CS 593/MA 592 - Intro to Quantum Computing
Spring 2024
Tuesday, February 13 - Lecture 6.1

Today's scribe: Mohamed Eltohfa

Reading:
Agenda:

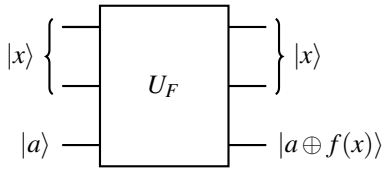
1. Phase kickback
2. Unstructured search
3. Grover's algorithm
4. Multiple solutions (covered in more detail in next lecture)

1 Phase Kickback

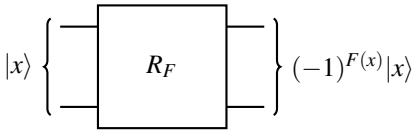
Previously, we mentioned that a Boolean function

$$F : \{0,1\}^n \to \{0,1\}$$

can be encoded into a unitary in two ways: a dilated gate that uses an ancilla qubit to encode F into a permutation of the computational basis vectors:



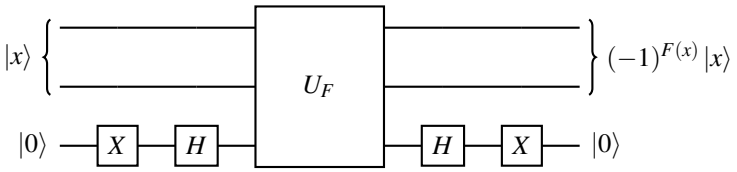
or a "diagonal" gate that does not use an ancilla qubit and encodes $F(x)$ in the *phase*



Claim. *Over some universal gate set, circuits with oracle access to U_F can simulate R_F and vice versa.*

Proof. We'll just do one direction. (We'll do the other one later, after we've discussed phase estimation.)

To simulte R_F using U_F, simply use the following circuit C (compare to Deutsch-Jozsa algorithm)



This simulates R_f, in the sense that when the ancilla is set of $|0\rangle$, we get

$$C|x_1,\cdots,x_n,0\rangle = (-1)^{F(x)}|x_1,\cdots,x_n,0\rangle = R_F|x_1,\cdots,x_n\rangle \otimes |0\rangle.$$

□

Take-away: quantum access to U_F is as sensible as quantum access to R_F (in practice, both are unrealistic, at least at this level of generality!).

2 Unstructured Search

Suppose we are given oracle access to

$$F : \{0,1\}^n \to \{0,1\} = \{NO, YES\},$$

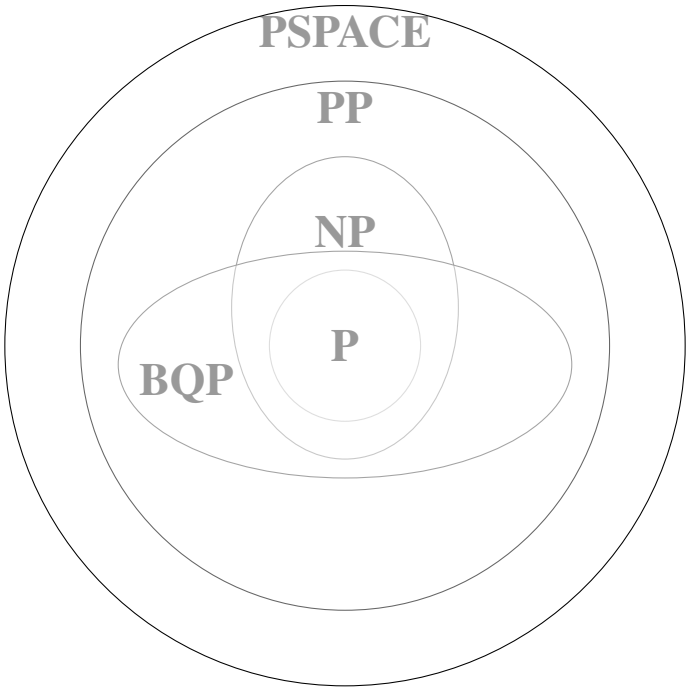
for example,

$$F : \{\text{possible passwords for a specific username}\} \to \{0,1\} = \{WRONG, CORRECT\}$$

which has $\#F^{-1}(1) = 1$, i.e., only one correct password. Q: How many calls to F must we make to guarantee that we find x, the correct password? A: if $\#F^{-1}(1) = 1$, then it's not too hard to show that 2^n calls are necessary classically (at leaset deterministically; probabilistically should be the same). On the other hand, as we shall see shortly, Grover's algorithm shows that at most $2^{n/2} = \sqrt{2^n}$ calls suffice if we have quantum access to F. (That means that quantum password checking functions are easier to crack, so don't use them, just use classical password checking functions!).

