Introduction to Quantum Information and Computation

LECTURE NOTES

Hardik Sharma
Jai Bhatnagar
Ashmit Chamoli
Kyrylo Shvyam Kumar
Romica Raisinghani
Sriteja Reddy Pashya
Keval Jain

31 January - 24 February 2023

Lecture 8 - Keval Jain

Quantum Computing

A quantum computer is a computer that exploits quantum mechanical phenomena, ie machines that obey quantum physics. It is a natural generalization of computing in general. Unlike classical computers, which use bits that can be either 0 or 1, quantum computers use quantum bits (qubits) that can exist in superposition.

Church Turing and Extended Church Turing Thesis

The Church-Turing thesis asserts that any function that is computable can be computed by a Turing machine, or equivalently, by any other computational model that is effectively equivalent to a Turing machine ie the set of functions that are computable by a Turing machine is identical to the set of functions that are computable by any other algorithmic method.

When one computer simulates another, there is usually some 'overhead' cost associated with the simulation. A simulation of one computer by another is efficient if the 'overhead' in resources used by the simulation is polynomial (simulating $O(f(x))$ to $O(f(x)^k)$ is efficient). We can write a computer program for a universal Turing machine that will simulate the evolution of any computing device on input any input. The Church-Turing thesis, however, does not make any claims about the efficiency of these algorithms.

Strong/Extended Church–Turing Thesis: A probabilistic Turing machine can efficiently simulate any realistic model of computation. The Strong Church Turing thesis (still classical) extends on this idea, and gives us an insight on the efficiency of computation. A "probabalistic Turing Machine" is a turing machine with the extra ability to make a random binary choice at each step (built in coin-flipper).

The problem with the classical strong Church–Turing Thesis is that it appears that classical physics is not powerful enough to efficiently simulate quantum physics. The answer may be a quantum version of the strong Church–Turing Thesis, where we replace the probabilistic Turing machine with some reasonable type of quantum computing model —
Quantum Strong Church–Turing Thesis: A quantum Turing machine can efficiently simulate any realistic model of computation.

Richard Feynman

Feynman was one of the founders of the field of quantum computing. He was particularly interested in the idea of using quantum mechanical systems to perform calculations that would be impossible or impractical to perform using classical computers. One of Feynman's most famous contributions to quantum computing is the concept of the quantum simulator. In a 1981 paper titled "Simulating Physics with Computers," Feynman proposed the idea of using a quantum computer to simulate the behavior of other quantum systems, such as atoms and molecules. He argued that this would be a much more efficient way of studying the properties of these systems than using classical computers.

Quantum Computing in Circuit Model

In quantum computing, the basic operations on quantum states are state preparation, evolution, and measurement, they form the basic building blocks of quantum computing.

1) State Preperation: In order to perform a quantum computation, we need to start with a well-defined quantum state. This is accomplished through a process called state preparation, where we prepare the qubits in a specific initial state. The state can be a simple state such as the computational basis state, or a more complex state that is prepared using a quantum algorithm or a combination of quantum gates.

2) Evolution: After preparing the initial state, we can apply quantum gates and operations to the qubits to manipulate and evolve the quantum state. Evolution can be described by a unitary transformation that maps the initial state to the final state.

$$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$$

$|\psi(0)\rangle$ - Initial state of the system at time t=0.
$|\psi(t)\rangle$ - State of the system at time t.
H - Hamiltonian Operator that represents total energy of the system. Eigen values of H are energy levels.
e^{-iHt} - Time evolution operator, it describes how the quantum state changes over time. It is unitary.

$$|\psi_f\rangle = U_t U_{t-1} ... U_1 |\psi_0\rangle$$

Where each Unitary is a quantum gate.

3) Measurement: Finally, we can extract information from the quantum state by performing a measurement. A measurement is a process that extracts a classical value from a quantum state. It collapsing the quantum state onto a classical state. The measurement outcome is random, and the probability distribution of the outcomes depends on the quantum state being measured.
The state f is observed with probability -

$$P = |\langle f|\psi_f\rangle|^2$$

Quantum designer prepare these operators $U_t, U_{t-1} ... U_1$ in a clever manner such that this probability is high. (Constructive interference in the initial and final states)

$$U_j = e^{-iH_j t}$$

$$|\psi_j\rangle = U_j|\psi_0\rangle$$

$$U_j{}^{\dagger}|\psi_f\rangle = |\psi_0\rangle$$

Also since $U^{\dagger} = U^{-1}$

$$U_j{}^{-1}|\psi_f\rangle = |\psi_0\rangle$$

Lecture 9 - Kyrylo Shyvam Kumar

• Recap:

1) State preparation: Here the inital state is prepared.We can start with initial state like $|0\rangle^{\otimes n}$ or some other sate we need, and later evolve it into final state by applying:

$$i\frac{\mathrm{d}\,|\psi\rangle}{\mathrm{d}t} = \frac{H}{\hbar}\,|\psi\rangle$$

2) Evolution: The initial state $|\psi_i\rangle$ evolves into new state, atre applying sequence of unitary operators.

$$|\psi_f\rangle = U_t U_{t-1} ... U_1\,|\psi_i\rangle$$

3) Measurement: Measure the final state in a computational basis to obtain $|f\rangle$, with probability $p = \langle f|\psi_f\rangle^2$

While designing quantum algorithms we want to maximize probability p of z_{good} in :

$$|\psi_f\rangle = \sqrt{p}\,|z_{good}\rangle + \sqrt{1-p}\,|z_{good}\rangle^{\perp}$$

• Classical Logic Gate:

Any boolean function $f : \{0,1\} \to \{0,1\}$ can be written as propositional formula of n variables. e.g. *A and B*.
OR
Logic gates are physical implementation of AND, OR, NOT, etc.

• Landauer's principle:

Any logically irreversible manipulation of information results in increase of entropy and release of heat. In other words, in order to erase information, it is necessary to dissipate energy.

Classically, $\{\wedge, \vee, \neg\}$ are universal. But only $\neg$ is actually reversible(results in no loss of information).

• Energy cost of computation:

A circuit which resets any bit to 0, has loss of information for the initial bit.
And entropy quantizes lack of information in following way:

$$S_{initial} = -\sum p_x \log x = -\frac{1}{2}\log\left(\frac{1}{2}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right)$$
$$S_{final} = 0$$
$$\Delta S = -\log(2)$$

The change of entropy will be positive for collection of system and surrounding. The exact bound of heat dissipated comes from Classius inequality.

• Classius Inequality:

Here ,T is ambient temperature, ΔQ is heat cost of a system.

$$\frac{\Delta Q}{T} \geq K_B \Delta S$$
$$\implies \Delta Q \geq K_B T \Delta S$$
$$\implies \Delta Q \geq K_B T \log 2$$

• Szilard's Engine:

It is a physical thought experiment demonstrating how just the possession of information might in principle have thermodynamic consequences. When the demon gets the location of a single molecule in container (right or left chamber of container), he gains information which injects energy into a system. This information later can produce work equal to $K_B T \log 2$, when demon allows molecule to push separation reversibly and again reach initial state.

