Introduction to Quantum Information and Computing - Notes

Aayush Acharya, Arnav Negi, Kriti Gupta, Manav Shah, Muhammed Shamil K, Shiven Sinha, Shrikara A, Swayam Agrawal, Vineeth Bhat, Yash Adivarekar

February 2023

Contents

1 Introduction

- Quantum computing - ”Natural generalization of computing”.
- Quantum computers are computers that obey quantum physics.

1.1 Why quantum computing?

- Extended Church Turing Thesis (ECT): Any algorithmic process can be efficiently simulated by a probabilistic Turing machine.
- David Deutsch:-Is there a physical model of computation that violates ECT?
- Computation devices built using the principles of quantum physics can offer a stronger version to their thesis.
- Do quantum computers that obey quantum mechanics violate ECT?
- Feynman asks if we can simulate quantum physics on a classical computer.
- Number of variables to keep track of is exponential in the size of the quantum system. For example, for a 100 electron system,2^{100} bits are needed in comparison to 100 qubits.

2 Quantum computing in the circuit model (Postulates of Quantum mechanics in action)

2.1 State Preparation

- Prepare the quantum computer in a given initial state
  $|\psi_0\rangle$=$|0\rangle^{\otimes n}$
- $|0\rangle$=$\begin{pmatrix}1\\0\end{pmatrix}$
  $|0\rangle \otimes |0\rangle$=$|0\rangle^{\otimes 2}$=$|0\rangle|0\rangle$=$|00\rangle$

2.2 Evolution

- Schrodinger’s equation :

  $i\frac{d|\psi\rangle}{dt}$=$H|\psi\rangle$

  $|\psi(t)\rangle$=$e^{-iHt}|\psi(0)\rangle$
- H: Hamiltonian observable for energy
- $u_t = e^{-iHt}$

- The initial state $|\psi_0\rangle$ evolves based on a series of unitary operators
  $|\psi_f\rangle$=$u_t u_{t-1} u_{t-2} ... u_2 u_1 |\psi_0\rangle$
  where each u_i is a quantum gate

2.3 Measurement

- Measure the final state in the computational basis: M=$|j\rangle\langle j|$ where $j \in \{0,1\}^n$

- Observe $|f\rangle$ with probability p such that:
  $p = |\langle f|\psi_f\rangle|^2$

3 Classical Logic Gates

Any boolean function $f : \{0,1\}^n \to \{0,1\}$ can be written as a propositional formula of n variables. A universal gate is a gate which can implement any Boolean function f without need to use any other gate type.

3.1 Reversibility of Logic Gates

Logical reversibility means that the output can be computed from the input, and vice versa. Reversible functions are bijective. This means that reversible gates (and circuits, i.e. compositions of multiple gates) have the same number of inputs as outputs.

A NOT gate is logically reversible because it can be undone.

The exclusive or XOR gate is irreversible because its two inputs cannot be unambiguously reconstructed from its single output, or alternatively, because information erasure is not reversible.

A motivation for implementing reversible computing is that they offer to improve the computational energy efficiency of computers.

4 Shannon Entropy

Shannon entropy is a measure of the amount of uncertainty or information contained in a quantum state. It is defined as: $\sum_i -p_i \log(p_i)$

Example: Consider a qubit with initial probability distrubution $P : \{0.5, 0.5\}$.We reset this qubit and we are asked to calculate the change in entropy.

The initial entropy = $-0.5 * log(0.5) - 0.5 * log(0.5) = \log(2)$.

The final entropy is 0 as the qubit is reset and thus no probabilistic measure.

Hence $\Delta S = -log(2)$

5 Landauer's Principle

Landauer's principle is a fundamental principle in quantum information theory that relates the amount of irreversible information loss in a computation to the physical process of erasing information. In essence, the principle states that erasing information must necessarily generate entropy, and that the minimum amount of energy required to perform an erasure is proportional to the entropy generated.

It can be formally stated as : Any irreversible operation that erases one bit of information must necessarily generate at least $k_B ln 2$ units of entropy, where k_B is the Boltzmann constant. This means that the minimum energy required to perform an erasure is given by:

$E = k_B T \ln 2$ where T is the temperature of the system in which the erasure is performed.

6 Clausius Inequality

The Clausius inequality is a fundamental principle of thermodynamics that places a limit on the amount of work that can be extracted from a system during a thermodynamic process.Mathematically:

$\Delta Q \geq k_B T \Delta S$

Here ΔS is the decrease in entropy of the system or increase in entropy of environment. ΔQ denotes the heat lost by the system to the environment.

7 Demonstrating the relationship between Measurement, Information and Thermodynamics

The Szilard Engine and Maxwell's Demon are two thought experiments in quantum mechanics that highlight the role of information and measurement in thermodynamics.

7.1 Szilard Engine

The Szilard Engine consists of a single particle confined in a box with a partition in the middle. The box is initially in a state of thermal equilibrium with its environment. The partition is moved to one side of the box, and the position of the particle is measured. If the particle is found on one side of the box, then the partition is locked into place, creating a smaller box on that side. This process can be repeated until the particle is confined to a small box, and the work that can be extracted from the process can be used to do useful work.

