

Quantum Computing

Instructor: Alexander Razborov, University of Chicago.

razborov@cs.uchicago.edu

Course Homepage:

<http://people.cs.uchicago.edu/~razborov/teaching/winter21.html>

(Mostly) Winter Quarter 2011, Spring Quarter 2013 and Winter
Quarter 2021

Contents

| | | |
|----------|---|-----------|
| 1 | Classical and Quantum computation: circuit model | 4 |
| 1.1 | Reversible Computation | 4 |
| 1.2 | Probabilistic Computation | 8 |
| 1.3 | Crash Course in Linear Algebra | 10 |
| 2 | Early Quantum Algorithms | 14 |
| 2.1 | Deutsch algorithm (1985) | 14 |
| 2.1.1 | Black-box model | 14 |
| 2.1.2 | Ingredients of Deutsch Algorithm | 14 |
| 2.1.3 | The First Try | 15 |
| 2.1.4 | Successful Try: Interference | 16 |
| 2.2 | Deutsch-Josza algorithm (1992) | 17 |
| 2.3 | Simon's algorithm (1994) | 19 |
| 3 | $BQP \subseteq PP$ | 21 |
| 4 | Famous Quantum Algorithms | 22 |
| 4.1 | Grover's search algorithm (1996) | 22 |
| 4.1.1 | A Geometrical Interpretation | 23 |
| 4.1.2 | Some Details | 24 |
| 4.2 | Factoring: Shor's Algorithm | 26 |
| 4.2.1 | Reductions | 26 |
| 4.2.2 | Linear Algebra | 28 |
| 4.2.3 | Part 1: Phase Estimation Algorithm | 29 |

| | | |
|----------|--|-----------|
| 4.2.4 | Part 2: How to Construct $ u_k\rangle$? | 31 |
| 4.3 | Discrete Logarithm | 32 |
| 4.4 | Hidden Subgroup Problem | 32 |
| 4.4.1 | First Application - Symmetric Group | 33 |
| 4.4.2 | Second Application - Dihedral Group | 33 |
| 5 | Quantum Probability | 34 |
| 5.1 | “Tracing out” or “partial measurement” | 36 |
| 5.2 | Superoperators | 37 |
| 6 | Quantum Complexity Theory: black-box model | 38 |
| 6.1 | Hybrid method: optimality of Grover’s search | 39 |
| 6.2 | Quantum Query Complexity vs. Other Complexity Measures | 41 |
| 6.3 | Ambainis’s Adversary Method | 46 |
| 6.4 | Quantum Query Complexity and Formula Size | 49 |
| 7 | Quantum Communication Complexity | 49 |
| 7.1 | Probabilistic Communication Complexity | 50 |
| 7.2 | Quantum Communication Complexity | 51 |
| 7.3 | Decomposition of quantum protocols | 54 |
| 7.4 | Lower bound for $QC_2(\text{IP}_2)$ | 55 |
| 7.5 | Lower bound for $QC_2(\text{DISJ})$ | 57 |
| 7.6 | Generalizations of the discrepancy method | 59 |
| 7.7 | Direct products | 59 |
| 8 | Quantum Error-Correcting Codes | 60 |
| 8.1 | Operator-sum representation of superoperators | 60 |
| 8.2 | Projective Measurements | 61 |
| 8.3 | Quantum Information Theory | 61 |
| 8.3.1 | Error Correcting Codes in Classical Information Theory | 61 |
| 8.3.2 | Correcting Against Quantum Bit Flip | 62 |
| 8.3.3 | Correcting Against Quantum Phase Flip | 62 |
| 8.3.4 | Correcting Against Simultaneous Bit and Phase Flip | 63 |
| 8.4 | Conditions for the Recovery Operator | 63 |
| 8.5 | Stabilizer Codes | 68 |
| 8.5.1 | Definition and some Examples | 68 |
| 8.5.2 | Conditions on Pauli subgroups | 70 |
| 8.5.3 | Error Correcting properties of V_S | 71 |

| | | |
|----------|---|-----------|
| 9 | Extra Material: Quantum Interactive Proofs | 74 |
| 9.1 | Classical Merlin-Arthur Proofs | 74 |
| 9.2 | Quantum Merlin-Arthur Proofs | 75 |
| 9.2.1 | Two Examples | 75 |

Lecture 1

Scribe: Yuan Li, University of Chicago.

Date: April 2, 2013

In this course, we will cover the following 3 topics.

1. Circuit (Turing) computations.
2. Black-box models.
3. Communication complexity.

And we will also talk about Quantum information, density matrices, and error-correcting codes.

1 Classical and Quantum computation: circuit model

1.1 Reversible Computation

In the classical computation world, Turing machine is probably the most popular computation model. Bernstein and Vazirani (1987) defined Quantum Turing machine. However, it's not a popular model in the quantum world, and we will deal with quantum circuit most of the time.