• Maxwell's demon:

With the help of Landauer's principle we can explain 2^{nd} law of thermodynamics in this thought experiment too. Erasing information is a thermodynamically irreversible process that increases the entropy of a system. Any demon must "generate" more entropy segregating the molecules than it could ever eliminate by the method described. To determine whether to let a molecule through, the demon must acquire information about the state of the molecule and either discard it or store it.

• Reversible gates:

In the reversible gates, we try not to erase any information ,leading to
$\Delta S_{information} = 0$
Just like time and space, energy is also a computing resource.

1. Fredkin Gate (Controlled SWAP):

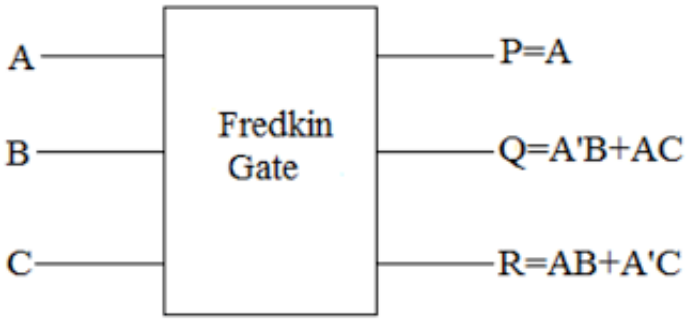


Figure 1: CSWAP Gate

In the above diagram: If $A = 1$, we $swap(B, C)$

So, when $A = 0 \rightarrow Q = B$ and $R = C$
when $A = 1 \rightarrow Q = C$ and $R = B$

Also $C - SWAP$ is a universal gate:

Figure 2: CSWAP Gate acting as AND, OR, NOT gates

From the above diagram it is visible that we can perform reversible AND, OR, NOT operations using this gate.

2. Controlled NOT Gate (CNOT):

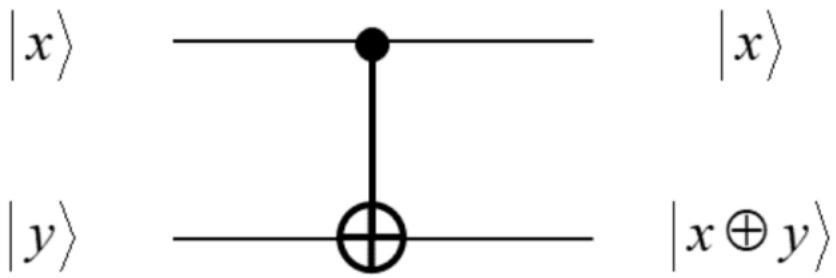


Figure 3: CNOT Gate

It is not universal as atleast 3 bits are needed to preserve information.

In quantum computations we need to clear garbage values. We cannot perform measurements as they may destroy superposition and entanglement. Here comes the need for uncomputation.

Lecture 10 - Romica Raisinghani

Reversible computing and garbage bits

The action of a reversible circuit in reversible computing may be written as:

$$(x, 0, 0) \rightarrow (x, f(x), g(x))$$

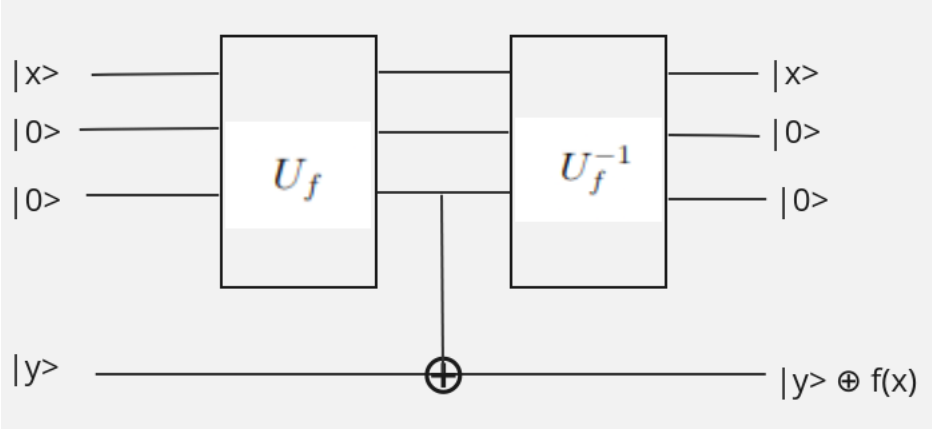
The above equation is a very useful way of writing the action of the reversible circuit, because it allows an idea known as **uncomputation** to be used to get rid of the garbage bits, for a small cost in the running time of the computation. Garbage bits are dependent on the input and they are a problem for the quantum computer as the garbage register gets entangled with the register computing f(x). Uncomputation is a technique, used in reversible circuits, for cleaning up temporary effects on ancilla bits so that they can be reused.

The idea is the following. Suppose we start with a four register computer in the state (x, 0, 0, y). The second register is used to store the result of the computation, and the third register is used to provide workspace for the computation, that is, the garbage bits g(x). We assume that the fourth register starts in an arbitrary state y.

We apply a reversible circuit to compute f , resulting in the state (x, f (x), g(x), y). Next, we use CNOTs to add the result f(x) bitwise to the fourth register, leaving the machine in the state (x, f (x), g(x), $y \oplus f(x)$). However, all the steps used to compute f (x) were reversible and did not affect the fourth register, so by applying the reverse of the circuit used to compute f we come to the state $(x, 0, 0, y \oplus f(x))$. The following equations show the uncomputation process described:

$$(x, 0, 0, y) \xrightarrow{U_f} (x, f(x), g(x), y) \xrightarrow{CNOT_{2,4}} (x, f(x), g(x), f(x) \oplus y) \xrightarrow{U_f^{-1}} (x, 0, 0, y \oplus f(x))$$

where y can be set to 0.
If x has n-bits , then we need to apply CNOT bitwise on all n-bits.
Below is the circuit representation for the description above:



Quantum Gates

Quantum gates are unitary operations on quantum circuits. A quantum logic gate (or simply quantum gate) is a basic quantum circuit operating on a small number of qubits. They are the building blocks of quantum circuits, like classical logic gates are for conventional digital circuits. The quantum gates acting on single,two and three qubit systems are briefly described below:

Single Qubit Quantum Gates

The computational basis in the single qubit system are given by:

$$[|0\rangle, |1\rangle]$$

All single qubit quantum gates are unitary matrices satisfying the below properties:

- $U^\dagger U = I$ (where $U^\dagger$ is the adjoint of U)
- $|U| = 1$
- The columns of U satisfy orthonormal condition

Any single qubit quantum gate is given by:

$$\mathrm{U} = e^{i\alpha} R_n(\theta)$$

where $R_n(\theta)$ is a rotational operator about any arbitary direction
Also,

$$R_n(\theta) = e^{-i\frac{\theta}{2}\vec{\sigma}\hat{n}}$$

where $\vec{\sigma} = (\hat{\sigma}_x, \hat{\sigma}_y, \hat{\sigma}_z)$ is a three component spin vector and $n = (\hat{n}_x, \hat{n}_y, \hat{n}_z)$ is a real unit vector in three dimensions such that $\hat{n}_x^2 + \hat{n}_y^2 + \hat{n}_z^2 = 1$. Hence,

$$U = e^{i\alpha}[cos(\tfrac{\theta}{2})I - isin(\tfrac{\theta}{2})\vec{\sigma}\hat{n}]$$

put $\alpha = \frac{\pi}{2}$ and $\theta = \pi$

$$U = \vec{\sigma}\hat{n}$$

$$U = \sigma_x n_x + \sigma_y n_y + \sigma_z n_z$$

Below are some of the single qubit quantum gates:

