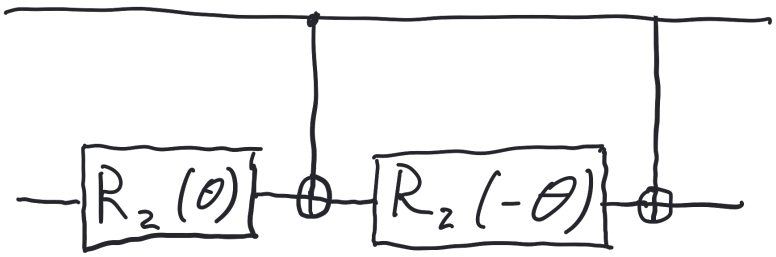
Notes 8.370/18.435 Fall 2022

Lecture 10 Prof. Peter Shor

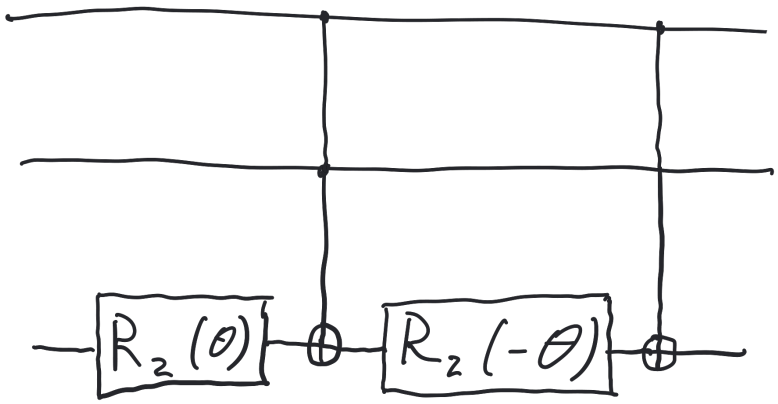
Today, we continue our discussion of gates permitting universal quantum computation. Recall that last time we showed how to make an arbitrary controlled R_y or R_z rotation, using the following circuit.



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. 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-$R_y(2\theta)$.

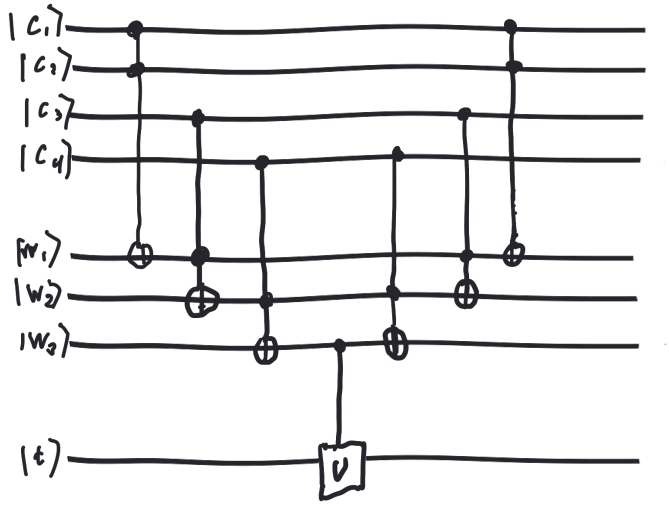
The next thing we want to do is show how to do a doubly controlled U gate, where we apply a U if both the first two qubits are $|1\rangle$ and an identify otherwise. If we can do a Toffoli gate, we can use the same technique we used for contructing for a controlled NOT to do this:



We will not show how to make a Toffoli gate in this lecture; this will be a homework assignment.

We can construct a doubly controlled unitary CC-U using the same technique we used to construct a C-U: Find angles α, β, and γ so that $U = R_z(\gamma)R_y(\beta)R_z(\alpha)$; we then have $\mathrm{CC}{-}U = \mathrm{CC}{-}R_z(\gamma)\mathrm{CC}{-}R_y(\beta)\mathrm{CC}{-}R_z(\alpha)$.

We now will construct a C^k-U, a circuit that applies a U gate to the target qubit if the k control qubits are in the state $|\,1\rangle$, and applies the identity otherwise. This is accomplished by the following circuit:



Here, we use $k-1$ extra work qubits which start in the state $|\,0\rangle$. If all the control bits are $|\,1\rangle$, then the first $k-1$ Toffoli gates set all the work qubits to $|\,1\rangle$, and a controlled-U gate applied to the last work bit applies a U to the target qubit. Otherwise, the last work qubit remains in the state $|\,0\rangle$, and an identity is applied to the target qubit. The

last $k-1$ Toffoli gates set all the work bits back to $|0\rangle$; this step will be necessary in quantum computation to make the interference work properly.

