Chapter 10

Grover's algorithm

In this chapter, we present the search algorithm given by Grover, which brings us a quadratic speed up when compared to classical algorithms. The aim is to determine whether a given element is in a given list. But, we present the problem in a general form. This chapter is written by modifying the corresponding chapters given in the lecture notes of John Watrous:

$$https://cs.uwaterloo.ca/\ watrous/LectureNotes.html$$

Suppose that we are given a function

$$f : \{0,1\}^n \to \{0,1\}.$$

Form the previous section, we know that we can design a reversible gate B_f for any such function that behaves as follows:

$$B_f|x\rangle|a\rangle = |x\rangle|a \oplus f(x)\rangle,$$

where $x \in \{0,1\}^n$ is any input and $a \in \{0,1\}$ is the ancilla qubit. Then, we can have a circuit using B_f. The generic problem here is to find an input making the value of the function f 1 or to say that there is no such an input. Classically, we need to test all possible 2^n inputs. Probabilistically, a best strategy can be picked some possible inputs randomly and then test them. But, we still need to test $\Omega(2^n)$ inputs for any fixed error bound. Each test can be seen as a query to the black-box and so we can define it as a complexity measure here. Remark that the quantum query complexity is a fundamental complexity measure among the scientist working on quantum algorithms.

The classical query complexity of our problem is $\Omega(2^n)$. We show that the quantum query complexity of the problem is $O(\sqrt{2^n})$. The main difference from the classical case is that we can query all possible inputs at once in a superposition, but, it is not sufficient to give the correct answer with high

probability. However, we can separate bad and good inputs by using some quantum tricks and then cleverly amplify the amplitudes of the good inputs, which uses $O(\sqrt{2^n})$ queries.

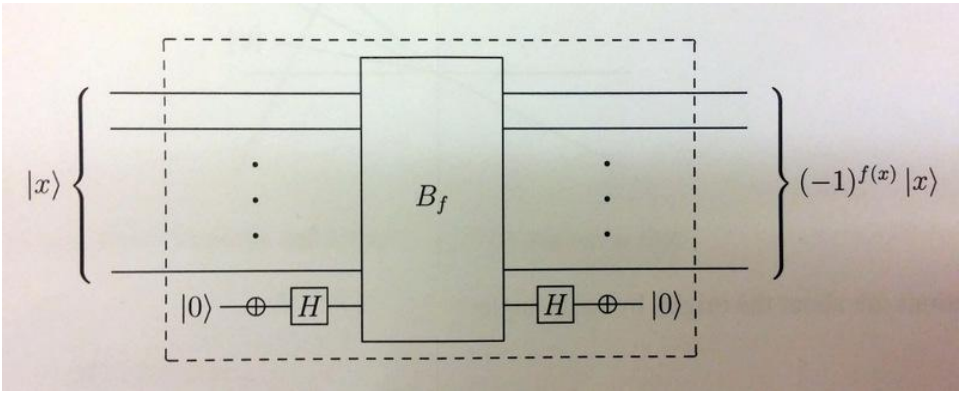
The description of Grover's search algorithm is very simple: after the initialization, we apply a combination unitary operators (a combination of two reflections leading to a rotation) for k times and then make a measurement. Before giving the description of the algorithm, we define the unitary operators in this combinations, operating on n qubits:

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

and, a special case of Z_f,

$$Z_0 = \begin{cases} -|x\rangle & \text{if } x = 0^n \\ |x\rangle & \text{if } x \neq 0^n \end{cases},$$

where $x \in \{0,1\}^n$. From the previous part, we know how to define such unitary matrices by help of phase kick-back effect that uses the black-box B_f and an ancilla qubit. You can find the diagram below:



Remark that Z_f uses a single query to B_f. The transformation Z_0 is a special case of Z_f, where the function returns 1 only if the input is 0^n. With the ancilla qubit, the black B_0 makes the following transformation:

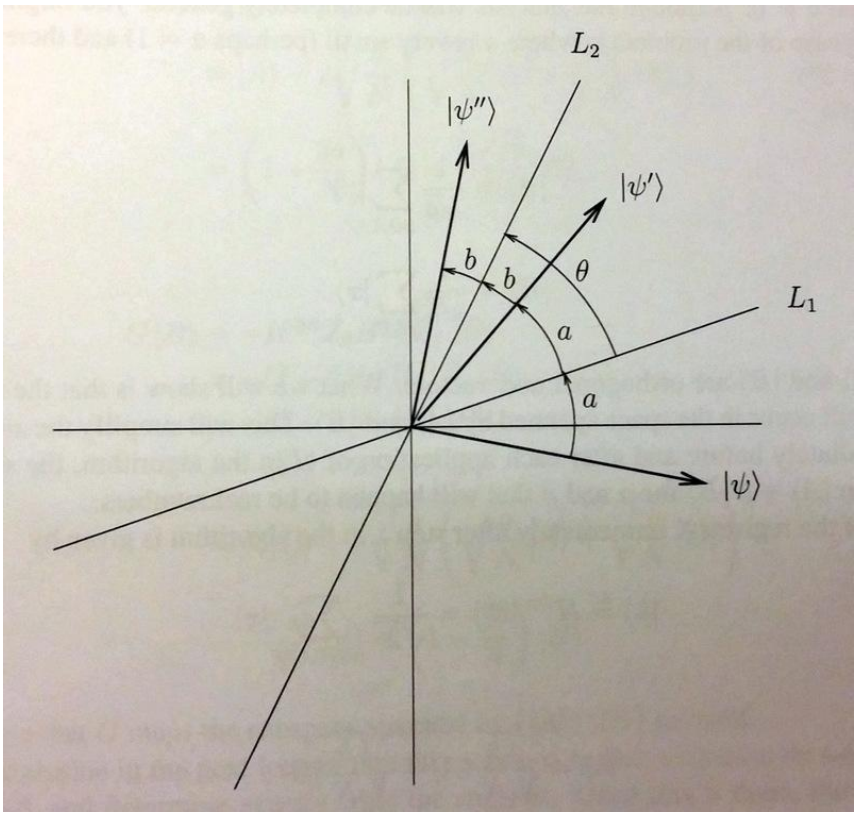
$$|x\rangle|a\rangle \to |x\rangle|a \otimes (\neg x_1 \wedge \cdots \wedge \neg x_n)\rangle.$$

When we replace B_f with B_0 in the above diagram, we obtain the circuit diagram for Z_0. Note that Z_0 does not need to use any query to B_f.

THE ALGORITHM

The input: n-qubit quantum register (X), initialized to $|0^n\rangle$
The output1: $x \in \{0,1\}^n$ satisfying $f(x) = 1$ or
The output2: an arbitrary x if f is constant, returning 0
Apply $H^{\otimes n}$
Repeat k times
    Apply $G = -H^{\otimes n} Z_0 H^{\otimes n} Z_f$ on X
Measure X and output the result

Here is a graphical representation of how two reflections can implement a rotation.



In the algorithm, $H^{\otimes n} Z_0 H^{\otimes n}$ and $-Z_f$ are kinds of reflections and so their combination $G = -H^{\otimes n} Z_0 H^{\otimes n} Z_f$ is a rotation by a particular angle. Before, representing their affects, we define the bad and good input(s):

$$A = \{x \in \{0,1\}^n \mid f(x) = 1\}$$
$$B = \{x \in \{0,1\}^n \mid f(x) = 0\}$$

Let their sizes be $a = |A|$ and $b = |B|$, respectively. If $a = 0$ or $b = 0$, the algorithm always returns a correct answer exactly. So, we assume that $a \neq b$ in the remaining part. We also define a two orthogonal vectors based on the sets A and B:

$$|A\rangle = \frac{1}{\sqrt{a}} \sum_{x \in A} |x\rangle$$
$$|B\rangle = \frac{1}{\sqrt{b}} \sum_{x \in B} |x\rangle$$

Now, we trace the computation. Let $N = 2^n$. At the beginning of the computation, we create a superposition of all inputs with equal amplitude:

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

which can be rewritten as

$$|h\rangle = \sqrt{\frac{a}{N}}|A\rangle + \sqrt{\frac{b}{N}}|B\rangle.$$

We trace the computation by omitting the ancilla qubit. The reader can verify that when implementing Z_f and Z_0, the ancilla qubit starts in state $|0\rangle$ and then returns again to $|0\rangle$ for the next usage. The matrix form of Z_0 is

$$Z_0 = \begin{pmatrix} -1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix},$$

which can be written as

$$Z_0 = I - 2|0^n\rangle\langle 0^n|,$$

where $|0^n\rangle\langle 0^n|$ is a zero matrix expect the first diagonal entry, which is 1. By using this representation, we can obtain that

$$H^{\otimes n} Z_0 H^{\otimes n} = H^{\otimes n}(I - 2|0^n\rangle\langle 0^n|)H^{\otimes n},$$

and, since $H^\dagger = H = H^{-1}$, we gets

$$H^{\otimes n} I H^{\otimes n} - 2H^{\otimes n}|0\rangle\langle 0|H^{\otimes n} = I - 2|h\rangle\langle h|.$$

Note that, as defined above $|h\rangle = H^{\otimes n}|0\rangle$, and so $\langle h| = \langle 0|(H^{\otimes n})^\dagger = \langle 0|H_n$.