Pauli-X Gate or NOT Gate

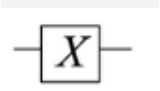
Now, when $\hat{n} = (1, 0, 0)$

$$U = \sigma_x$$

This is the **Pauli-X** gate also called **NOT** gate

$$U = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

It is represented as:



It acts in the following way:
$\mathrm{X}|0\rangle = |1\rangle$
$\mathrm{X}|1\rangle = |0\rangle$

Pauli-Y Gate

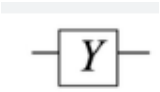
When $\hat{n} = (0, 1, 0)$

$$U = \sigma_y$$

This is the **Pauli-Y** gate

$$U = \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

It is represented as:



It acts in the following way:
$Y|0\rangle = i|1\rangle$
$X|1\rangle = -i|0\rangle$

Pauli-Z Gate

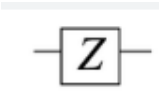
When $\hat{n} = (0, 0, 1)$

$$U = \sigma_z$$

This is the **Pauli-Z** gate

$$U = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

It is represented as:



It acts in the following way:
$X|0\rangle = |0\rangle$
$X|1\rangle = -|1\rangle$

Hadamard Gate

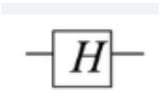
When $\hat{n} = (\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}})$

$$U = \frac{1}{\sqrt{2}}(\sigma_x + \sigma_z)$$

This is the **Hadamard** gate

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

It is represented as:



It acts in the following way:
$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle$
$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$

Also, $H|+\rangle = |0\rangle$
$H|-\rangle = |1\rangle$

We know that the rotational operator is given by:

$$R_n(\theta) = e^{-i\frac{\theta}{2}\vec{\sigma}\hat{n}}$$

Let us calculate the rotational operator $R_z(\theta)$ along the z-axis:

$$R_z(\theta) = e^{-i\frac{\theta}{2}\sigma_z}$$

$$R_z(\theta) = cos(\tfrac{\theta}{2})I - isin(\tfrac{\theta}{2})\sigma_z$$

$$R_z(\theta) = \begin{pmatrix} cos(\frac{\theta}{2}) & 0 \\ 0 & cos(\frac{\theta}{2}) \end{pmatrix} - \begin{pmatrix} isin(\frac{\theta}{2}) & 0 \\ 0 & -isin(\frac{\theta}{2}) \end{pmatrix}$$

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

Multiply each element of the above matrix by $e^{i\theta/2}$. This is known as **adding a global phase**. By doing this the operation remains the same but there is a change of phase.
Hence,

$$R_\theta(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$$

This acts on the computational basis in the following way:
$R_\theta|0\rangle = |0\rangle$
$R_\theta|1\rangle = e^{i\theta}|1\rangle$

It will act on any arbitrary state $\alpha|0\rangle + \beta|1\rangle$ as:
$R_\theta(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle + e^{i\theta}\beta|1\rangle$

Now substitute $\theta = \frac{\pi}{2}$ in R_θ

$$R_\theta(\tfrac{\pi}{2}) = S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

This is called the **phase** operator.

Now substitute $\theta = \frac{\pi}{4}$ in R_θ

$$R_\theta(\tfrac{\pi}{4}) = T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$$

This is called the **square root of S** or **T** operator.

Two Qubit Quantum Gates

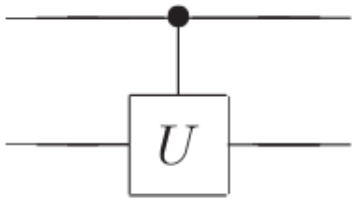
The two qubit quantum gates takes two qubits as input and outputs two qubits. The computational basis in the two qubit system are given by:

$$[|00\rangle, |01\rangle, |10\rangle, |11\rangle]$$

The gates in the two qubit system are known as **controlled unitary operators**. The controlled unitary matrix for a two qubit quantum gate is given by:

$$CU = \begin{pmatrix} I_2 & O_2 \\ O_2 & U \end{pmatrix}$$

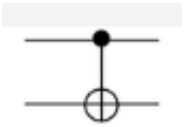
where I_2 is a 2×2 identity matrix, O_2 is a 2×2 null matrix and U is a single qubit quantum gate. A general representation of these gates is given by:



Below are some of the two qubit quantum gates:

Controlled-NOT or CNOT Gate

This gate has two input qubits, known as the control qubit and the target qubit, respectively. One of the outputs of this gate is the control qubit itself, and the other output depends whether the control qubit is set or not. If the control qubit is $|1\rangle$, the second output is the flipped target qubit. In other words, if the control qubit is set, then the other output is the action of NOT gate on the target qubit. It is represented by:



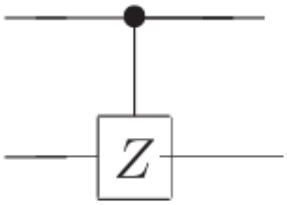
The matrix representation is given by:

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

This acts on the computational basis in the following way:
$CNOT|00\rangle = |00\rangle$
$CNOT|01\rangle = |01\rangle$
$CNOT|10\rangle = |11\rangle$
$CNOT|11\rangle = |10\rangle$
where the first qubit in input is the control qubit and the second is the target qubit.

Controlled-Z Gate

In this case, if the control qubit is $|1\rangle$, the other output is the action of Pauli-Z gate on the target qubit.
It is represented by:



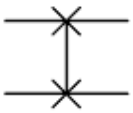
The matrix representation is given by:

$$CZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

This acts on the computational basis in the following way:
$CZ|00\rangle = |00\rangle$
$CZ|01\rangle = |01\rangle$
$CZ|10\rangle = |10\rangle$
$CZ|11\rangle = -|11\rangle$
where the first qubit in input is the control qubit and the second is the target qubit.

Swap Gate

The SWAP gate is two-qubit operation. Expressed in basis states, the SWAP gate swaps the state of the two qubits involved in the operation.
It is represented by:



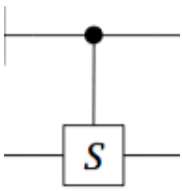
The matrix representation is given by:

$$Swap = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

This acts on the computational basis in the following way:
$Swap|00\rangle = |00\rangle$
$Swap|01\rangle = |10\rangle$
$Swap|10\rangle = |01\rangle$
$Swap|11\rangle = |11\rangle$

Controlled Phase or Cphase Gate

In this case, if the control qubit is $|1\rangle$, the other output is the action of single qubit S gate on the target qubit.
It is represented by:



The matrix representation is given by:

$$CPhase = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{pmatrix}$$

This acts on the computational basis in the following way:
$CS|00\rangle = |00\rangle$
$CS|01\rangle = |01\rangle$
$CS|10\rangle = |10\rangle$
$CS|11\rangle = i|11\rangle$
where the first qubit in input is the control qubit and the second is the target qubit.