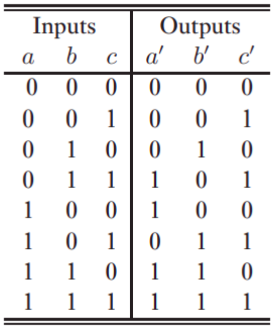
This engine thus demonstrates that measurement and information are intimately connected to thermodynamics. In particular, the act of measuring the position of the particle changes the state of the system, and this change can be used to extract work from the system. This suggests that information is a form of physical resource that can be converted into work.

8 Reversible Logic Gates

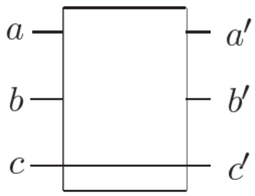
8.1 Fredkin Gate

Also know as Controlled SWAP Gate

The Fredkin gate has three input bits and three output bits, which we refer to as a, b, c and a’, b’, c’, respectively. The bit c is a control bit, whose value is not changed by the action of the Fredkin gate, that is, c’ = c. The reason c is called the control bit is because it controls what happens to the other two bits, a and b. If c is set to 0 then a and b are left alone, a’ = a, b’ = b. If c is set to 1, a and b are swapped, a’ = b, b’ = a. It is easy to see that the Fredkin gate is reversible, because given the output a’, b, c’, we can determine the inputs a, b, c. In fact, to recover the original inputs a, b and c we need only apply another Fredkin gate to a’, b’, c

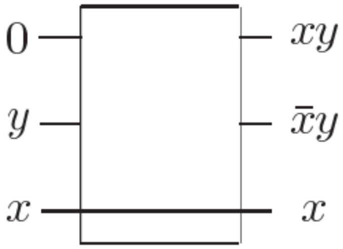


Inputs			Outputs		
a	b	c	a'	b'	c'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	1	1	0
1	1	1	1	1	1



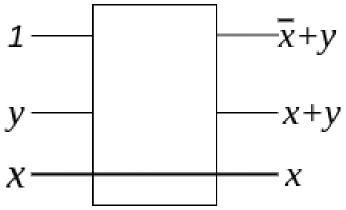
The Fredkin gate is not only reversible, it’s a universal logic gate as well.

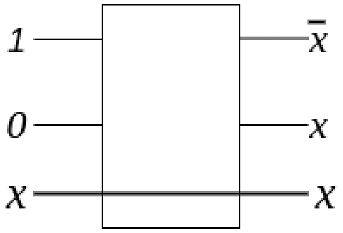
Reversible AND Gate (R-AND):



Reversible OR Gate (R-OR):

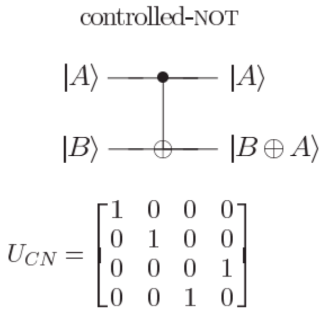
Reversible NOT Gate (R-NOT):





8.2 CNOT Gate

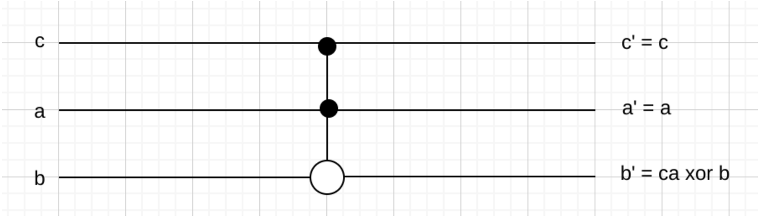
Controlled-NOT Gate: This gate has two input qubits, known as the control qubit and the target qubit.The action of the gate may be described as follows. If the control qubit is set to 0, then the target qubit is left alone. If the control qubit is set to 1, then the target qubit is flipped. The following image shows the circuit and the unitary matrix corresponding to the gate.


controlled-NOT
$|A\rangle$
$|A\rangle$
$|B\rangle$
$|B \oplus A\rangle$
$U_{CN} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

The reason the CNOT gate is not universal on its own is because it can only generate entangled states where the control and target qubits are either both 0 or both 1. This limits the set of transformations that can be generated.Reversibility is implied from the truth table.

9 Tofolli Gate

Also known as *CCNOT* Gate.



The $CCNOT$ gate preserves the control, c, and first input, a, and only performs an operation on the last input, b, to give the output

$$b' = ca \oplus b$$

which means that for b to be flipped, both a and c must be 1.

Further, this gate is universal and reversible.

10 Some problems

1. We receive some garbage bits that we don't require in the output of a quantum gate. For example, consider the operation carried out by a unitary u_f:
$$|x, 0, 0\rangle \xrightarrow{u_f} \Sigma_y \alpha |y\rangle |f(y)\rangle |g(y)\rangle$$
The term $|g(y)\rangle$ might not be needed and is termed as garbage or ancillary.
2. The garbage bits are entangled with the input and cannot be reset as it will make the operation irreversible.
3. We cannot measure the garbage register either as measurement will collapse the superposition to a particular state which isn't desirable.

