Lecture 9

Universal quantum gates

Single qubit + CNOT gates

Single qubit and CNOT gates together can be used to implement an arbitrary two-level unitary operation on the state space of n qubits.

Suppose $U$ is a two-level unitary matrix which acts non-trivially on the space spanned by the computational basis states $|s\rangle$ and $|t\rangle$, where $s = s_1 \ldots s_n$ and $t = t_1 \ldots t_n$. Let $\tilde{U}$ be the non-trivial $2 \times 2$ unitary submatrix of $U$.

$$U = \begin{bmatrix} a & 0 & 0 & 0 & 0 & 0 & c \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ b & 0 & 0 & 0 & 0 & 0 & d \end{bmatrix} \quad \tilde{U} = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$$

Goal: to construct a circuit implementing $U$ from single qubit and CNOT gates.

Use Gray codes: A Gray code connecting binary numbers $s$ and $t$ is a sequence of binary numbers, starting with $s$ and concluding with $t$, such that adjacent members of the list differ in one bit.

Example: $s=101001$, $t=110011$.

Gray code $g_1 \ldots g_m$, $g_1 = s$, $g_m = t$:

$$
\begin{array}{cccccccc}
g_1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
g_2 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
g_3 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
g_4 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
\end{array}
$$
Basic idea of the quantum circuit implementing U

- Swap the states $|g_1\rangle$ and $|g_2\rangle$.
- Swap the states $|g_2\rangle$ and $|g_3\rangle$ and continue until we swap $|g_{m-2}\rangle$ and $|g_{m-1}\rangle$.
- Apply a controlled-$\tilde{U}$ operation, with the target qubit located at the single bit where $g_{m-1}$ and $g_m$ differ.
- Undo the swap operations: swap $|g_{m-2}\rangle$ and $|g_{m-1}\rangle$, and so on until $|g_2\rangle$ and $|g_1\rangle$ are swapped.

Example: consider the U gate on the previous page.

Question for the class: what logical operation does gate U perform?

\[
\begin{array}{c|c}
|000\rangle & U \\
|001\rangle & \rightarrow \\
|010\rangle & \rightarrow \\
|011\rangle & \rightarrow \\
|100\rangle & \rightarrow \\
|101\rangle & \rightarrow \\
|110\rangle & \rightarrow \\
|111\rangle & \rightarrow \\
\end{array}
\]

Since the U acts non-trivially only on states $|000\rangle$ and $|111\rangle$, the Gray codes is

\[
\begin{array}{c|c|c|c}
A & B & C \\
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 1 & 1 \\
1 & 1 & 1 \\
\end{array}
\]
The circuit to implement the gate U is:

\[ A \]
\[ B \]
\[ C \]

\[ \tilde{U} \]

**Class exercise: work out what this circuit does gate by gate.**

Hint: you only need to keep track on four states listed in the Gray code as the remaining four states are not affected at all.

<table>
<thead>
<tr>
<th>ABC</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>000⟩</td>
<td></td>
<td>001⟩</td>
</tr>
<tr>
<td></td>
<td>001⟩</td>
<td></td>
<td>000⟩</td>
</tr>
<tr>
<td></td>
<td>011⟩</td>
<td></td>
<td>011⟩</td>
</tr>
<tr>
<td></td>
<td>111⟩</td>
<td></td>
<td>111⟩</td>
</tr>
</tbody>
</table>

\[ \text{swap |1000⟩ and |1001⟩} \]
\[ \text{swap |1001⟩ and |1011⟩} \]
\[ \text{apply } \tilde{U} \text{ if both } B \text{ and } A \text{ are |11⟩} \]

\[ 4 \]
\[ a|001⟩ + c|111⟩ \]
\[ a|000⟩ + c|111⟩ \]
\[ |000⟩ \]
\[ |011⟩ \]
\[ |011⟩ \]
\[ 5 \]
\[ b|001⟩ + d|111⟩ \]
\[ b|000⟩ + d|111⟩ \]
\[ \text{unswap the states back} \]
**Measurement**

**Principle of deferred measurement:** Measurements can always be moved from an intermediate stage of a quantum circuit to the end of the circuit.

**Principle of implicit measurement:** Without loss of generality, any unterminated quantum wires (qubits which are not yet measured) at the end of the quantum circuit may be assumed to be measured.

In order for a measurement to be reversible, it must reveal no information about the quantum system being measured!

**Summary of the quantum circuit model of computation**

- **Classical resources.** For example, many schemes for quantum error-correction involve classical computations to maximize efficiency.

- **A suitable state space.** For a quantum circuit operating on n qubits the state space is $2^n$ dimensional Hilbert space. Computational basis: $|x_1 \ldots x_n\rangle$, where $x_i = 0, 1$.

- **Ability to prepare states in the computational basis.** (Any computational basis state can be prepared in at most n steps.)

- **Ability to perform quantum gates.** The set of Hadamard + phase + CNOT + $\pi/8$ gates is universal.

