

Quantum Computing, an Introduction with Grover's Search

Yan-Tong Lin

Advisor: Ming-Hsuan Kang

Department of Computer Science
National Chiao Tung University

Individual Study I, December 30, 2020

Outline

- 1 Introduction to Classical Computing
- 2 Introduction to Quantum Computing
- 3 Quantum Parallelism
- 4 Grover's Search
- 5 Concluding Remarks

Motivation

One might state the main goal of theoretical computer science as “study the power and limitations of the strongest-possible computational devices that Nature allows us.”

Since our current understanding of Nature is quantum mechanical, theoretical computer science should arguably be studying the power of quantum computers, not classical ones.

— Ronald de Wolf

- 1 Introduction to Classical Computing
 - Classical Circuit Model
- 2 Introduction to Quantum Computing
- 3 Quantum Parallelism
- 4 Grover's Search
- 5 Concluding Remarks

Classical Circuit Model

- Bits are used to store information
 - e.g. 0001 as 1, 0101 as 5
- Gates are used to manipulate them
 - e.g. *NAND* gate
 - one can show that *NAND* can achieve universal computation
- We model physical phenomenon to do these

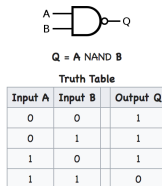


Figure: NAND gate and its truth table

- 1 Introduction to Classical Computing
- 2 Introduction to Quantum Computing
 - Postulates of Quantum Mechanics
 - Notations
 - Quantum Circuit Model
- 3 Quantum Parallelism
- 4 Grover's Search
- 5 Concluding Remarks

Postulates of Quantum Mechanics

In the past decades, physicists discover that Nature is "quantum".

Mathematical Formulation of the Postulates

- States of Systems¹ are Unit Vectors in Hilbert Spaces²
- Evolutions are linear transforms on the Hilbert space which map states to states.
- Measurements are Collections of Operators
- States of a Compositional System are Tensor Products of States of Component Systems

Remark

The postulates produces some interesting properties (superposition and entanglement)

¹To be more rigorous, closed systems

²complete inner product space, e.g. \mathbb{C}^n

Notations

- In quantum physics, we often use $|\phi\rangle$ to denote column vector
- and use $\langle\phi|$ to denote $|\phi\rangle^{\dagger 3}$
- $\langle\phi|\psi\rangle$ is inner product of $|\phi\rangle$ and $|\psi\rangle$

³ \dagger means conjugate transpose

Quantum Circuit Model

- Qubit
- Quantum Gate
- Oracle

Qubit

- States of Systems are Unit Vectors in Hilbert Spaces
- qubit: $|0\rangle, |1\rangle$
- superposition: $\alpha|0\rangle + \beta|1\rangle, |\alpha|^2 + |\beta|^2 = 1$

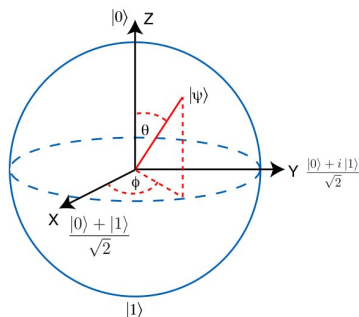


Figure: Bloch Sphere: $|\psi\rangle = e^{i\delta}(\cos \frac{\theta}{2}|0\rangle + e^{i\phi} \sin \frac{\theta}{2}|1\rangle)$









Multiple Qubits

- States of a Compositional System are Tensor Products of States of Component Systems
- $|00\rangle := |0\rangle \otimes |0\rangle$
- $|00\rangle, |01\rangle, |10\rangle, |11\rangle$
- n qubit system has 2^n bases

- Evolutions are linear transforms on the Hilbert space which map states to states.
- Norm-Preserving + Linear \implies Unitary
- Functions on one qubit: Members of $SU(2) \cong SO(3)$
- Like in classical computation, it can be shown that there exists finite universal gate sets⁴

⁴ETHZ slide

Quantum Gates — Examples

Operator	Gate(s)		Matrix
Pauli-X (X)			$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)			$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)			$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)			$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)			$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)			$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)			$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

- O_f : quantum version of a given function f
- Requirements
 - Unitary
 - Can be efficiently made up by some finite gate set (X is like *NOT*)
- Example
 - $O_f(|x\rangle) = |f(x)\rangle$
 - $O_f(|x, y\rangle) = |x, y \oplus f(x)\rangle$

- 1 Introduction to Classical Computing
- 2 Introduction to Quantum Computing
- 3 Quantum Parallelism**
 - Deutsch-Jozsa Algorithm
- 4 Grover's Search
- 5 Concluding Remarks

A Taste of Power: Quantum Parallelism

Problem

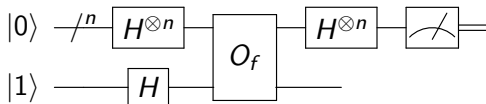
$f : \{0, 1\}^n \rightarrow \{0, 1\}$ is either balanced or constant. Given f , O_f as black boxes, judge if f is constant.