Three Qubit Quantum Gates

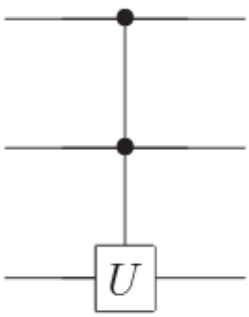
The three qubit quantum gates takes three qubits as input and outputs three qubits. The computational basis in the three qubit system is given by:

$$[|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle]$$

The gates in the two qubit system are known as **controlled controlled unitary operators**. A general representation of these gates is given by:

$$CCU = \begin{pmatrix} I_4 & O_4 \\ O_4 & CU \end{pmatrix}$$

where I_4 is a 4×4 identity matrix, O_4 is a 4×4 null matrix and CU is a two qubit quantum gate. A general representation of these gates is given by:



Below are some of the three qubit quantum gates:

Controlled controlled NOT gate or Toffoli Gate

This gate has three input qubits, two of them are control qubits and one is the target qubit. Two outputs are the control qubits itself and when both the control qubits are set, the third output is the action of NOT gate on the target qubit.

It is represented by:

The matrix representation is given by:

$$Toffoli = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

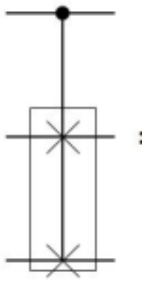
This acts on the computational basis in the following way:
$Toffoli|000\rangle = |000\rangle$
$Toffoli|001\rangle = |001\rangle$
$Toffoli|010\rangle = |010\rangle$
$Toffoli|011\rangle = |011\rangle$
$Toffoli|100\rangle = |100\rangle$
$Toffoli|101\rangle = |101\rangle$
$Toffoli|110\rangle = |111\rangle$
$Toffoli|111\rangle = |110\rangle$

where the first two qubits in input are the control qubit and the third is the target qubit.

Controlled Swap or CSWAP Gate or Fredkin Gate

In this case, there is one control qubit and two target qubits. If the control qubit is $|1\rangle$, then the other two qubits are swapped. In other words, one output is the control qubit itself whereas the other two outputs are the action of the swap gate if the control qubit is set.

It is represented by:



The matrix representation is given by:

$$Cswap = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

This acts on the computational basis in the following way:
$Cswap|000\rangle = |000\rangle$
$Cswap|001\rangle = |001\rangle$
$Cswap|010\rangle = |010\rangle$
$Cswap|011\rangle = |011\rangle$
$Cswap|100\rangle = |100\rangle$
$Cswap|101\rangle = |110\rangle$
$Cswap|110\rangle = |101\rangle$
$Cswap|111\rangle = |111\rangle$

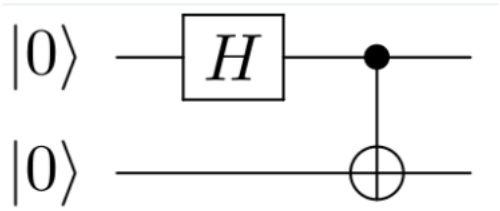
where the first qubit in input is the control qubit and the other two qubits are the target qubits.

Quantum Circuits

A quantum circuit is a model for quantum computation, similar to classical circuits, in which a computation is a sequence of quantum gates, measurements, initializations of qubits to known values, and possibly other actions.
The following are some of the properties of quantum circuits:

- Quantum circuits are a collection of quantum gates which must be unitary.
- Quantum circuits are acylic. There is no feedback system in quantum circuits.
- FAN-IN and FAN-OUT is not allowed in quantum circuits. This is because a qubit can neither be created nor be destroyed.
- Quantum circuits are reversible.

Now let us look at how the combination of quantum gates makes quantum circuits and let us evaluate the final states after computation.



The equivalent unitary operator of this circuit is given by:

$$U = CNOT(H \otimes I)$$

The initial state is : $|\psi_i\rangle = |0\rangle \otimes |0\rangle = |00\rangle$
The final state is given by:

$$|\psi_f\rangle = CNOT(H \otimes I)|00\rangle$$

$$|\psi_f\rangle = CNOT(H|0\rangle \otimes I|0\rangle)$$

$$|\psi_f\rangle = CNOT(|+\rangle \otimes |0\rangle)$$

$$|\psi_f\rangle = CNOT[\tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle]$$

$$|\psi_f\rangle = \tfrac{1}{\sqrt{2}} CNOT(|00\rangle + |10\rangle)$$

$$|\psi_f\rangle = \tfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Since, the final state cannot be written as a product of two separate states, this state is referred to as the **Entangled state or bell state**

Lecture 11 - Sriteja Pashya

General Quantum Circuits:

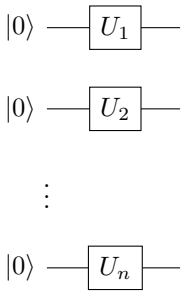
Elementary quantum gates can be composed into bigger quantum elements.

Parallel

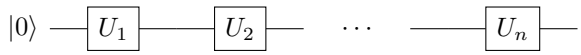
The composition of n qubits $|0\rangle$ through the gates U_i in parallel is given by

$$(\mathcal{U}_1 \otimes \mathcal{U}_2 \otimes \cdots \otimes \mathcal{U}_n) |00\cdots0\rangle$$
$$= (\mathcal{U}_1 \otimes \mathcal{U}_2 \otimes \cdots \otimes \mathcal{U}_n) |0\rangle^{\otimes n}$$

dimension of the output 2^n.



Series



The composition of 1 qubit $|0\rangle$ through the gates U_i in series is given by

$$(\mathcal{U}_n \mathcal{U}_{n-1} \cdots \mathcal{U}_1) |0\rangle$$

dimension of the output 2.

Complexity Classes

The complexity of a circuits can be measured in various ways. Some of the most common and important complexity classes are:

- **Depth Complexity**: The depth of a circuit is the maximum number of gates that must be traversed in any path from an input to an output. It measures the number of computational steps required to evaluate the circuit. A circuit with a smaller depth generally requires less time to execute than one with a larger depth.
- **Gate Complexity**: The gate complexity of a circuit is the total number of gates used to construct it. This metric measures the amount of hardware or software resources that the circuit requires to be implemented. The more gates a circuit has, the more complex it is to build and maintain.
- **Query Complexity**: The query complexity of a circuit is the minimum number of queries required to compute a particular function. A query corresponds to a single evaluation of the function at a specific input. The query complexity is particularly relevant for problems such as database search, where the number of queries required can be a bottleneck.

  In quantum circuits, it is possible to compute some functions with fewer queries than classical circuits.

