = \begin{pmatrix} I_2 & 0 \\ 0 & R_y(\theta) \end{pmatrix}, \qquad \text{C}{-}R_z(\theta) = \begin{pmatrix} I_2 & 0 \\ 0 & R_z(\theta) \end{pmatrix}.$$

where I_2 is the 2×2 identity matrix.

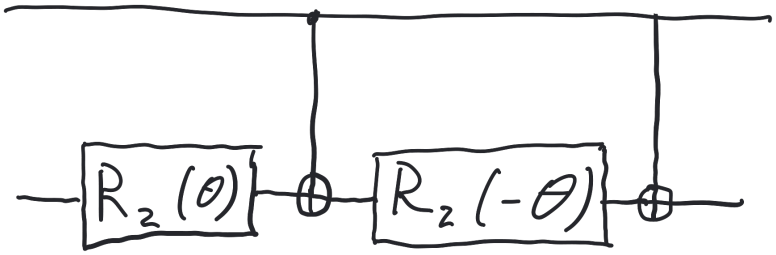
We first show how to make controlled rotations C-$R_z(\theta)$ and C-$R_y(\theta)$. First, however, let's do a straightforward matrix calculation:

$$\sigma_x R_z(\theta)\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} e^{i\theta/2} & 0 \\ 0 & e^{-i\theta/2} \end{pmatrix} = R_z(-\theta)$$

Similarly,

$$\sigma_x R_y(\theta)\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \cos\theta/2 & -\sin\theta/2 \\ \sin\theta/2 & \cos\theta/2 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} \cos\theta/2 & \sin\theta/2 \\ -\sin\theta/2 & \cos\theta/2 \end{pmatrix} = R_y(-\theta)$$

Now, consider the following circuit. If the first qubit is a $|\,0\rangle$, then we have $R_z(-\theta)R_z(\theta)$



applied to the second qubit, and these two operations cancel each other out. Now, suppose the first qubit is a $|\,1\rangle$. Recall that a CNOT is also a controlled σ_x. So if the first qubit is $|\,1\rangle$, we have $R_z(\theta)$ applied to the second qubit, followed by $\sigma_x R_z(-\theta)\sigma_x$. This is $R_z(\theta)$, which when multiplied by the first $R_z(\theta)$ gives $R_z(2\theta)$. We thus have a circuit for a C-$R_z(2\theta)$.

The same circuit with $R_z(\theta)$ replaced by $R_y(\theta)$ gives the C-$R_y(2\theta)$.

We now show how to make an arbitrary controlled unitary on two qubits. A C-U gate applies the identity to the target qubit if the control qubit is $|\,0\rangle$ and applies U to the target qubit if the control qubit is $|\,1\rangle$. That is, it is the matrix

$$\begin{pmatrix} & & 0 & 0 \\ & I_2 & 0 & 0 \\ 0 & 0 & & \\ 0 & 0 & & U \end{pmatrix}$$

where the control qubit is the first qubit and the target qubit is the second one, and I_2 is the 2×2 identity matrix.

Now, how can we apply C-U for an arbitrary one-qubit unitary U. Recall that we can express U as $R_z(\theta_3)R_y(\theta_2)R_z(\theta_1)$ for appropriately chosen θ_1, θ_2, and θ_3. Similarly

$$\text{C-}U = \text{C-}R_z(\theta_3)\,\text{C-}R_y(\theta_2)\,\text{C-}R_z(\theta_1).$$

This gives the iimplementation of C-U for an arbitrary one-qubit U.