Now, we see the affect of G on $|A\rangle$ and $|B\rangle$:

$$\begin{aligned}
G|A\rangle &= -H^{\otimes n} Z_0 H^{\otimes n} Z_f |A\rangle \\
&= (I - 2|h\rangle\langle h|)(-Z_f)|A\rangle \\
&= (I - 2|h\rangle\langle h|)|A\rangle \\
&= |A\rangle - 2|h\rangle\langle h||A\rangle \\
&= |A\rangle - 2|h\rangle \left(\sqrt{\frac{a}{N}}\langle A| + \sqrt{\frac{b}{N}}\langle B|\right)|A\rangle \\
&= |A\rangle - 2|h\rangle\sqrt{\frac{a}{N}} \\
&= |A\rangle - 2\sqrt{\frac{a}{N}}\left(\sqrt{\frac{a}{N}}|A\rangle + \sqrt{\frac{b}{N}}|B\rangle\right) \\
&= \left(1 - \frac{2a}{N}\right)|A\rangle - \frac{2\sqrt{ab}}{N}|B\rangle
\end{aligned}$$

and

$$\begin{aligned}
G|A\rangle &= -H^{\otimes n} Z_0 H^{\otimes n} Z_f |B\rangle \\
&= -(I - 2|h\rangle\langle h|)(Z_f)|B\rangle \\
&= -(I - 2|h\rangle\langle h|)|B\rangle \\
&= -|B\rangle + 2|h\rangle\langle h||B\rangle \\
&= -|B\rangle + 2|h\rangle \left(\sqrt{\frac{a}{N}}\langle A| + \sqrt{\frac{b}{N}}\langle B|\right)|B\rangle \\
&= -|B\rangle + 2|h\rangle\sqrt{\frac{b}{N}} \\
&= -|B\rangle - 2\sqrt{\frac{b}{N}}\left(\sqrt{\frac{a}{N}}|A\rangle + \sqrt{\frac{b}{N}}|B\rangle\right) \\
&= \frac{2\sqrt{ab}}{N}|A\rangle - \left(1 - \frac{2b}{N}\right)|B\rangle.
\end{aligned}$$

We end up as a linear combination of $|A\rangle$ and $|B\rangle$. Therefore, we can say that G maps a vector in the space spanned by $\{|A\rangle, |B\rangle\}$ to a vector in the

same space:

$$\begin{aligned} G|A\rangle &= \frac{b-a}{N}|A\rangle - \frac{2\sqrt{ab}}{N}|B\rangle \\ G|B\rangle &= \frac{2\sqrt{ab}}{N}|A\rangle - \frac{a-b}{N}|B\rangle. \end{aligned}$$

We represent G as a single matrix partitioned based on the elements by A's and B's. Suppose that the first part is for B and the second part is for A:

$$M = \begin{pmatrix} \frac{b-a}{N} & -\frac{2ab}{N} \\ \\ \frac{2ab}{N} & \frac{b-a}{N} \end{pmatrix}.$$

Matrix M is a rotation matrix since it is indeed a square of the rotation matrix

$$R_\theta = \begin{pmatrix} \sqrt{\frac{b}{N}} & -\sqrt{\frac{a}{N}} \\ \\ \sqrt{\frac{a}{N}} & \sqrt{\frac{b}{N}} \end{pmatrix},$$

with angle $\theta \in (0, \pi/2)$ satisfying $\sin\theta = \sqrt{\frac{a}{N}}$ and $\cos\theta = \sqrt{\frac{b}{N}}$:

$$R_\theta = \begin{pmatrix} \cos\theta & -sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}.$$

The equality $M = R_\theta^2 = R_{2\theta}$ can be followed as

$$R_\theta^2 = \begin{pmatrix} \sqrt{\frac{b}{N}} & -\sqrt{\frac{a}{N}} \\ \\ \sqrt{\frac{a}{N}} & \sqrt{\frac{b}{N}} \end{pmatrix} \begin{pmatrix} \sqrt{\frac{b}{N}} & -\sqrt{\frac{a}{N}} \\ \\ \sqrt{\frac{a}{N}} & \sqrt{\frac{b}{N}} \end{pmatrix} = \begin{pmatrix} \frac{b-a}{N} & -\frac{2ab}{N} \\ \\ \frac{2ab}{N} & \frac{b-a}{N} \end{pmatrix} = M = R_{2\theta}.$$