For example, F could come from a Boolean circuit, where we know that CSAT is NP-complete. The Strong Exponential Time Hypothesis (SETH) says that any NP-complete problem is expected to have running time 2^n, where n is the length of an instance. (In particular, SETH implies that P $\neq$ NP). The SETH is remarkable because it says solving NP-complete problems is no easier than unstructured search. Grover's algorithm "violates" the SETH (except not really, because quantum oracle access vs classical oracle access is an apples vs oranges comparison). So, despite "violating" SETH, Grover's algorithm CANNOT give a polynomial time quantum algorithm for NP-complete problems. Note if there are m inputs such that $F(x) = 1$, then Grover's algorithm only requires $\mathcal{O}(2^{n/2}/m)$ calls to the oracle, and in fact, this is optimal, which we show in the next lecture.

For what it's worth, the mainstream ideology of CS theory is that the following picture is correct, with all of the classes distinct:



3 Grover's Algorithm

Suppose we have an F with only one winner w (i.e., $F(w) = 1$ for a unique w) and oracle access to F through R_F (see circuit above). How can we use R_F to find w? We might guess that we should feed (like we usually do) the equal superposition

$$|\psi\rangle = \frac{1}{2^{n/2}} \sum_{x:=0}^{2^n-1} |x\rangle$$

(which of course, we can prepare using $H^{\otimes n}$) into R_F to get the state:

$$R_F |\psi\rangle = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} (-1)^{F(x)} |x\rangle = -\frac{|w\rangle}{2^{n/2}} + \frac{1}{2^{n/2}} \sum_{x \neq w} |x\rangle .$$

But what did R_f do to $|\psi\rangle$?

Let us define the (unnormalized) state

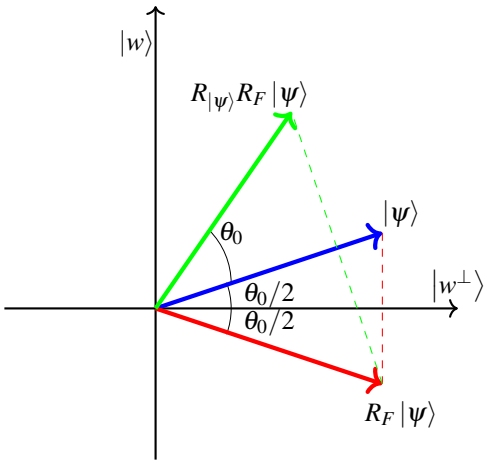
$$\left|w^{\perp}\right\rangle = \frac{1}{2^{n/2}} \sum_{x \neq w} |x\rangle ,$$

which is orthogonal to $|w\rangle$.

Geometrically, R_F reflects $|\psi\rangle$ across the $w^{\perp}$ axis in the "$w, w^{\perp}$" plane, as shown in the figure below.

This does not "help"! But at the very least, it did not hurt. (Note even though in the picture R_F seems to make $|\psi\rangle$ even further from w, considered as quantum states (which are only distinct up to global phases), $R_F |\psi\rangle$ is exactly as close to $|w\rangle$ as $|\psi\rangle$.)

But maybe we can do something else to $R_F |\psi\rangle$ to move it closer to $|w\rangle$?... What if we reflect it across $|\psi\rangle$ now? Call this operation $R_{|\psi\rangle}$.



You might notice in the figure that we are assuming $|\psi\rangle$ is closer to $|w^\perp\rangle$ than to $|w\rangle$. Well, this is correct as long as $n > 1$, because there is only one w with $f(w) = 1$, but ψ is the equal superposition of all $x \in \{0,1\}^n$.[1] In particular, if we apply $R_{|\psi\rangle}R_F$ to $|\psi\rangle$, we move *closer* to $|w\rangle$ by angle θ_0. We call $R_{|\psi\rangle}R_F$ the *Grover iteration* or *Grover operator*. The basic idea of Grover's algorithm is that if we apply $R_{|\psi\rangle}R_F$ the "correct" number of times—let's call this number r—then we can move $|\psi\rangle$ to be very close to $|w\rangle$.[2]

To make this work, we need to understand two things:

1. That we can implement $R_{|\psi\rangle}$ with a quantum circuit, and
2. What r is, which will essentially amount to understanding all the angles in the figure.