This is solved using uncomputation.

11 Uncomputation

Say we wish to obtain the solution $|f(x) \oplus y\rangle$ given the inputs $|x\rangle$ and $|y\rangle$ as follows:

$$|x\rangle|0\rangle|0\rangle|y\rangle \xrightarrow{u_f} |x\rangle|f(x)\rangle|g(x)\rangle|y\rangle$$

Where the unitary u_f is applied on the first three qubits.

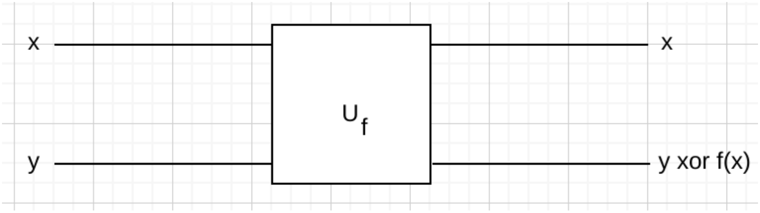
$$|x\rangle|f(x)\rangle|g(x)\rangle|y\rangle \xrightarrow{CNOT_{2,4}} |x\rangle|f(x)\rangle|g(x)\rangle|f(x) \oplus y\rangle$$

We then use the $CNOT$ gate on the 2nd and 4th qubits, i.e., $|f(x)\rangle$ and $|y\rangle$

$$|x\rangle|f(x)\rangle|g(x)\rangle|f(x) \oplus y\rangle \xrightarrow{u_f^{-1}} |x\rangle|0\rangle|0\rangle|f(x) \oplus y\rangle$$

We use the inverse of the unitary operation used earlier to uncompute the 2nd and 3rd bits.

All of these operations are represented by the short hand notation:



12 Quantum Circuits

We note that

1. Quantum gates are unitary operations on quantum states.
2. There exists a set of universal gates for quantum circuits usually denoted by G_O, which contains a small number of single as well as two qubit gates.

12.1 Single Qubit Gates

These include

1. Pauli matrices (which have been elucidated on earlier in the first half)
2. Hadamard matrix

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

   They transform from the basis $\{|0\rangle, |1\rangle\}$ to $\{|+\rangle, |-\rangle\}$.

$$H|0\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$
$$H|1\rangle = |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

3. Phase change matrices

$$R_\phi = \begin{bmatrix}1 & 0\\ 0 & e^{i\phi}\end{bmatrix}$$

   The phase change matrix adds a change of phase ϕ to the the state $|1\rangle$ as

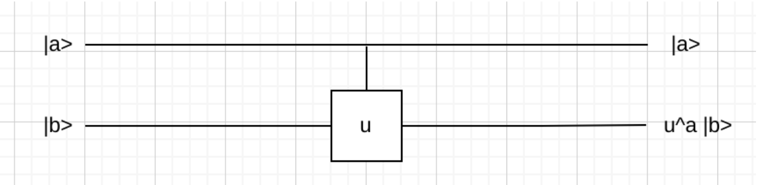
$$R_\phi(|\alpha|0\rangle + \beta|1\rangle) = |\alpha|0\rangle + e^{i\phi}\beta|1\rangle$$

   Some accepted ways of denoting common phase changes are $S = R_{\frac{\pi}{2}}$ and $T = R_{\frac{\pi}{4}}$.

12.2 Two Qubit Gates

12.2.1 Controlled two qubit gates

For example, if we wish to apply the unitary u only on the second state in $|ab\rangle$, i.e., as a *controlled* operation on a single qubit gate, we use the circuit:



The matrix representation of the operation is

$$\mathbf{I} \otimes u = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \multicolumn{2}{c}{\multirow{2}{*}{u}} \\ 0 & 0 & & \end{bmatrix}$$

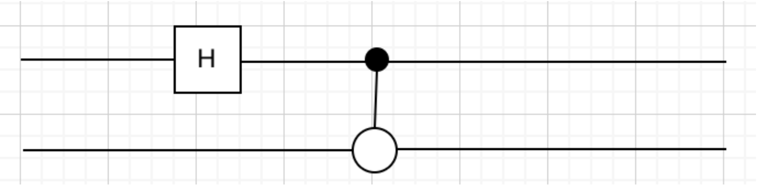
The operations carried out are:

$$|00\rangle \xrightarrow{\mathbf{I}\otimes u} |00\rangle$$
$$|01\rangle \xrightarrow{\mathbf{I}\otimes u} |01\rangle$$
$$|10\rangle \xrightarrow{\mathbf{I}\otimes u} |1\rangle(u|0\rangle)$$
$$|11\rangle \xrightarrow{\mathbf{I}\otimes u} |1\rangle(u|1\rangle)$$

For example, if we wish to model the $CNOT$ gate, we take $u = \sigma_x$ if the above circuit.