- It requires $O(2^n)$ queries of f for a classical circuit to judge
- quantum: $O(1)$ query

Deutsch-Jozsa Algorithm I

Lemma

$$H^{\otimes n}|z\rangle = \frac{1}{\sqrt{2^n}} \sum_y (-1)^{y \cdot z} |y\rangle$$



Deutsch-Jozsa Algorithm II

$$|\psi_0\rangle = |0\rangle^{\otimes n} |1\rangle$$

$$|\psi_1\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle (|0\rangle - |1\rangle)$$

$$|\psi_2\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle (|f(x)\rangle - |1 \oplus f(x)\rangle)$$

$$= \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle (|0\rangle - |1\rangle)$$

$$|\psi_3\rangle = \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \left[\sum_{y=0}^{2^n-1} (-1)^{x \cdot y} |y\rangle \right]$$

$$= \frac{1}{2^n} \sum_{y=0}^{2^n-1} \left[\sum_{x=0}^{2^n-1} (-1)^{f(x)} (-1)^{x \cdot y} \right] |y\rangle$$

Deutsch-Jozsa Algorithm III

Now consider $|y\rangle = |0\rangle$

$$\frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} (-1)^{x \cdot 0} |0\rangle$$

Question

Why does this work? interference!

- 1 Introduction to Classical Computing
- 2 Introduction to Quantum Computing
- 3 Quantum Parallelism
- 4 Grover's Search
 - Problem Definition
 - the Good/Bad basis
 - Geometric Intuition
 - Complexity
 - Implementation
- 5 Concluding Remarks

Unstructured Search

Given a set X of N items and a function $f : X \rightarrow \{0, 1\}$, suppose there are $M = \epsilon N$ out of N items satisfy $f(x) = 1$. Find an instance $x \in S$ such that $f(x) = 1$.

- For classical algorithms, $O(N)$ -time is required to get an instance with high probability.
- For quantum algorithms, assume an oracle U_ω is given such that

$$U_\omega |x\rangle = (-1)^{f(x)} |x\rangle.$$

With Grover's Search, $O(\sqrt{N})$ -time is suffice.

the Good/Bad basis

Represent uniform state vector by l.c. of uniform "good" and uniform "bad" vectors.

$$|s\rangle = \sqrt{\frac{1}{N}} \sum_x |x\rangle$$

$$|\omega\rangle = \sqrt{\frac{M}{N}} \sum_{x \in f^{-1}(1)} |x\rangle$$

$$|s'\rangle = \sqrt{\frac{N-M}{N}} \sum_{x \notin f^{-1}(1)} |x\rangle$$

$$\text{then } |s\rangle = \sin \frac{\theta}{2} |\omega\rangle + \cos \frac{\theta}{2} |s'\rangle \text{ where } \sin \frac{\theta}{2} = \sqrt{\frac{M}{N}}$$

Geometric Intuition

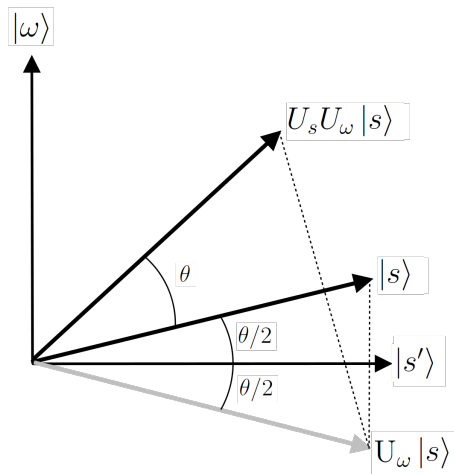


Figure: One step of Grover's Iteration: Rotation by θ

We want to stop when the state vector passes close to $|\omega\rangle$.

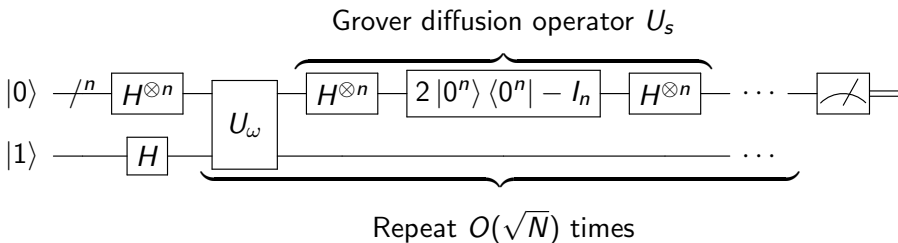
Suppose Grover's iteration is performed r times, the probability of success is exactly $\sin^2\left(\left(r + \frac{1}{2}\right)\theta\right)$.

$r \approx \pi\sqrt{N}/4$ is the first r we can get a high probability.

Implementation

$$U_\omega = I - 2|\omega\rangle\langle\omega|$$

$$U_s = 2|s\rangle\langle s| - I$$



- 1 Introduction to Classical Computing
- 2 Introduction to Quantum Computing
- 3 Quantum Parallelism
- 4 Grover's Search
- 5 Concluding Remarks**

Concluding Remarks

- Amazing algorithms like Shor's factoring algorithm
- Quantum Fourier Transformation and Quantum Phase Estimation
- Quantum Walks, the generalization of Grover's Search
- QAOA for solving CSP problems