Hadamard Gate

The Hadamard gate is a single-qubit quantum gate that maps the state $|0\rangle$ to $|+\rangle$ and $|1\rangle$ to $|-\rangle$. It is a unitary matrix and is given by

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

$|\psi\rangle$ — H —

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

$|0\rangle$ — H —
$|0\rangle$ — H —
⋮
$|0\rangle$ — H —
} n bits ⟶

$|0\rangle^{\otimes n}$ — $H^{\otimes n}$ — $|+\rangle^{\otimes n}$

$$(\mathcal{H}\otimes\mathcal{H}\otimes\cdots\otimes\mathcal{H})|0\rangle^{\otimes n} = |+\rangle^{\otimes n}$$
$$|+\rangle^{\otimes n} = \left[\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right]^{\otimes n}$$
$$= \frac{1}{\sqrt{2^n}}\sum_{z\in\{0,1\}^n}|z\rangle$$

let $x \in \{0,1\}^n$, then

$$\mathcal{H}^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}}((-1)^{|x\rangle|0\rangle}|0\rangle + (-1)^{|x\rangle|1\rangle}|1\rangle)$$
$$= \frac{1}{\sqrt{2^n}}\sum_{z\in\{0,1\}^n}(-1)^{x\cdot z}|z\rangle$$

Universality of Quantum Gates

A set of gates is said to be universal for quantum computation if any unitary operation may be approximated to arbitrary accuracy by a quantum circuit involving only those gates
Some examples of universal set of quantum gates are

- CNOT, single qubit gates
- CNOT, Hadamard, T gates

Solovay-Kitaev Theorem

Let $G = \{CNOT, H, S, T\}$ gates
It is possible to approximate any unitary gate in 1 or 2 qubits upto an error ϵ by using $polylog(\frac{1}{\epsilon})$ gates from G

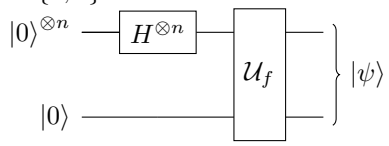
$$||\mathcal{U}|\psi\rangle - \mathcal{U}_t\mathcal{U}_{t-1}\ldots\mathcal{U}|\psi\rangle|| \leq \epsilon$$

Where $||v||$ is the Spectral Norm of v which is the maximum eigenvalue of density martix of v, and $t \in O(polylog(1/\epsilon))$

Formally the Solovay-Kitaev Theorem states that for any gate U on a single qubit, and given any $\epsilon > 0$, it is possible to approximate U to a precision ϵ using $O(polylog(1/\epsilon))$ gates from a fixed finite set.

Quantum Parallelism

Quantum parallelism is a concept in quantum computing that allows multiple computations to be performed simultaneously on a quantum computer. This is achieved through the use of qubits being in superposition.
Consider $f : \{0,1\}^n \to \{0,1\}$
$z \in \{0,1\}^n$



Initially

$$|0\rangle^{\otimes n} \xrightarrow{H^{\otimes n}} |+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} |z\rangle$$

Then

$$|+\rangle^{\otimes n} \xrightarrow{\mathcal{U}_f} |\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} |z\rangle\,|f(z)\rangle$$

With one query we can find the superposition of all 2^n possible output states.
Therefore, measurement will be $\{|z\rangle\,|f(z)\rangle\,, \frac{1}{2^n}\}$

Lecture 12 - Ashmit Chamoli

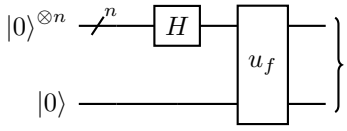
Recap

1. Universality of Quantum Circuits

   - Proved using **Solvay-Kitaev Theorem**: Any 't' gate unitary can be ϵ-approximated by $\mathcal{O}(t \cdot polylog(\frac{1}{\epsilon}))$ gates from a universal set.
     Example of a universal set: G = {CNOT, H, T}.

2. Quantum Parallelism

   - We can calculate the value of a boolean function at all points in a single query. Although, the output state here is not of practical use as it is just an equal superposition of all values and there is no constructive interference.



   Here,

$$H|x\rangle = \frac{1}{\sqrt{2}} \sum_{z\in\{0,1\}} (-1)^{xz} |z\rangle$$

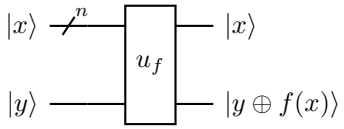
   for $x \in \{0,1\}$. And,

$$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} (-1)^{x\cdot z} |z\rangle$$

   for $x \in \{0,1\}^n$, where $x \cdot z = x_1 z_1 + ... + x_n z_n$ is the dot product of bit strings x and z.

Quantum Black Box: Phase-Kickback Oracle

Let $f : \{0,1\}^n \mapsto 0,1$ be a boolean function. Let's draw the black box for this function in a quantum circuit



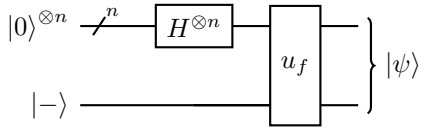
Now, when $f(x) = 0$,

$$|x\rangle|y\rangle \xmapsto{u_f} |x\rangle|y\rangle \quad \text{and} \quad |x\rangle|-\rangle \xmapsto{u_f} |x\rangle|-\rangle$$

and when $f(x) = 1$,

$$|x\rangle|y\rangle \xmapsto{u_f} |x\rangle|\bar{y}\rangle \quad \text{and} \quad |x\rangle|-\rangle \xmapsto{u_f} -|x\rangle|-\rangle$$

We observe that when $|y\rangle = |-\rangle$, we get a "phase kickback" of $-1^{f(x)}$ in the output. If we take the input $|x\rangle$ as $|0\rangle^{\otimes n}$ and apply a Hadamard gate in front of it and take $|y\rangle = |-\rangle$, we get



Where $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} (-1)^{f(z)} |z\rangle |-\rangle$.

Deutsch Algorithm

Problem Statement

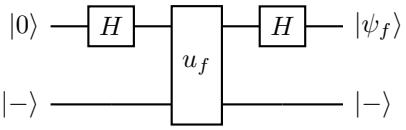
Suppose we are given u_f as a blackbox for a boolean function $f : \{0,1\} \mapsto 0,1$ with the promise that either (i) $f(0) = f(1)$ or (ii) $f(0) \neq f(1)$. How many queries to u_f are needed to determine which is the case?

Classical Solution

Classically, we need exactly 2 queries.

Solution using Quantum Algorithm

We need only 1 query.



Where $|\psi_f\rangle = \frac{1}{\sqrt{2}}[(-1)^{f(0)}|+\rangle + (-1)^{f(1)}|-\rangle]$. We see the quantum transformations step-wise:

$$\begin{aligned}
|0\rangle|-\rangle &\xmapsto{H\otimes\mathbb{1}} \frac{1}{\sqrt{2}}[|0\rangle + |1\rangle]|-\rangle \\
&\xmapsto{u_f} \frac{1}{\sqrt{2}}[(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle]|-\rangle \\
&\xmapsto{H\otimes\mathbb{1}} \frac{1}{\sqrt{2}}[(-1)^{f(0)}|+\rangle + (-1)^{f(1)}|-\rangle]
\end{aligned}$$

The final state of the first register is

$$\begin{aligned}
&\frac{1}{\sqrt{2}}[(-1)^{f(0)}|+\rangle + (-1)^{f(1)}|-\rangle] \\
=&\frac{1}{2}[(-1)^{f(0)}(|0\rangle + |1\rangle) + (-1)^{f(1)}(|0\rangle - |1\rangle)] \\
=&\frac{1}{2}[(-1)^{f(0)} + (-1)^{f(1)}]|0\rangle + [(-1)^{f(0)} - (-1)^{f(1)}]|1\rangle
\end{aligned}$$

Now, 2 cases arise:
Case $f(0) = f(1)$. Here, $|\psi_f\rangle = |0\rangle$ and so the probability of seeing 0 when we measure the first register becomes 1.
Case $f(0) \neq f(1)$. Here, $|\psi_f\rangle = |1\rangle$ and so the probability of seeing 1 when we measure the first register becomes 1.
Therefore in both cases, we have a clear outcome and the algorithm is complete.