12.2.2 Another example of two qubit gates

Consider the following circuit



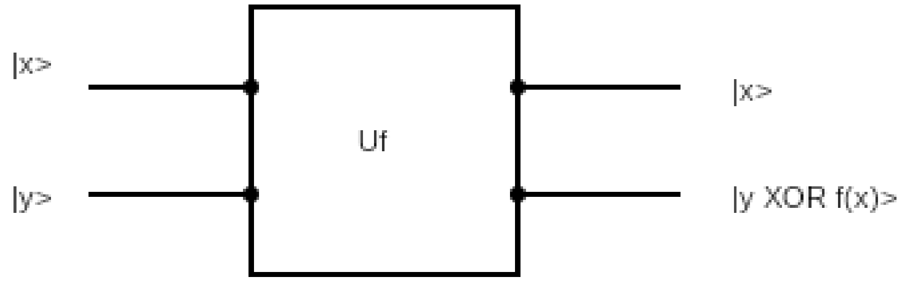
If we apply this circuit on the input $|00\rangle$, then we can model the output using the following calculations

$$|00\rangle \xrightarrow{H\otimes\mathbf{I}} |+0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$$

$$\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \xrightarrow{CNOT} \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

13 Phase Kickback Oracle

Consider the CNOT gate where U_f denotes the unitary operation CNOT.



The action of U_f is given by:

$$\begin{aligned} |x\rangle|y\rangle \xrightarrow{U_f} & |x\rangle|y\rangle \text{ if } f(x)=0 \\ & |x\rangle|\overline{y}\rangle \text{ if } f(x)=1 \end{aligned}$$

$|x\rangle, |y\rangle \in \{0,1\}$

Consider the case when $|y\rangle = |-\rangle$

$$\begin{aligned} |x\rangle|-\rangle \xrightarrow{U_f} & U_f\frac{(|x\rangle|0\rangle + |x\rangle|1\rangle)}{\sqrt{2}} \\ =& \frac{|x\rangle(|0\oplus f(x)\rangle + |1\oplus f(x)\rangle)}{\sqrt{2}} \\ =& |x\rangle|-\rangle \text{ if } f(x)=0 \\ & -|x\rangle|-\rangle \text{ if } f(x)=1 \\ =& (-1)^{f(x)}|x\rangle|-\rangle \end{aligned}$$

If $x \in \{0,1\}^n$ and before U_f, $H^{\otimes n}$ is applied on $|x\rangle$, then the output is:

$$\begin{aligned} & U_f\frac{1}{\sqrt{2^n}}\sum_{z\in\{0,1\}^n}|z\rangle|-\rangle && \textit{from the action of } H \\ =& \frac{1}{\sqrt{2^n}}\sum_{z\in\{0,1\}^n}(-1)^{f(x)}|z\rangle|-\rangle && \textit{from the action of } U_f \end{aligned}$$

Usually, the $|-\rangle$ in the second register is dropped when writing the phase kickback since it remains unchanged in output and is considered implicit when using the phase kickback oracle.

14 Deutsch Algorithm

14.1 The Problem

Suppose U_f is given as a black box for a boolean function $f : \{0,1\} \to \{0,1\}$, with the promise that either:

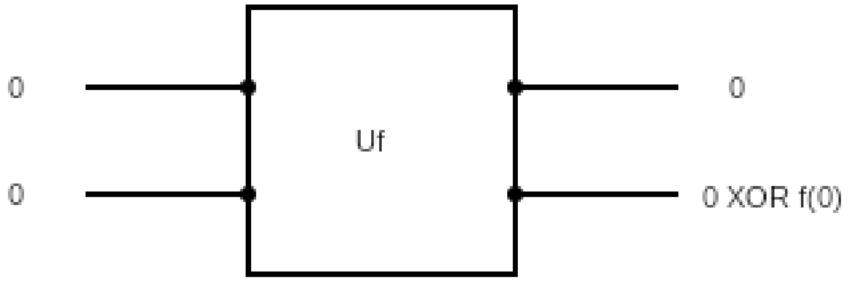
(i) $f(0) = f(1)$

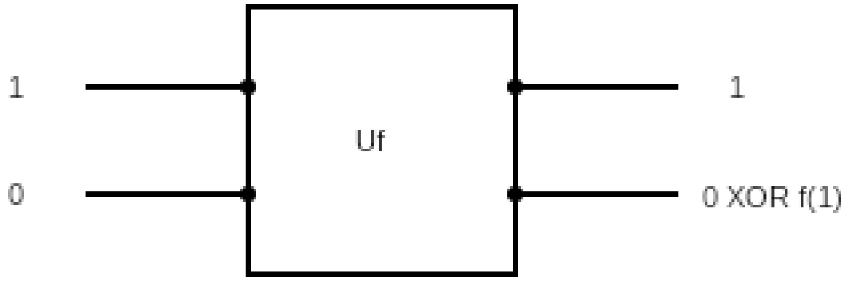
(ii) $f(0) \neq f(1)$

How many queries do we need to make to U_f to determine which of the two is true?