- **Ability to perform measurements in the computational basis.**
Quantum algorithms

Note: The complexity class NP
Problems in NP: yes answer can be easily verified with the aid of an appropriate witness.
Classical factoring algorithm

Modular arithmetic

Arithmetic of remainders (p.626 of the textbook)

\[ 18 = 2 \cdot 7 + 4 \equiv 18 \pmod{7} \]

For any positive integers \( x \) and \( n \)

\[ x = kn + r \leftarrow \text{unique representation} \]

\( k \) is a non-negative integer and \( 0 \leq r \leq n-1 \).

Modular arithmetic = ordinary arithmetic in which we pay attention to remainders only. Notation \((\text{mod } n)\) is used to indicate that we are working in modular arithmetic.

Class exercise: Prove that \( 2=5=8=11 \pmod{3} \)

\[
\begin{align*}
x &= kn + r \\
2 &= 0 \cdot 3 + 2 \\
5 &= 1 \cdot 3 + 2 \\
8 &= 2 \cdot 3 + 2 \\
11 &= 3 \cdot 3 + 2
\end{align*}
\]

Class exercise: calculate \( 7^n \pmod{15} , \ n = 1, 2, 3, 4 \)

\[
\begin{align*}
h=1 & \quad 7^1 = 7 = 15 \cdot 0 + 7 \Rightarrow 7^1 \pmod{15} = 7 \\
h=2 & \quad 7^2 = 49 = 15 \cdot 3 + 4 \Rightarrow 7^2 \pmod{15} = 4 \\
h=3 & \quad 7^3 = 343 = 15 \cdot 22 + 13 \Rightarrow 7^3 \pmod{15} = 13 \\
h=4 & \quad 7^4 = 2401 = 15 \cdot 160 + 1 \Rightarrow 7^4 \pmod{15} = 1
\end{align*}
\]
Classical factoring algorithm: How to factor 15?

(1) Pick a number less than 15 (for example 7).

(2) Calculate \(7^n \mod 15\)

<table>
<thead>
<tr>
<th>(n)</th>
<th>(7^n)</th>
<th>(15 \times m)</th>
<th>(7^n \mod 15)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>49</td>
<td>45</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>343</td>
<td>330</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>2401</td>
<td>2400</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>16807</td>
<td>16800</td>
<td>7</td>
</tr>
</tbody>
</table>

\[ R = 4 \]

\text{pattern} \\
7 4 13 1 \\
is repeated

**Question: is it a coincidence that pattern repeated itself after we got 1?**

If you think that it is not, look for proof.

\[
7^5 = 7^4 \cdot 7 = (2400 + 1) \cdot 7 = (15 \cdot 160 + 1) \cdot 7 \\
= 15 \cdot 160 \cdot 7 + 7 = 7 \mod 15 = 7 \quad \text{and so on...}
\]

The point of calculating \(7^n \mod 15\) was to find period \(R\). This is the step that is hard for classical computers for large \(n\).

3) Calculate greatest common divisor \(\gcd\{7^{\text{R/2}} \pm 1, 15\}\)

\[
7^{\text{R/2}} - 1 = 48; \quad \gcd\{48, 15\} = 3 \\
7^{\text{R/2}} + 1 = 50; \quad \gcd\{50, 15\} = 5
\]

There is an efficient classical algorithm for finding \(\gcd\). See pages 627-629 of the textbook for description [Euclid's algorithm].
Cryptography (p.582-583 of the textbook)

Cryptography: art of enabling two parties to communicate in private

**Private key cryptography**

Alice sends a message to Bob. To communicate in private, they must have encoding key to encrypt the message and decoding key to decrypt the message.

**Example:** Vernam cipher (or a one time pad)

Original message: Q U A N T U M

Encryption key (random string): G Q Y R W A D

Encrypted message: W L Y F Q U P

Message is send over public channels.

Received message: W L Y F Q U P

Decryption key: G Q Y R W A D

Decrypted message: Q U A N T U M

**Great feature:** as long as the key string is a secret and in used ones, it is secure.

**Problem:** Secure key distribution. Vernam cipher is secure as long as the number of key bits is at least as large as the size of the message encoded as keys can not be reused! Key bits must be delivered in advance, guarded, and then destroyed.

**Solution:** quantum key distribution!
**Public key distribution**

**RSA cryptosystems**

Basic idea of public key cryptosystems (much like a mailbox)

Alice sets a mailbox. Public key is available to the public. Only Alice can get the mail out of the mailbox. Alice has secret key.

**Result:** anyone in the world can communicate with Alice privately.

**Note:** there are two distinct keys; a public key and a private key (which only Alice has).

**How does it work?**

Suppose Bob wishes to send private message to Alice.

1. Alice generates two keys, a public key (P) and a secret (private) key (S).
2. Bob gets a copy of a public key (P).
3. Bob encrypts the message using P. Encryption stage is very difficult to reverse! Like a trap door for the mail: if you put in your mail you can not get it out. Bob sends the encrypted message.
4. Alice uses a secret key to decrypt the message.

**Problem:** There is no known scheme which is proven to be secure. It is just widely believed that it is!

In order to discuss how RSA encryption actually works, we need more modular arithmetic.