Quantum Computing
Lecture 4

Models of Quantum Computation

Quantum Circuits

A quantum circuit is a sequence of unitary operations and measurements on an \( n \)-qubit state.

![Quantum Circuit Diagram]

Note: each \( U_i \) is described by a \( 2^n \times 2^n \) matrix.

Algorithms

A quantum algorithm specifies, for each \( n \), a sequence

\[ O_n = O_1 \ldots O_k \]

of \( n \)-qubit operations.

The map \( n \rightarrow O_n \) must be computable.

\textit{i.e. the individual circuits must be generated from a common pattern.}

Model of Computation

As a model of computation, this is parasitic on classical models.

\textit{what is computable is not independently determined}

Purely quantum models can be defined. We will see more on this in Lecture 8.

What computations can be performed in the model as defined?

\textit{What functions can be computed?}
\textit{What decision problems are decidable?}

Can all such computations be performed with some fixed set of unitary operations?
Simulating Boolean Gates

Could we find a quantum circuit to simulate a classical $\text{And}$ gate?

\[
\begin{array}{c}
a \\
\text{And} \\
b \end{array}
\rightarrow a \land b
\]

This would require $\text{And} : |00\rangle \mapsto |0x\rangle$, $|01\rangle \mapsto |0y\rangle$

$|10\rangle \mapsto |0z\rangle$, $|11\rangle \mapsto |1w\rangle$

There is no unitary operation of this form.

Unitary operations are reversible. No information can be lost in the process.

Computing a Function

If $f : \{0,1\}^n \rightarrow \{0,1\}^m$ is a Boolean function, the map

$|x\rangle \mapsto |f(x)\rangle$

may not be unitary.

We will, instead seek to implement

$|x\rangle \otimes |0\rangle \mapsto |x\rangle \otimes |f(x)\rangle$

Exercise: Describe a unitary operation that implements the Boolean $\text{And}$ in this sense.

One-Qubit Gates

We have already seen the Pauli Gates:

\[
X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},
Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix},
Z = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
\]

Another useful one-qubit gate is the Hadamard gate:

\[
H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}
\]

Gates on a Multi-Qubit State

When we draw a circuit with a one-qubit gate, this must be read as a unitary operation on the entire state.

\[
\begin{array}{c}
U \\
\end{array}
\]

$U \otimes I$

This does not change measurement outcomes on the second qubit.
**Controlled Not**

The *Controlled Not* is a 2-qubit gate:

\[
\begin{align*}
|a\rangle & \quad \text{The controlled not flips the second qubit if the first qubit is } |1\rangle \\
|b\rangle & \quad \text{and leaves it unchanged if it's } |0\rangle
\end{align*}
\]

\[
C = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0
\end{bmatrix}
\]

**Controlled U**

More generally, we can define, for any single qubit operation \(U\), the *Controlled U* gate:

\[
\begin{align*}
\text{Particularly useful is the controlled-Z gate:}
\end{align*}
\]

**Toffoli Gate**

The *Toffoli Gate* is a 3-qubit gate. It has a classical counterpart which can be used to simulate standard Boolean operations.

\[
\begin{align*}
|a_1\rangle & \quad \text{A permutation matrix is a unitary matrix where all entries are } 0 \text{ or } 1. \\
|a_2\rangle & \quad \text{Any } 2^n \times 2^n \text{ permutation matrix can be implemented using only Toffoli gates.}
\end{align*}
\]

**Classical Reversible Computation**

A Boolean function \(f : \{0,1\}^n \rightarrow \{0,1\}^n\) is reversible if it’s described by a \(2^n \times 2^n\) permutation matrix.

For any function \(g : \{0,1\}^n \rightarrow \{0,1\}^m\), there is a reversible function \(g' : \{0,1\}^{m+n} \rightarrow \{0,1\}^{m+n}\) with

\[
g'(x,0) = (x, g(x)).
\]

Toffoli gates are universal for reversible computation.

The Toffoli gate cannot be implemented using 2-bit classical gates.
Quantum Toffoli Gate

The Toffoli gate can be implemented using 2-qubit quantum gates.

\[
\begin{align*}
H & \quad Q & \quad Q & \quad Q' & \quad Q & \quad P \\
\end{align*}
\]

where, \( P = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \) and \( Q = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \).

Universal Set of Gates

Fact: Any unitary operation on \( n \) qubits can be implemented by a sequence of 2-qubit operations.

Fact: Any unitary operation can be implemented by a combination of C-NOTs and single qubit operations.

Fact: Any unitary operation can be approximated to any required degree of accuracy using only C-NOTs, \( H \), \( P \) and \( Q \).

These can serve as our finite set of gates for quantum computation.

Deutsch-Josza Problem

Given a function \( f : \{0,1\} \to \{0,1\} \), determine whether \( f \) is constant or balanced.

Classically, this requires \( two \) calls to the function \( f \).

But, if we are given the quantum black box:

\[
\begin{align*}
|a\rangle & \quad U_f & \quad |a\rangle \\
|b\rangle & \quad b \oplus f(a) \\
\end{align*}
\]

One use of the box suffices

Deutsch-Josza Algorithm

\[
\begin{align*}
|0\rangle & \quad H & \quad U_f & \quad H \\
\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) & \quad & \quad & \\
\end{align*}
\]

\( U_f \) with input \( |x\rangle \) and \( |0\rangle - |1\rangle \) is just a phase shift.

It changes phase by \(-1)^{f(x)} \).

When \( |x\rangle = H|0\rangle \), this gives \(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle \).

Final result is \([(-1)^{f(0)} + (-1)^{f(1)}]|0\rangle + [(-1)^{f(0)} - (-1)^{f(1)}]|1\rangle \)

which is \( |0\rangle \) if \( f \) is constant and \( |1\rangle \) is balanced.