Deutsch-Jozsa Algorithm

Problem Statement

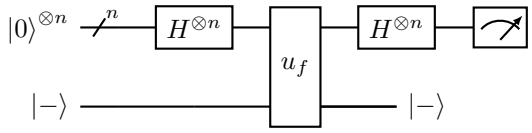
Suppose we are given u_f as a blackbox for a boolean function $f : \{0,1\}^n \mapsto 0,1$ with the promise that either (i) f is **BALANCED** or (ii) f is **CONSTANT**. How many queries to u_f are needed to determine which is the case?
CONSTANT: $f(x) = 0\ \forall x \in \{0,1\}^n$ or $f(x) = 1\ \forall x \in \{0,1\}^n$
BALANCED: $f(x) = 0$ for $\frac{2^n}{2}$ values of x and $f(x) = 1$ for the other $\frac{2^n}{2}$ values of x.

Classical Solution

Classically, we need $2^{n-1}+1$ queries in the worst case. This is when we check the value of the function on 2^{n-1} points and all of them turn out to be the same, so we will need one more query to decide between (i) and (ii).

Solution using Quantum Algorithm

We still need only 1 query.



We see the transformations step-wise:

$$
\begin{aligned}
|0\rangle^{\otimes n} &\xmapsto{H^{\otimes n}\otimes \mathbb{1}} \frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} |z\rangle\,|-\rangle \\
&\xmapsto{u_f} \frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} (-1)^{f(z)}\,|z\rangle\,|-\rangle \\
&\xmapsto{H^{\otimes n}\otimes \mathbb{1}} \frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} (-1)^{f(z)} \sum_{x\in\{0,1\}^n} (-1)^{x\cdot z}\,|x\rangle\,|-\rangle \\
&= \frac{1}{2^n} \sum_{z\in\{0,1\}^n} \sum_{x\in\{0,1\}^n} (-1)^{f(z)+x\cdot z}\,|x\rangle\,|-\rangle
\end{aligned}
$$

Let the final state of the first register be $|\psi\rangle$ Now,

$$
\begin{aligned}
p(|0...0\rangle) &= (\langle 0...0|\,|\psi\rangle)^2 \\
&= (\frac{1}{2^n} \sum_{z\in 0,1^n} (-1)^{f(z)})^2 \\
&= \begin{cases} 1, & \text{when f is } \textbf{CONSTANT} \\ 0, & \text{when f is } \textbf{BALANCED} \end{cases}
\end{aligned}
$$

Therefore, when we measure the final register, we get $|0\rangle^{\otimes n}$ when f is **CONSTANT** and something other than $|0\rangle^{\otimes n}$ if f is **BALANCED**.

Physics Behind the Deutsch Problem

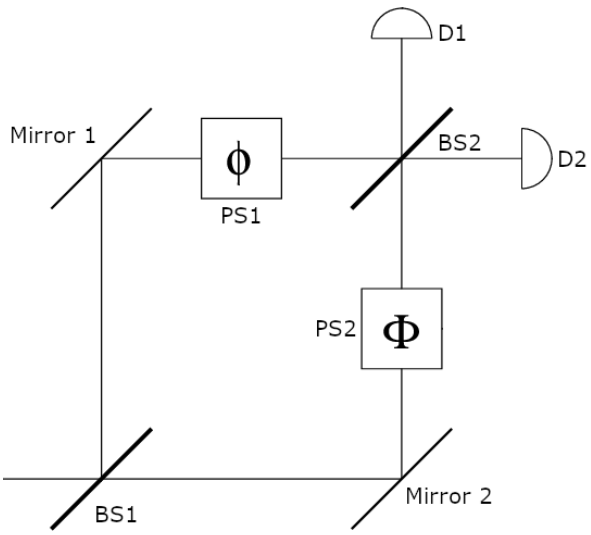


Figure 1: Mach-Zehnder Inferometer

Let $|0\rangle, |1\rangle$ denote the vertical and horizontal paths respectively.
Let $\Delta\phi = |\phi - \Phi|$. We are promised that either $\Delta\phi = 0$ or $\Delta\phi = \pi$. Which is the case ?

Lets look at the transformations at each step:

$$
\begin{aligned}
|0\rangle &\xmapsto{BS_1} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\
&\xmapsto{PS_1|PS_2} \frac{1}{\sqrt{2}}(e^{i\phi}\,|0\rangle + e^{i\Phi}\,|1\rangle) \\
&= \frac{e^{\frac{(\phi+\Phi}{2})}}{\sqrt{2}}[e^{i\frac{\phi-\Phi}{2}}\,|0\rangle + e^{-i\frac{\phi-\Phi}{2}}\,|0\rangle] \\
&\xmapsto{BS2} \frac{1}{2}[e^{\frac{i\Delta\phi}{2}}(|0\rangle + |1\rangle) + e^{-i\frac{-i\Delta\phi}{2}}(|0\rangle - |1\rangle)] \\
&= cos(\frac{\Delta\phi}{2})\,|0\rangle + isin(\frac{\Delta\phi}{2})\,|1\rangle
\end{aligned}
$$

If $\Delta\phi = 0$ the final state is $|0\rangle$.
If $\Delta\phi = \pi$ the final state is $i\,|1\rangle \equiv |1\rangle$.
Therefore, only D_1 clicks in the first case while only D_2 clicks in the second case.

Lecture 13 - Jai Bhatnagar

The Quantum Search Algorithm (Grover's Algorithm)

It offers a quadratic speedup over classical method.
i.e. if the classical method takes $f(n)$ time, this algorithm will take $\sqrt{f(n}$ time.

Problem Statement

Let us take a set of $N = 2^n$ elements i.e. $X = \{x_1, x_2, x_3, ..., x_N\}$ and a Boolean function $f : X \to \{0, 1\}$.
Find an element $x^* \in X$ such that $f(x^*) = 1$.

Solution

Number of Queries required Classically:

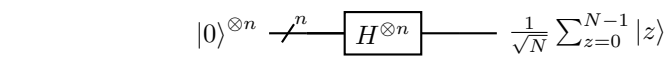
- Best Case - 1
- Worst Case - N
- Average Case - $\frac{N+1}{2}$

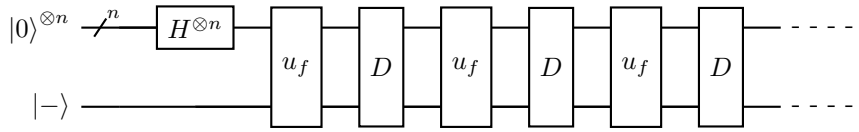
Therefore number of queries required classically would be $O(N)$.