14.2 Classical Reversible Computation

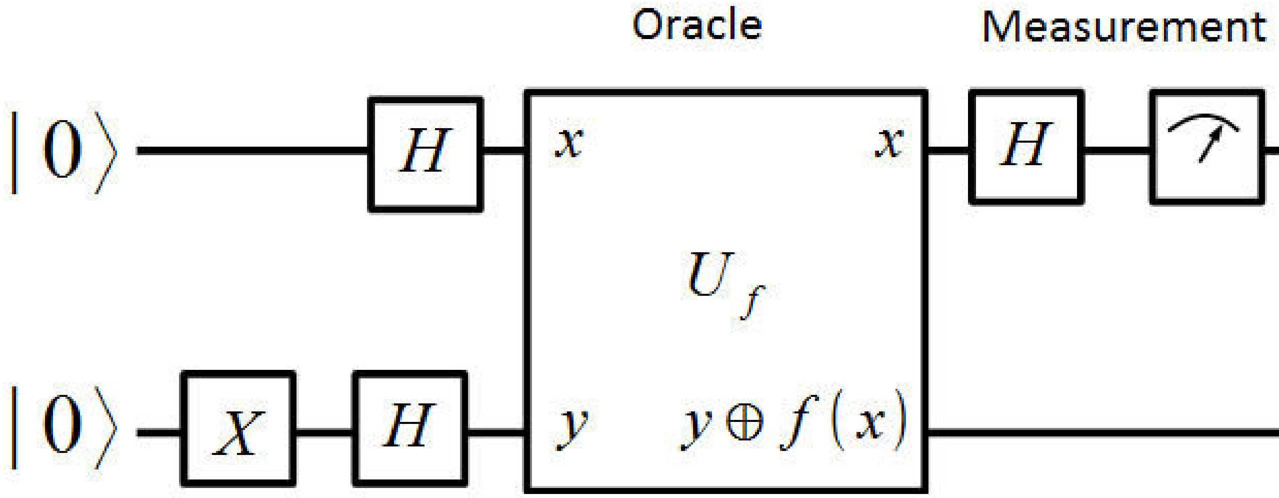
Classically, two queries to U_f are needed, one to determine the value of $f(0)$ and one to determine the value of $f(1)$.
We can then compare the two and decide which promise is true.





14.3 Quantum Computation

Consider the following quantum circuit:



14.3.1 Finding Final State

Finding the output:

$$
\begin{aligned}
|0\rangle|-\rangle &\xrightarrow{H\otimes\mathbb{I}} |+\rangle|-\rangle && \textit{Apply H to first register}\\
&\xrightarrow{U_f} \frac{1}{\sqrt{2}}((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|-\rangle)|-\rangle && \textit{Apply phase kickback}\\
&\xrightarrow{H\otimes\mathbb{I}} \frac{1}{\sqrt{2}}((-1)^{f(0)}|+\rangle + (-1)^{f(1)}|-\rangle)|-\rangle && \textit{Apply H to first register}\\
&= \frac{1}{\sqrt{2}}\left(\frac{(-1^{f(0)})(|0\rangle + |1\rangle)}{\sqrt{2}} + \frac{(-1^{f(1)})(|0\rangle - |1\rangle)}{\sqrt{2}}\right)|-\rangle
\end{aligned}
$$

By rearranging terms containing $|0\rangle$ and $|1\rangle$, we obtain the final state $|\psi\rangle$ as

$$
|\psi\rangle = \frac{1}{2}\left(((-1)^{f(0)} + (-1)^{f(1)})|0\rangle + ((-1)^{f(0)} - (-1)^{f(1)})|1\rangle\right)
$$

Note that the $|-\rangle$ in the second register has been dropped since it is implicit for a phase kickback oracle.

14.3.2 Probability Distribution

The probabilities of the final states being $|0\rangle$ and $|1\rangle$ can be calculated from the square of the corresponding amplitudes of $|0\rangle$ and $1\rangle$, giving

$$\mathbb{P}(|0\rangle) = \frac{1}{4}\left((-1)^{f(0)} + (-1)^{f(1)}\right)^2$$
$$\mathbb{P}(|1\rangle) = \frac{1}{4}\left((-1)^{f(0)} - (-1)^{f(1)}\right)^2$$

14.3.3 Resolving a Promise

To resolve which one of the two promises are true, we measure the final state. If $f(0) = f(1)$

$$\langle 0|\psi\rangle = 1$$
$$\langle 1|\psi\rangle = 0$$

If $f(0) \neq f(1)$

$$\langle 0|\psi\rangle = 0$$
$$\langle 1|\psi\rangle = 1$$

Thus, if the final state is orthogonal to $|1\rangle$, then $f(0) = f(1)$ and if it is orthogonal to $|0\rangle$, then $f(0) \neq f(1)$.

14.3.4 Comparison

We observe that the quantum computer needs only 1 query while the classical reversible computer needed 2 queries to U_f.

15 Deutsch-Jozsa Algorithm

15.1 The Problem

This is a generalisation of the Deutsch algorithm that we previously saw. In this algorithm, the boolean function f is from n-bit strings to a bit, i.e.
$f : \{0,1\}^n \to \{0,1\}$.