Let's address the first point first. Note that $R_{|\psi\rangle}$ has nothing to do with either R_F or $|w\rangle$. In the $w, w^\perp$ plane, we can describe $R_{|\psi\rangle}$ by:

$$R_{|\psi\rangle} = 2|\psi\rangle\langle\psi| - \mathbb{I} = 2H^{\otimes n}|0\rangle\langle 0|H^{\otimes n} - \mathbb{I} = 2H^{\otimes n}|0\rangle\langle 0|H^{\otimes n} - H^{\otimes n}\mathbb{I}H^{\otimes n} = H^{\otimes n}(2|0\rangle\langle 0| - \mathbb{I})H^{\otimes n}$$

We know how to implement $H^{\otimes n}$, so it remains to show how to implement $R_g := 2|0\rangle\langle 0| - \mathbb{I}$ with a quantum circuit. As a sanity check, note that R_g is in fact a unitary on all of $(\mathbb{C}^2)^{\otimes n}$; in fact, R_g is exactly the "phase oracle" for the Boolean function g where

$$g : \{0,1\}^n \to \{0,1\}$$
$$0 \mapsto 0$$
$$0 \neq x \mapsto 1$$

In other words, $g(x_1, \cdots, x_n) = OR(x_1, \cdots, x_n) = OR(x_1, OR(x_2, \cdots, x_n))$ is a big OR of all its inputs. Obviously we can implement this using a classical Boolean circuit with $n-1$ many standard 2-ary OR gates. Thus: R_g can be implemented with a quantum circuit of size poly(n).

Now let's address the second point, which is arguably more interesting. From the figure, θ_0 is twice the angle between $|\psi\rangle$ and $|w^\perp\rangle$. We can use the inner product to figure out this angle. In general,

$$\cos\angle(u,v) = \frac{|\langle u|v\rangle|}{\|u\|\,\|v\|}.$$

For our case,

$$\theta_0 = 2\arccos\left(\frac{|\langle\psi|w^\perp\rangle|}{\|\psi\|\,\|w^\perp\|}\right).$$

[1] The angle $\theta_0/2$ in the figure is exaggerated for clarity but it's actually very small. Then again, it is not *too* small, and this pedantic sounding point, in fact, is exactly why Grover's algoirthm works as well as it does!

[2] Note that we cannot just pick r "really" big like $r = 2^n$, since then we will rotate $|\psi\rangle$ too far!

We have $\|\psi\| = 1$,

$$\left\|w^{\perp}\right\| = \sqrt{\left\langle w^{\perp} \middle| w^{\perp} \right\rangle} = \sqrt{\frac{2^n - 1}{2^n}} = \sqrt{1 - \frac{1}{2^n}}$$

, and $|\left\langle \psi \middle| w^{\perp} \right\rangle| = \frac{2^n - 1}{2^n}$. So,

$$\theta_0 = 2\arccos(\sqrt{1 - \frac{1}{2^n}}),$$

from which we get (using trigonometry)

$$\sin^2(\theta_0/2) = \frac{1}{2^n}.$$

Now using the Taylor expansion for arcsin

$$\theta_0 = 2\arcsin(\sqrt{\frac{1}{2^n}}) = 2\sqrt{\frac{1}{2^n}} + \mathcal{O}(2^{-3n/2}) = \Theta(\frac{1}{2^{n/2}})$$

On the other hand, the angle between $|\psi\rangle$ and $|w\rangle$ is essentially 90° (but not exactly, otherwise the Grover iteration would not make any progress)! Angle $\angle(|\psi\rangle, |w\rangle)$ satisfies

$$\angle(|\psi\rangle, |w\rangle) = \frac{\pi}{2} - \frac{\theta_0}{2} = \frac{\pi}{2} - \Theta(2^{-n/2})$$

Each application of $R_{|\psi\rangle} R_F$ rotates the plane by θ_0. So, if we do this operation exactly $r \sim 2^{n/2}$ times, the angle between $(R_{|\psi\rangle} R_F)^r |\psi\rangle$ and $|w\rangle$ will be within $2^{-n/2}$ of 0.

Therefore, if we measure $(R_{|\psi\rangle} R_F)^r |\psi\rangle$ in the computational basis, then with probability greater than $2/3$ (there are some minor fudge factors here), we will get outcome w. Since each application of $R_{|\psi\rangle} R_F$ requires exactly one call of R_F and $r \sim 2^{-n/2}$, this is exactly what we wanted to show!