Quantum Solution

From phase kickback oracle:

$$|x\rangle \to (-1)^{f(x)}|x\rangle$$





$$G = Du_f$$

$$|S\rangle = \frac{1}{\sqrt{N}} \sum_z |z\rangle$$

$$|x^*\rangle \approx G^K|S\rangle$$

where $|S\rangle$ is the state after the Hadamard gate is applied.

Suppose there are M such x^* ($M << N$), then out goal is:

$$G^K|S\rangle = \frac{1}{\sqrt{M}} \sum_{f(x^*)=1} |x^*\rangle$$

We have,

$$\begin{aligned} H^{\otimes n}|0\rangle^{\otimes n} &= |S\rangle \\ &= \frac{1}{\sqrt{N}} \sum_{x\in\{0,1\}^n} |x\rangle \\ &= \frac{1}{\sqrt{N}} \left(\sum_{x':f(x')=1} |x'\rangle + \sum_{x":f(x")=0} |x"\rangle \right) \\ &= \frac{1}{\sqrt{N}} \left(\frac{\sqrt{M}}{\sqrt{M}} \sum_{x':f(x')=1} |x'\rangle + \frac{\sqrt{N-M}}{\sqrt{N-M}} \sum_{x":f(x")=0} |x"\rangle \right) \end{aligned}$$

Let,

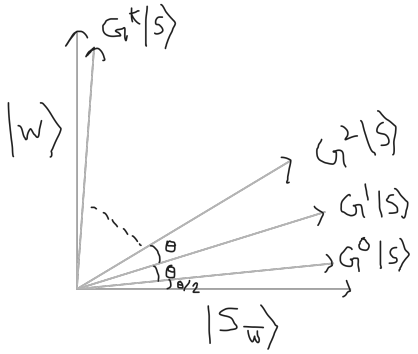
$$|w\rangle = \frac{1}{\sqrt{M}} \sum_{x':f(x')=1} |x'\rangle$$

$$|S_{\overline{w}}\rangle = \frac{1}{\sqrt{N-M}} \sum_{x":f(x")=0} |x"\rangle$$

Since $|w\rangle$ and $|S_{\overline{w}}\rangle$ are orthogonal they span the space S.
Therefore we can treat this space as 2 dimensional with basis $\{|w\rangle, |S_{\overline{w}}\rangle\}$.

$$\begin{aligned} \implies |S\rangle &= \sqrt{\frac{M}{N}}|w\rangle + \sqrt{\frac{N-M}{N}}|S_{\overline{w}}\rangle \\ &= sin\left(\frac{\theta}{2}\right)|w\rangle + cos\left(\frac{\theta}{2}\right)|S_{\overline{w}}\rangle \end{aligned}$$

Since $sin\left(\frac{\theta}{2}\right)$ is very small $\left(\sqrt{\frac{M}{N}}\right)$, the starting state has very large overlap with $|S_{\overline{w}}\rangle$.
Eventually we want $|S\rangle$ to have a large overlap with $|w\rangle$.
Geometrically, G rotates $|S\rangle$ towards $|w\rangle$.



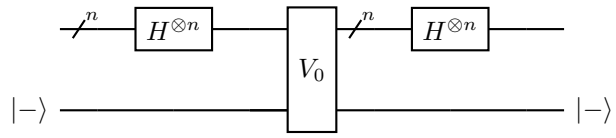
Since,

$$\left(\sqrt{\frac{M}{N}}\right)^2 + \left(\sqrt{\frac{N-M}{N}}\right)^2 = 1$$

After applying u_f:

$$|S\rangle \to -sin\left(\frac{\theta}{2}\right)|w\rangle + cos\left(\frac{\theta}{2}\right)|S_{\overline{w}}\rangle$$

The D gate:

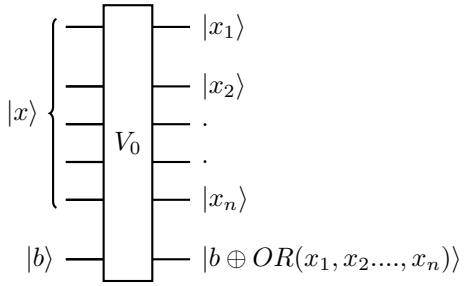


where V_0 performs a controlled phase shift i.e.
If $|x\rangle = |0^{\otimes n}\rangle$, no phase shift.
If $|x\rangle \neq |0^{\otimes n}\rangle$, flip the phase.
Here $|x\rangle$ is he input state to the gate V_0.

We can represent V_0 mathematically as:

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

$$\implies V_0\left|0^{\otimes n}\right\rangle = \left|0^{\otimes n}\right\rangle$$
$$\implies V_0\left|x\right\rangle = -\left|x\right\rangle \qquad \text{(for } |x\rangle \neq |0^{\otimes n}\rangle\text{)}$$



When $|b\rangle = |-\rangle$,

$$V_0 = (-1)^{OR(x_1,x_2,x_3,....,x_n)}\left|x\right\rangle$$

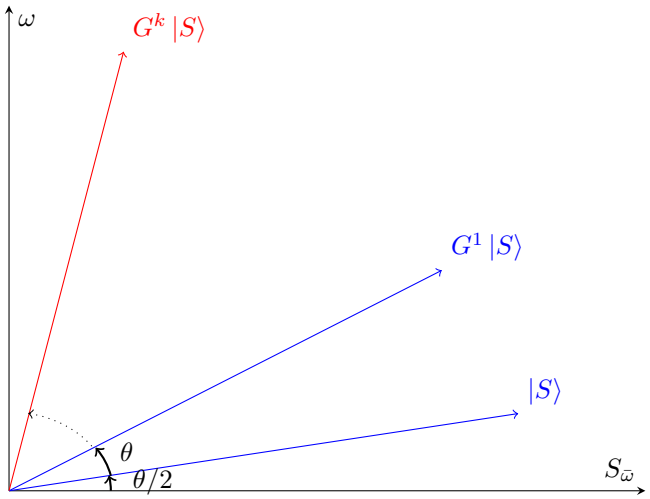
Now we have,

$$\begin{aligned}\implies D &= H^{\otimes n}V_0H^{\otimes n}\\ &= 2H^{\otimes n}\left|0^{\otimes n}\right\rangle\left\langle 0^{\otimes n}\right|H^{\otimes n} - H^{\otimes n}H^{\otimes n}\\ &= 2\left|S\right\rangle\left\langle S\right| - I\end{aligned}$$

Lecture 14 - Hardik Sharma

Recap

- Action of G, interpreted geometrically.



  That is, G *rotates* the initial state which has a lot of contribution from the non-solution states $S_{\bar{\omega}}$, towards a final state that has significant contribution from the solution superposition.