Promises:

(i) f is constant, i.e. $f(x) = 0 \ \forall x \in \{0,1\}^n$ or $f(x) = 1 \ \forall x \in \{0,1\}^n$

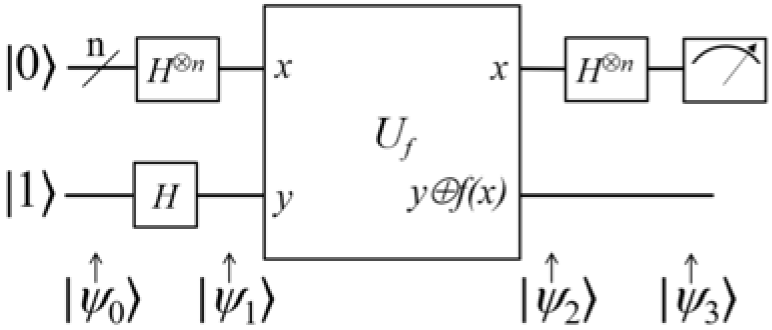
(ii) f is balanced, i.e.

$$\begin{aligned} f(x) = 0 \; &for \; \frac{2^n}{2} \; values \; of \; x \\ f(x) = 1 \; &for \; the \; other \; \frac{2^n}{2} \; values \; of \; x \end{aligned}$$

15.2 Classical Reversible Computing

Classically, to resolve a promise with probability 1, the worst case number of queries needed to U_f is $\frac{2^n}{2}+1$. This is when out of the 2^n possible n-bit strings to evaluate, the first half, i.e. $\frac{2^n}{2}$ strings all give the same output, either 0 or 1. Now, we need one additional query to resolve a promise. If it is the same as the result of the first half of bit strings, then f is constant, else f is balanced.

15.3 Quantum Computing



15.3.1 Finding the Final state

$$\begin{aligned}
|0\rangle^{\otimes n}|-\rangle \xrightarrow{H^{\otimes n}\otimes \mathbb{I}} & \frac{1}{\sqrt{2^n}}\left(\sum_{x\in\{0,1\}^n} |x\rangle\right)|-\rangle && Applying\ H^{\otimes n} onfirstregisterset \\
\xrightarrow{U_f} & \frac{1}{\sqrt{2^n}}\left(\sum_{x\in\{0,1\}^n} (-1)^{f(x)}|x\rangle\right)|-\rangle && Applying\ phase\ kickback \\
\xrightarrow{H^{\otimes n}\otimes \mathbb{I}} & \frac{1}{\sqrt{2^n}}\left(\sum_{x\in\{0,1\}^n} (-1)^{f(x)}\frac{1}{\sqrt{2}}\left(\sum_{z\in\{0,1\}^n} (-1)^{x\cdot z}|z\rangle\right)\right) && Applying\ H^{\otimes n} onfirstregisterset \\
|\psi\rangle = & \frac{1}{2^n}\sum\sum_{x,z\in\{0,1\}^n}\left((-1)^{f(x)+x\cdot z}\ |z\rangle\right) && Final\ state
\end{aligned}$$

Checking the inner product of the final state with an n-bit string of 0s,

$$\begin{aligned} \langle 00\dots 0|\psi\rangle = \frac{1}{2^n}\sum_{x\in\{0,1\}^n} & (-1)^{f(x)} && f(x)\in\{0,1\} \\ =&1 && \textit{if } f(x)=0,\ f(x) \textit{ is constant} \\ &-1 && \textit{if } f(x)=1,\ f(x) \textit{ is constant} \\ &0 && \textit{if } f(x) \textit{ is balanced} \end{aligned}$$

15.3.2 Probabilities of Final State

Finding the probabilities of the final state using the squares of amplitudes,

$$\begin{aligned} \mathbb{P}(|00\dots 0\rangle) =&(\pm 1)^2 = 1 && \textit{if } f(x) \textit{ is constant} \\ & 0^2 = 0 && \textit{if } f(x) \textit{ is balanced} \end{aligned}$$

15.3.3 Resolving a Promise

If $f(x)$ is constant, then the measured final state $|\psi\rangle$ will be $|00\dots 0\rangle$ with probability 1.
If $f(x)$ is balanced, then the measured final state $|\psi\rangle$ is a state other than $|00\dots 0\rangle$ with probability 1.

15.3.4 Comparison

Compared to the classical reversible computer, which needed a worst case of $\frac{2^n}{2}+1$ queries to C_f, the quantum computer needs only 1 query to U_f to resolve a promise. This is an exponential speedup. Note, however, that this exponential speedup is when we must resolve the correct promise with probability 1. If we allow for an ε uncertainty to both, the speedup offered by the quantum computer will reduce to $\mathcal{O}(\log\frac{1}{\varepsilon})$, as seen in Assignment 1.

16 Quantum Search - Grover's Algorithm

Quantum search algorithms provide quadratic speedup compared to classical algorithms. That is if it takes classical algorithms $O(N)$ steps to run, it would take a quantum algorithm $O(\sqrt{N}$ steps to run with a high probability of success.

16.1 Problem Statement

Let us have a set X of $N = 2^n$ elements

$$X = \{x_1, x_2, ..., x_N\}$$

and a Boolean function $f : X \to \{0, 1\}$. x_i are bitstrings of length n.
Find elements $x* \in X$ such that $f(x*) = 1$.