Then, we can say that G makes an rotation by the angle 2θ in the space spanned by $\{|A\rangle, |B\rangle\}$. Then, we can rewrite the initial state $|h\rangle$ as

$$|h\rangle = \sqrt{\frac{b}{N}}|B\rangle + \sqrt{\frac{a}{N}}|A\rangle = \cos\theta|B\rangle + \sin\theta|A\rangle.$$

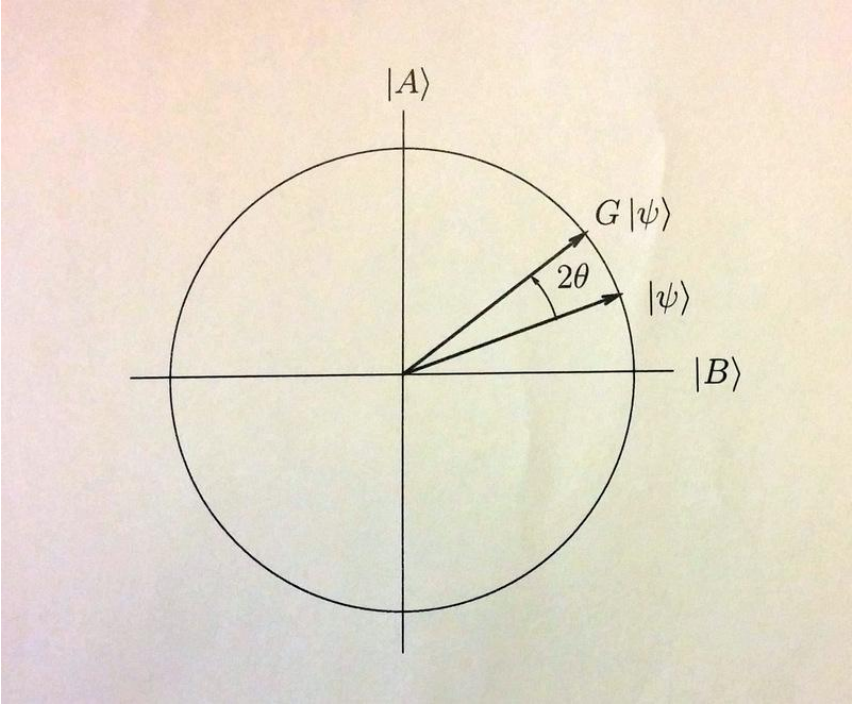
So after a single iteration of G, we obtain the new state as

$$\cos(3\theta)|B\rangle + \sin(3\theta)|A\rangle,$$

and so, after the k iterations, the new state will be

$$\cos((2k+1)\theta)|B\rangle + \sin((2k+1)\theta)|A\rangle.$$

You can see a single rotation implemented by G on a state $|\psi\rangle$:



Now, we can determine the value of k. Any k satisfying

$$\sin((2k+1)\theta) \approx 1$$

leads us to obtain a quantum algorithm giving a correct answer with high probability. Then, the value of k can be determined by

$$(2k+1)\theta \approx \frac{\pi}{2} \Rightarrow k \approx \frac{\pi}{4\theta} - \frac{1}{2}.$$

If there is a single good input, $a = 1$, then we can calculate k as follows:

$$\sin\theta = \sqrt{\frac{a}{N}} \Rightarrow \sin\theta = \sqrt{\frac{1}{N}},$$

and so

$$\theta = \sin^{-1}\sqrt{\frac{1}{N}} \approx \frac{1}{\sqrt{N}}.$$

We can pick k as

$$k = \left\lfloor \frac{\pi}{4}\sqrt{N} \right\rfloor,$$

which would be our quantum query complexity. Then, the accepting probability will be

$$\sin^2((2k+1)\theta).$$

If we compute the possible accepting probabilities for $N = 2, 4, 8, \ldots, 2^n, \ldots$, we will see that the accepting probability takes the values respectively at least

$$0.5, 1.0, 0.94, 0.96, 0.99, 0.99, \ldots.$$

We can compute k and follow the related analyses for other possible values of a's. But, as a programmer, we usually do not know the value of a in advance. We can use many different strategies. One proposed strategy is given below:

Set $m = 1$
Repeat
    Choose $k \in \{1, \ldots, m+1\}$ uniformly
    Execute Grover's algorithm
    Stop the computation if the found x satisfies $f(x) = 1$
    Set $m = \lfloor (8/7)m \rfloor$
Until $m > \sqrt{N}$
Output that f is a constant function, equal to 0.

This strategy finds a good input a with a probability at least $\frac{1}{4}$ by using

$$O\left(\sqrt{\frac{N}{a}}\right)$$

queries, where the success probability can be amplified by using this strategy a few more times.