We now have constructed enough gates to show how to make an arbitrary unitary from $R_z(\theta)$, $R_y(\theta)$, CNOT, and Toffoli gates. For this, we will need what the textbook calls a two-level gate. This is a gate on n qubits which is diagonal except for two rows and two columns, which contain a 2×2 unitary matrix in them. An example of such a gate is:

$$\begin{pmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & a & & b & & & \\ & & & 1 & & & & \\ & & c & & d & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{pmatrix},$$

where $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is a small unitary matrix.

To show that we can construct an arbitrary matirx U from two-level matrices, we need to show that we can find two-level matrices T_i so that

$$T_1^\dagger T_2^\dagger T_3^\dagger \dots T_m^\dagger = U.$$

We will work backwards, showing that there are two-level matrices $T_i^\dagger$ such that

$$U T_m T_{m-1} \dots T_1 = I.$$

Since the conjugate transpose of T_i (which is also the inverse of T_i, because T_i is unitary) is a two-level matrix, this will show that U is a product of two-level matrices.

When you multiply U by a two-level matrix, it will only affect two columns and two rows of U, since a $r \times r$ two-level matrix is an identity in $r-2$ of its rows and columns. Now, for an arbitrary $r \times r$ matrix U, there is a two-level matrix T_1 (which only affects the first and last columns) that sets the lower left entry of UT, $UT(r,1)$, to 0. Now that UT_1 is 0, we can find another two-level matrix T_2 that sets the $(r-1,1)$ entry UT_1T_2 to 0. If we continue setting entries of $UT_1T_2 \dots T_k$ to 0, and work from the lower right up, when we apply a new two-level matrix, we will never undo one of the zeroes, because we have already set al the entries to the left and below the entry we are working on to 0. Thus, doing this, we can set all the entries below the diagonal to 0. Because the length of all rows and columns in a unitary matrix are 1, this means that all the entries above the diagonal are also 0, and that the diagonal contains unit complex numbers. These can all be set to 1 by applying another sequence of two-level matrices.

So how can we construct an arbitrary two-level matrix? We have already constructed one class of two-level matrices. If we have n qubits, then the multiply con-

trolled gate with $n-1$ control qubits and one target qubit looks like:

$$\begin{pmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & a & b \\ & & & & & & c & d \end{pmatrix},$$

We only need to show that we can move the unitary in the last two rows and columns to an arbitrary pair of rows and columns. We can do this with an appropriate sequence of NOTs and CNOTS, which we will describe next.

Actually, for ease of exposition, it's better to put the unitary matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ at the top left of the matrix rather than the bottom right. We can do that by applying a NOT gate to all but the last qubit, giving this matrix.

$$M_{0,1} = \begin{pmatrix} a & b & & & & & & \\ c & d & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{pmatrix},$$

Now, suppose we want to construct the matrix $M_{2,4}$ where the unitary is in rows and columns 2 and 4, rather than 0 and 1.

$$M_{2,4} = \begin{pmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & a & & b & & & \\ & & & 1 & & & & \\ & & c & & d & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{pmatrix},$$

What we can do is construct a permutation matrix P that takes the basis vector $|\, e_2\rangle = |\, 010\rangle$ to the basis vector $|\, e_0\rangle = |\, 000\rangle$ and the basis vector $|\, e_4\rangle = |\, 100\rangle$ to basis vector $|\, e_1\rangle = |\, 001\rangle$, and then use P to construct $M_{2,4}$ by applying $P^{-1}M_{0,1}P$. The matrix $P^{-1}M_{0,1}P$ only affects the $|\, 010\rangle$ and the $|\, 100\rangle$ coordinates of a vector v, because the P^1 undoes everything that $M_{0,1}$ does not affect.

So how do we move two arbitrary basis vectors to $|\, 000\rangle$ and $|\, 001\rangle$? We will give a proof by example; once you've seen how it works for two specific basis vectors, extending the procedure to an arbitrary two basis vectors is straightforward. What we

wnat to do is to move the basis vectors $|\,010\rangle$ and $|\,100\rangle$ to the basis vectors $|\,000\rangle$ and $|\,001\rangle$? We use NOTS and CNOTs for this. First, we can move the coordinate $|\,010\rangle$ to $|\,000\rangle$ by applying a NOT gate to the second qubit. Applying this NOT gate takes $|\,100\rangle$ to $|\,110\rangle$. We next take $|\,110\rangle$ to $|\,001\rangle$ by applying CNOT gates. These CNOT gates do not affect $|\,000\rangle$, because it contains all 0s. We can do this as follows:

$$\begin{aligned}
\mathrm{CNOT}_{1,3}\,|\,110\rangle &= |\,111\rangle \\
\mathrm{CNOT}_{1,2}\,|\,111\rangle &= |\,101\rangle \\
\mathrm{CNOT}_{3,1}\,|\,101\rangle &= |\,001\rangle\,.
\end{aligned}$$

we thus have constructed P, and this lets us produce $M_{2,4}$. So we are done.