The classical algorithm to solve this would always need $O(N)$ queries to the function f.

It's complexity is $O(N) = O(2^n)$ both in the average case and the worst case classically. However the quantum approach allows us to speed this up quadratically. This is achieved as shown below.

16.2 The Quantum Approach

We have seen in previous lectures the following observation:

$$H^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{N}} \sum_{z \in 0,1^n} |z\rangle$$

Hence the Hadamard gate when applied to $|0\rangle^{\otimes n}$ converts it to an equal superposition of all states in the computational basis. Now from now on let $|S\rangle = \frac{1}{\sqrt{N}} \sum_{z \in 0,1^n} |z\rangle$.

We now introduce a phase kickback oracle:

$$U_f : |x\rangle \rightarrow (-1)^{f(x)} |x\rangle$$

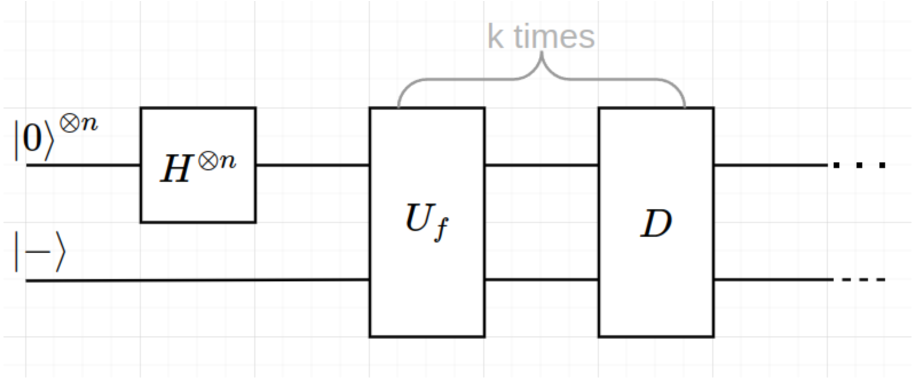
This gate will flip the phase of $|x\rangle$ for all $x*$, else it will keep the state unchanged. Applying this to our state $|S\rangle$:

$$U_f |S\rangle = \frac{1}{\sqrt{N}} (\sum_{x \notin x*} |x\rangle - \sum |x*\rangle)$$

So only phase of $|x*\rangle$ is flipped.

16.3 The algorithm

The grover's algorithm is then defined as follows:



$$G = (DU_f)^k |S\rangle$$

for a suitable k. D is a gate that is called the diffuser. We now rewrite $|S\rangle$ as follows:

$$|S\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle$$

$$= \frac{1}{\sqrt{N}}(\sum_{x':f(x')=1} |x'\rangle + \sum_{x'':f(x'')=1} |x'\rangle)$$

Now let, $|\{x : f(x) = 1\}| = M$. Define the following:

$$|\omega\rangle = \frac{1}{\sqrt{M}} \sum_{x':f(x')=1} |x'\rangle$$

$$|S\omega\rangle = \frac{1}{\sqrt{N-M}} \sum_{x'':f(x'')=0} |x''\rangle$$

Then,

$$|S\rangle = \frac{\sqrt{M}}{\sqrt{N}} |\omega\rangle + \frac{\sqrt{N-M}}{\sqrt{N}} |S\omega\rangle$$

Now we see that, $\langle S\omega|\omega\rangle = 0$, so the basis $\{|\omega\rangle, |S\omega\rangle\}$ forms an orthonormal set.

Let $sin(\theta/2) = \sqrt{\frac{M}{N}}$ and $cos(\theta/2) = \sqrt{\frac{N-M}{N}}$.

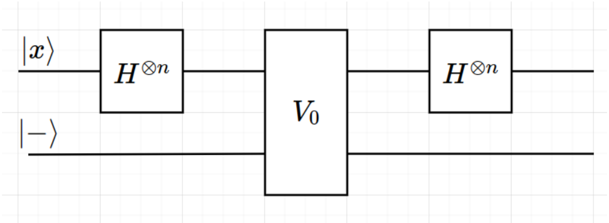
So,

$$|S\rangle = sin(\theta/2) |\omega\rangle + cos(\theta/2) |S\omega\rangle$$

$$U_f |S\rangle = -sin(\theta/2) |\omega\rangle + cos(\theta/2) |S\omega\rangle$$

16.4 D Gate

In the algorithm, the D gate is given by the following circuit:



The V_0 gate performs a controlled phase shift. If $|x\rangle = |0\rangle^{\otimes n}$ then no phase shift happens, else the phase is flipped.