- $\mathcal{U}_f$ is just $\hat{\sigma}_z$ in the $\{\omega, S_{\bar{\omega}}\}$ basis. This is evident by the action of $\mathcal{U}_f$, on the elements of the basis, $\{\omega, S_{\bar{\omega}}\}$

  1. $\mathcal{U}_f\omega \to -\omega$
  2. $\mathcal{U}_f S_{\bar{\omega}} \to S_{\bar{\omega}}$

- The boolean function that V_0 computes is the *OR* of all the bits in the input. This is 0 only when all the input bits are themselves zero. This also gives :

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

  Since, V_0 flips the phase of all inputs that are not comprised of all-zeroes.

- Evaluating D.

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

This Class

We have seen previously that

$$|S\rangle = \sin\left(\frac{\theta}{2}\right)|\omega\rangle + \cos\left(\frac{\theta}{2}\right)|S_{\bar{\omega}}\rangle$$

Putting this value of $|S\rangle$ in the equation for D,

$$\begin{aligned} D &= 2\left(\sin\left(\frac{\theta}{2}\right)|\omega\rangle + \cos\left(\frac{\theta}{2}\right)|S_{\bar{\omega}}\rangle\right)\left(\sin\left(\frac{\theta}{2}\right)\langle\omega| + \cos\left(\frac{\theta}{2}\right)\langle S_{\omega}|\right) - \mathbb{I} \\ &= 2\left(\sin^2\left(\frac{\theta}{2}\right)\right)|\omega\rangle\langle\omega| + 2\left(\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\theta}{2}\right)\right)|\omega\rangle\langle S_{\bar{\omega}}| \\ &+ 2\left(\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right)\right)|S_{\bar{\omega}}\rangle\langle\omega| + 2\left(\cos^2\left(\frac{\theta}{2}\right)\right)|S_{\bar{\omega}}\rangle\langle S_{\bar{\omega}}| - \mathbb{I} \end{aligned}$$

Choosing the ordered basis to be $\{S_{\bar{\omega}}, \omega\}$, we have, in the matrix representation :

$$\begin{aligned} D &= \begin{pmatrix} 2\left(\cos^2\left(\frac{\theta}{2}\right)\right) - 1 & 2\left(\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right)\right) \\ 2\left(\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\theta}{2}\right)\right) & 2\left(\sin^2\left(\frac{\theta}{2}\right)\right) - 1 \end{pmatrix} \\ &= \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{pmatrix} \end{aligned}$$

Therefore, we have :

$$\begin{aligned} G &= D\mathcal{U}_f \\ &= \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \\ &= \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} = R_\theta \end{aligned}$$

Which is just the rotation matrix corresponding to an angle θ. We can also give an intutive reasoning for why this is a rotation matrix, we know that D is a reflection matrix, since it reflects about the initial State S, and so is $\mathcal{U}_f$, since it reflects about ω. And because product of two reflections, is a rotation, G is a rotation matrix.

Therefore, the entire algorithm boils down to rotating the quantum state in the $\{$*solution, non-solution*$\}$ basis over and over until we get close to a state that has large overlap of the solution superposition.

Therefore, $G^k = R_\theta^k = R_{k\theta}$

$$\boxed{G^k \left|S\right\rangle = R_{k\theta} \left|S\right\rangle = \sin\left(\left(\tfrac{2k+1}{2}\theta\right)\right)\left|\omega\right\rangle + \cos\left(\left(\tfrac{2k+1}{2}\theta\right)\right)\left|S_{\bar{\omega}}\right\rangle} \tag{1}$$

How to determine k

For the algorithm to work, we need to determine k, that is, after how many steps do we need to terminate, otherwise we rotate off to unsuitable states, where constribution of the solution superposition would decrease.

We know that in the final state, we want the *amplitude* of $|\omega\rangle$ to be as large as possible.

That is $\sin\left(\frac{2k+1}{2}\theta\right) \approx 1$

This implies,

$$\begin{aligned} \frac{2k+1}{2}\theta &\approx \frac{\pi}{2} \\ \theta = 2\sin^{-1}\left(\sqrt{\frac{M}{N}}\right) &\approx \frac{\pi}{2k+1} \end{aligned}$$

since we have that $M \ll N$, therefore, we can approximate $sin^{-1}(\theta)$ to θ.

Therefore,

$$\begin{aligned} 2\sqrt{\frac{M}{N}} &\approx \frac{\pi}{2k+1} \\ 2k+1 &\approx \frac{\pi\sqrt{N}}{2\sqrt{M}} \\ \therefore k &\approx \frac{\pi\sqrt{N}}{4\sqrt{M}} - \frac{1}{2} = O\left(\sqrt{\frac{N}{M}}\right) \end{aligned}$$

Therefore the number of queries we need are *subquadratic* to the case of classical algorithms

What about M?

We see that the number of queries we make needs to be optimal for the algorithm to works, otherwise, we might rotate more or less which would result in solutions with lesser contribution of the solution superposition.

But, the expression for the number of queries depends on M, which might be unknown.

There are a few strategies regarding this :

1. **Quantum Counting** We approximate the number of solutions at the begining of the algorithm using some specific constraints/observations.
2. **Randomized Quantum Search** We repeatedly apply the grover's algorithm for values of M from the set $\{2^0, 2^1, 2^2, ...\}$, it can be shown that with a high probability, a satisfactory solution will be found after $O\left(\sqrt{\frac{N}{M}}\right)$ queries.

What if M ≈ N ?

We see that $H^{\otimes n}$ acts as a quantum algorithm that transforms the initial state $|0^n\rangle$ to the state $|S\rangle = \sin\theta/2\,|\omega\rangle + \cos\theta/2\,|S_{\bar{\omega}}\rangle$, where the amplitude of the solution superposition is $\sqrt{p} = \sin\theta/2$. G^k transforms this amplitude to 1 in $O(\sqrt{\frac{1}{p}})$ queries.

But if $M \approx N$, then we need not apply the query box at all, we can just apply the *Hadamard* gate and measure the corresponding state.

Quantum Amplitude Amplification

The generalization of the above would be an algorithm as follows :

$$\mathcal{U}\,|0^n\rangle = \sqrt{p}\,|\psi_{good}\rangle + \sqrt{1-p}\,|\psi_{bad}\rangle$$

$$(\mathcal{A}^{\frac{1}{\sqrt{p}}})\mathcal{U}\,|0^n\rangle = \sqrt{p}\,|\psi_{good}\rangle + \sqrt{1-p}\,|\psi_{bad}\rangle$$

Other Models Of Quantum Computation

1. **Adiabatic Model**
   Consider two Hamiltonians, H_0 whose ground state is easily prepared and H_f whose ground state is hard to prepare.

   Then we deal with $H(s) = (1-s)H_0 + sH_f$, where s stores some description of time. At $s = 0$, the system is in ground state of H_0, which is easily prepared. We slowly change s to 1 and in the end we get ground state of H_f, which was hard to prepare directly.

   The time taken to change s from 0 to 1 should be kept according to the bound :

   $$T > \frac{1}{min_s(\Delta S)}$$

   Where ΔS is the difference between the energy of the states.

2. Other Popular Models :
   - Quantum Walks
   - Measurement Based Quantum Computing
   - Topological Quantum Computation

Some active areas of research/development :

- NISQ (Noisy Intermediate Sized Quantum) Machines
  Heuristic, Non-provable nature due to the noise.