Topics in Theoretical Computer Science May 9, 2016

Lecture 11 (Notes)

Lecturer: Ola Svensson *Scribes: Ola Svensson*

Disclaimer: These notes were written for the lecturer only and may contain inconsistent notation, typos, and they do not cite relevant works. They also contain extracts from the two main inspirations of this course:

1. The book *Computational Complexity: A Modern Approach* by Sanjeev Arora and Boaz Barak;
2. The course *http://theory.stanford.edu/ trevisan/cs254-14/index.html* by Luca Trevisan.

1 Introduction

Recall last lecture:

- (Gentle) introduction to quantum computing
- Two-slit experiment;
- Qubits, superposition, simple operations (rotations, reflections).
- The EPR paradox (the parity game)

Today:

- Definition of Quantum Computation and **BQP**
- Relation to classic computing (**BPP** $\subseteq$ **BQP**)
- Grover's search algorithm

First some comments on Homework III:

- The response of the prover in an IP protocol may be adaptive so one cannot write down all the answers to all possible questions in PSPACE.
- In a public coin protocol the prover does not see all random bits of the verifier at once ("fresh" random bits are revealed in each round).
- Be mathematically correct; better to write true things than "random things". Please come and talk to me or the TAs if you are uncertain about things.

2 Quantum Computing and BQP

Recall that:

- The "memory" of a quantum computer, called a *quantum register*, consists of m qubits.
- Its state is a *superposition* of all 2^m basic states:

$$\text{a vector } \mathbf{v} = \langle v_{0^m}, v_{0^{m-1}1}, \ldots, v_{1^m} \rangle \in \mathbb{C}^{2^m}, \text{ where } \sum_x |v_x|^2 = 1.$$

  Here, v_x denotes the "amplitude" of the basic state $|x\rangle$.

- When *measuring* the register (i.e., reading its value), we will obtain the value x with probability $|v_x|^2$. Furthermore, this operation will *collapse* the state of the register to the vector $|x\rangle$.

2.1 Quantum operations

What kind of operations can we allow on a quantum state?

- It needs to preserve that $\sum_x |v_x|^2 = 1$. In other words, the operation should map unit vectors to unit vectors.
- It turns out that the operation should be a linear function.

From these two conditions, we get the following definition.

Definition 1 *A quantum operation for an m-qubit register that maps its previous state to the new state can be described by a unitary matrix $U \in \mathbb{C}^{2^m \times 2^m}$.*

Recall that a unitary matrix is a matrix U satisfying $UU^* = I$ where U^* denotes the conjugate transpose of U. That is $U^*_{x,y} = \bar{U}_{y,x}$. A matrix is unitary iff the columns (or the rows) form an orthonormal basis of the space $\mathbb{C}^{2^m}$.

The following facts are immediate from the definition of a quantum operation:

Composability: If A_1, A_2 are unitary matrices (i.e., quantum operations) then A_1A_2 is also unitary (a quantum operation).

Invertability: Every quantum operation A_1 has a corresponding "inverse" operation A_1^* that cancels it. So quantum computing is reversible. Is classical computing reversible?

2.1.1 Some examples of quantum operations

Flipping bits: If we wish to "flip" the first qubit in an m-qubit regiester, i.e., apply the NOT operation on the first qubit, then this can be done as a quantum operation that maps the basis state $|b, x\rangle$ to the basis state $|1-b, x\rangle$. The matrix of this operation is a permutation on the standard basis, and permutation matrices are always unitary.

Note on notation: The above operation only affects the first qubit, so the remaining qubits in x are unaffected. To aboid clutter, we will often drop the unaffected qubits from the notation. Thus the NOT operation can be described as $|0\rangle \mapsto |1\rangle$ and $|1\rangle \mapsto |0\rangle$.

Rotation on single qubit Rotating a single qubit by angle Θ (as we did in the last lecture) corresponds to the operation

$$|0\rangle \mapsto \cos\theta|0\rangle + \sin\theta|1\rangle, \text{ and } |1\rangle \mapsto -\sin(\theta)|0\rangle + \cos\theta|1\rangle.$$

In matrix form, this is described as

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix},$$

which is unitary.

The Hadamard operation The Hadamard gate is a single qubit operation that (up to normalization) maps $|0\rangle$ to $|0\rangle + |1\rangle$ and $|1\rangle$ to $|0\rangle - |1\rangle$. The corresponding matrix is

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

2.1.2 Exercises

Exercise 1 *What state do you get if you start with the basic state $|0^m\rangle$ and then apply the Hadamard gate to very qubit?*

Exercise 2 *Why can't OR and AND simply be implemented as quantum operations?*

Exercise 3 *Show that OR and AND can be implemented as quantum operations if we store the result in a 3rd qubit (that is initialized to $|0\rangle$).*

2.2 The class BQP

To define the number of elementary steps needed in a quantum computer requires some care because of composability: any two operations can be combined into one. However, more complex operations seems more unlikely to implement than elementary operations defined as follows:

Definition 2 (Elementary Quantum Operations) *A quantum operation is called elementary, or sometimes a quantum gate, if it acts on three or less qubits of the register.*

(The constant three is arbitrary: any constant above two would lead to an equally powerful model.)