This gives:

$$V_0 = 2 |0\rangle^{\otimes n} \langle 0|^{\otimes n} - \mathbb{I}$$

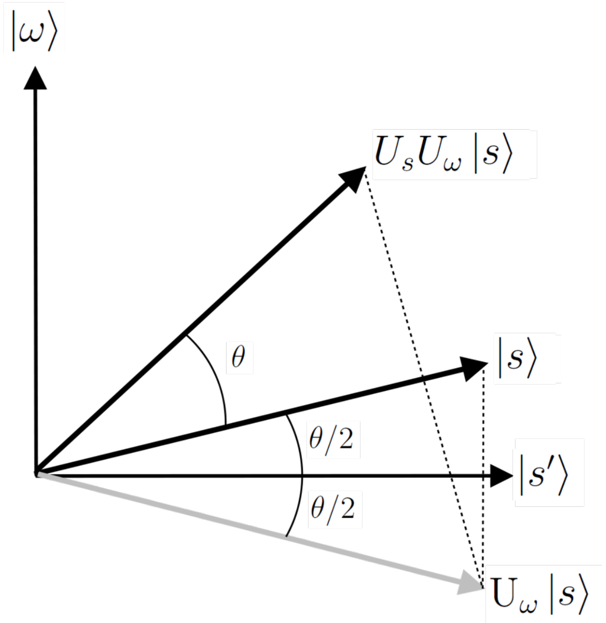
we can also write,

$$V_0 : |x\rangle \rightarrow (-1)^{OR(x_1,x_2,\dots)} |x\rangle$$

since if at least one bit is non 0, there is a phase kickback. So,

$$\begin{aligned} D &= H^{\otimes n} V_0 H^{\otimes n} \\ &= 2H^{\otimes n} |0\rangle^{\otimes n} \langle 0|^{\otimes n} H^{\otimes n} - H^{\otimes n} \mathbb{I} H^{\otimes n} \\ &= 2|S\rangle\langle S| - (H^2)^{\otimes n} \\ &= 2|S\rangle\langle S| - \mathbb{I} \end{aligned}$$

16.5 Working of the algorithm



In the above diagram of the algorithm, the U_f gate flips the state along the axis for $|S\omega\rangle$ and the D gate flips it across the state $|S\rangle$. The combined effect is a rotation by angle of θ counter clockwise on the diagram. Hence the state's overlap with the solution state ω increases.

However then for the algorithm to work we must have $M << N$. This would mean θ is considerably smaller than $\pi/2$. Then to get maximum overlap, we take k as follows:

$$\theta/2 + k\theta \approx \pi/2$$

So S is basically

$$sin(\theta/2)] |w\rangle + cos(\theta/2) |S_{\overline{w}}\rangle$$

$D = 2 |S\rangle \langle S| - I$

$$D = 2[|w\rangle \langle w| sin^2(\theta/2) + |w\rangle \langle S_{\overline{w}}| sin(\theta/2)cos(\theta/2)$$
$$+ |S_{\overline{w}}\rangle \langle w| sin(\theta/2)cos(\theta/2) + |S_{\overline{w}}\rangle \langle S_{\overline{w}}| cos^2(\theta/2)] - I$$

After all D is a 2 × 2 matrix
So in the basis $|S_{\overline{w}}\rangle , |w\rangle$ D can be represented as :

$D = \begin{bmatrix} cos(\theta) & sin(\theta) \\ sin(\theta) & -cos(\theta) \end{bmatrix}$
We are given $G = D \cdot u_f$

$G = \begin{bmatrix} cos(\theta) & sin(\theta) \\ sin(\theta) & -cos(\theta) \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
$G = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}$
As we can see G is a rotation matrix.
$|S\rangle = G^k \cdot |S\rangle$
Let's see the product one time :
$G |S\rangle = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} |S\rangle$
$G |S\rangle = sin(\theta + \theta/2) |w\rangle + cos(\theta + \theta/2) |S_{\overline{w}}\rangle$
After k times :
$|S\rangle = sin(k\theta + \theta/2) |w\rangle + cos(k\theta + \theta/2) |S_{\overline{w}}\rangle$
At that time : $sin(k\theta + \theta/2) = 1$
$\implies \theta(k + 1/2) = \pi/2$
$\implies \theta = \frac{\pi}{2k+1}$ We know that
$sin(\theta/2) = \sqrt{\frac{M}{N}} \approx \theta/2$
$2k + 1 = \frac{\pi\sqrt{N}}{2\sqrt{M}}$
$k = \frac{\pi\sqrt{N}}{2\sqrt{M}} - \frac{1}{2}$
$k \approx O(\sqrt{\frac{N}{M}})$
So finally
$G^k |S\rangle = \frac{1}{\sqrt{M}} \sum_{f(|x\rangle=1)} |x\rangle$

Now what if M is unknown :-
(a) Estimate 'M' before only (Quantum Counting)
(b) Randomized Quantum Search

Amplification of amplitude :
$G^k H^{\otimes n} |0^n\rangle = G^k[sin(\theta/2)] |w\rangle + cos(\theta/2) |S_{\overline{w}}\rangle]$
$\sqrt{p} = sin(\theta/2)$

To amplify this term to 1 I need $\frac{1}{\sqrt{p}}$ queries.
$A^{\frac{1}{\sqrt{p}}} u \left|0\right\rangle \approx \left|\psi_{good}\right\rangle$

17 Models of Quantum Computing

1. Adiabatic Model :
$H(s) = (1-s)H_0 + sH_k$
$s \in [0,1]$
2. Quantum walks
3. MDQC
4. Topological Quantum Channel