Definition 3 (Quantum computation and BQP) *Let $f : \{0,1\}^* \to \{0,1\}$ and $T : \mathbb{N} \to \mathbb{N}$ be some functions. We say that f is computable in quantum $T(n)$-time if there is a polytime classical TM that on input $(1^n, 1^{(T(n)})$ for any $n \in \mathbb{N}$ outputs the description of quantum gates $F_1, \dots, F_T$ such that for every $x \in \{0,1\}^n$, we can compute $f(x)$ by the following process with probability at least $2/3$:*

1. *Initialize an m-qubit quantum register to the state $|x0^{m-n}\rangle$ where $m \leq T(n)$.*
2. *Apply one after the other $T \leq T(n)$ elementary quantum operations $F_1, \dots, F_T$ to the register.*
3. *Measure the register and let Y denote the obtained value.*
4. *Output Y_1.*

A Boolean function $f : \{0,1\}^* \to \{0,1\}$ is in **BQP** *if there is some polynomial $p : \mathbb{N} \to \mathbb{N}$ such that f is computable in quantum $p(n)$-time.*

Remark We can easily generalize the above definition to functions with more than one bit of output.

Remark It is tempting to think that the quantum model gives exponential speedup as the states of registers are described by 2^m-dimensional vectors and operations. However, this is not the case, one can describe ordinary probabilistic computation in a similar way. The added power of quantum computing seems instead to come from that here we allow vectors to have negative coefficients.

2.3 Classical computation as a subcase of quantum computers

Lemma 4 *If $f : \{0,1\}^n \to \{0,1\}^m$ is computable by a Boolean circuit of size S then there is a sequence of $O(S + m + n)$ elementary quantum operations computing the mapping $|x\rangle 0 \dots 0\rangle \mapsto |xf(x)0 \dots 0\rangle$*

Proof Easy proof in textbook. We sketch it here:

- Any gate AND, OR, or NOT can be implemented as an elementary operation. Thus we can compute f with roughly one operation per gate.

- We then use that quantum computing is reversible to clean up the tape so that we end with nice 0 qubits.

■

Corollary 5 **P** $\subseteq$ **BQP** *and more generally* **BPP** $\subseteq$ **BQP**.

Exercise 4 *Why does the above corollary follow from Lemma 4?*

3 Grover's Search Algorithm

- Consider the **NP**-complete problem SAT of finding, given an n-variable Boolean formula φ, whether there exists an assignment $a \in \{0,1\}^n$ such that $\varphi(a) = 1$.
- Using classical (even probabilistic) algorithms we do not know how to solve this problem faster than the trivial time of $poly(n)2^n$.
- Grover's result says that quantum computers can do it faster.

Theorem 6 *There is a quantum algorithm that given as input every polynomial-time computable function $f : \{0,1\}^n \to \{0,1\}$ (i.e., represented as a circuit computing f) finds in $poly(n)2^{n/2}$ time a string a such that $f(a) = 1$ (if such a string exists).*

For intuition, it is best described geometrically. We assume that f has a single satisfying assignment a (this is w.l.o.g. as can be further read about in the textbook).

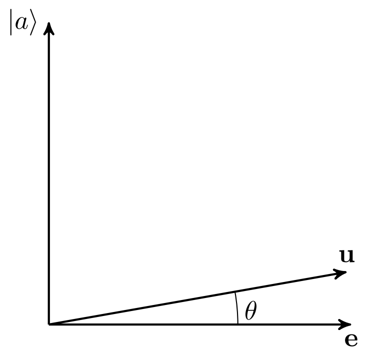
- Consider an n-qubit register [1], and let $\mathbf{u}$ denote the uniform state vector of this register. That is

$$\mathbf{u} = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle.$$

- Also consider the two orthogonal state vectors (that we cannot compute since we don't know a)

$$|a\rangle \qquad \text{and} \qquad \mathbf{e} = \frac{1}{\sqrt{2^n - 1}} \sum_{x \neq e} |x\rangle.$$

- Note that $\mathbf{u}$ lies in the span of $|a\rangle$ and $\mathbf{e}$. Indeed, $\mathbf{u} = \frac{1}{2^{n/2}}|a\rangle + \frac{\sqrt{2^n-1}}{2^{n/2}}\mathbf{e}$. The vectors can be illustrated as follows:



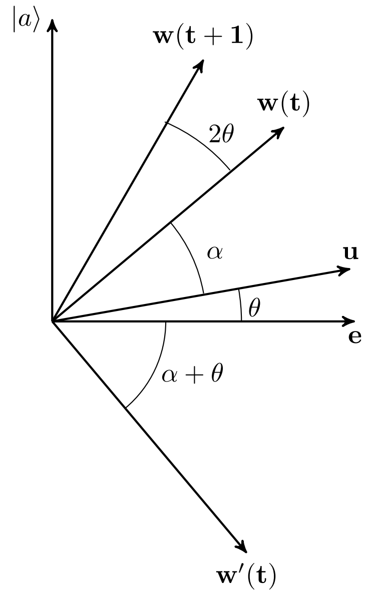
[1]We don't explicitly write out the extra (polynomially many) qubits we need to do some operations.

- What can we say about the angle θ? Well, we have $\cos(\theta) = \langle \mathbf{u}, \mathbf{e} \rangle$ and also $\sin(\theta) = \cos(\pi/2 - \theta) = \langle |a\rangle, \mathbf{u} \rangle = 1/2^{n/2}$. Now using that, $\phi \geq \sin(\phi)$ for $\phi \in [0, \pi/2]$ we get that

$$\theta \geq \frac{1}{2^{n/2}} \qquad \text{(actually very close to equality)}$$

- Now let $w(1) = u$ and repeat the following for $t = 1, \ldots \frac{1}{\theta}\frac{\pi}{4}$ ($O(2^{n/2}$ many iterations):
    1. Let $w'(t)$ be the reflection of $w(t)$ around $\mathbf{e}$.
    2. Let $w(t+1)$ be the reflection of $w'(t)$ around $\mathbf{u}$.

Of course, it is not clear how to implement the reflections above. But let's first see that the algorithm makes sense. It is easiest seen by looking at the illustration of the execution of one iteration:



In other words, every iteration decreases the degree by 2θ (until we passed $|a\rangle$ when the degree would start to increase again)

So after $\frac{1}{\theta}\frac{\pi}{4}$ iterations the degree between the final w and $|a\rangle$ would be

$$\frac{\pi}{2} - \theta - 2\theta\frac{1}{\theta}\frac{\pi}{4} \leq 0.00001$$

so when we measure w we would get $|a\rangle$ with good probability.

It remains to see how implement the two steps (the reflection around $\mathbf{e}$ and the reflection around $\mathbf{u}$).

3.1 Reflecting around e

Recall that in order to reflect a vector $\mathbf{w}$ around $\mathbf{e}$, we express $\mathbf{w}$ as $\alpha\mathbf{e} + \mathbf{e}^{\perp}$ and we wish to output the vector $\alpha\mathbf{e} + \mathbf{e}^{\perp}$. Thus the reflection around $\mathbf{e}$ is equal to $\sum_{x \neq a} \mathbf{w}_x |x\rangle - \mathbf{w}_a |a\rangle$.

Although, we of course don't know $|a\rangle$ this operation is easy to do:

1. Since f is computable in polytime, we can compute the transformation $|x\sigma\rangle|x(\sigma \oplus f(x))\rangle$ in polynomial time. This transformation maps $|x0\rangle$ to $x0\rangle$ for $x \neq a$ and $|a0\rangle$ to $|a1\rangle$.

2. Then we apply the elementary transformation $|0\rangle \mapsto |0\rangle$ and $|1\rangle \mapsto -|1\rangle$ on the qubit σ. This maps $|x0\rangle$ to $|x0\rangle$ for $x \neq a$ and $|a1\rangle$ to $-|a1\rangle$.

3. Finally, we apply the transformation $|x\sigma\rangle \mapsto |x(\sigma \oplus f(x))\rangle$ again, mapping $|x0\rangle$ to $|x0\rangle$ for $x = a$ and maps $|a1\rangle$ to $|a0\rangle$.

The final result is that the vector $|x0\rangle$ is mapped to itself for $x \neq a$ but $|a0\rangle$ is mapped to $-|a0\rangle$. Ignoring the last qubit, this is exactly reflection around $|a\rangle$.

3.2 Reflecting around u

This is a little bit less surprising as here we know the vector that want to reflect around. We can do as follows:

1. Apply the Hadamard operation to each qubit, mapping $\mathbf{u}$ to $|0\rangle$.

2. Then reflect around $|0\rangle$ (this can be done as when reflecting around $\mathbf{e}$ by letting f be the function that is 1 iff the input is all zeros).

3. Apply the Hadamard operation to each qubit, mapping back $\mathbf{u}$ to the uniform state.

Exercise 5 *How can you make a classic TM reversible?*