Let's recall the definition of *circuit* in the classical computation world. Let

$$f : \{0, 1\}^* \rightarrow \{0, 1\}^*$$

be a function to be computed, where $\{0, 1\}^*$ denotes the set of all finite binary strings. For circuits, the input length is fixed, and thus we consider the slice of f , that is, let

$$f_n : \{0, 1\}^n \rightarrow \{0, 1\}^{m(n)}$$

(we assume for simplicity that the output size m depends only on the size of the input n). We say that a sequence of circuits $\{C_n\}$ computes f if $C_n(x_1, \dots, x_n) = f_n(x_1, \dots, x_n)$ for every n .

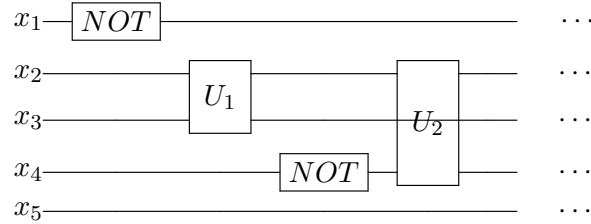
Let P be the class of functions computable in polynomial time (in $|x|$) on a Turing machine. Equivalently, function f is in P , if $\{f_n\}$ is computable by *uniform* polynomial size circuits ("uniform" refers to the fact that the circuits itself should be easily constructible; we do not dwell into further details here).

In the classical model, we usually draw a circuit in the top-down or bottom-up fashion. To induce the definition of quantum circuit, let's draw

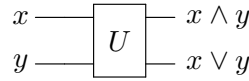
it in a different way – from left to right. Then the size of a quantum circuit is defined to be the number of gates, which coincides with the definition in classical circuit. Here comes our first assumption, which is different from classical circuits.

Postulate #1. All gates used in our circuits have the same number of input and output wires.

For example, our circuit may look like



Notice that the wire x_3 does not go through gate U_2 , that is, U_2 is not applied to x_3 . Here, gates U_1 and U_2 might look like this.



However, the above gate viewed as a function from $\{0,1\}^2 \rightarrow \{0,1\}^2$ is not a permutation. (To see this, it's impossible to have $x \wedge y = 1$ and $x \vee y = 0$.)

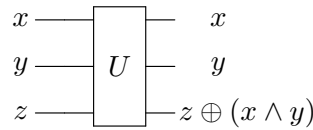
Now comes our second postulate.

Postulate #2. All gates are reversible (permutations), and thus the circuit is reversible.

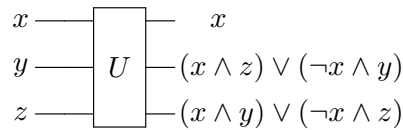
For example, the following is a NOT gate.



The following gate is called TOFFOLI gate, and it's easy to check the reversibility (in fact, it is an involution).



The following gate flips y and z if x is 1, otherwise nothing is changed.



Let $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ be a function. If f is not a permutation, it's clear that f can not be computed by a reversible circuit. If it is, then the problem is how to compute f given a prescribed set of gates? It seems that the quantum computing cares little about it in this exact form because the following concepts make a variant of this question way more natural.

Let $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ be an arbitrary function. Define the controlled- f

$$f_{\oplus} : \{0, 1\}^{n+m} \rightarrow \{0, 1\}^{n+m}$$

by

$$\begin{aligned} & f_{\oplus}(x_1, \dots, x_n, y_1, \dots, y_m) \\ &= (x_1, \dots, x_n, y_1 \oplus f_1(x), \dots, y_m \oplus f_m(x)). \end{aligned}$$

In particular, $f_{\oplus}(x_1, \dots, x_n, 0, \dots, 0) = (x_1, \dots, x_n, f_1(x), \dots, f_m(x))$. f_{\oplus} is reversible (e.g because it is an involution).

Under this notation,

$$\text{NOT} = 1_{\oplus},$$

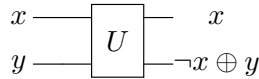
where 1 is the function in 0 arguments that outputs 1, and

$$\text{TOFFOLI} = (x \wedge y)_{\oplus}.$$

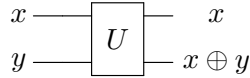
Define

$$\text{CNOT} = (\text{NOT})_{\oplus},$$

which looks like the following.



Or sometimes (there's a bit of confusion in the literature), it refers to the following:



In quantum computing, we often employ extra bits to help the computation, which are called *ancilla bits*. For example, if we are using L ancilla bits, we should compute a function that does the following:

$$(x_1, \dots, x_n, 0^{m+L}) \rightarrow (x_1, \dots, x_n, f_1(x), \dots, f_m(x), 0^L)$$

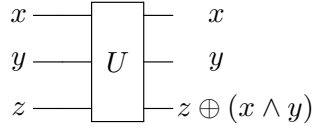
(and may behave arbitrarily on other inputs).

We will assume that ancilla bits can be employed in reversible (and later quantum) computation.

Now, we have finished our definition of a reversible circuit. A natural question is, can it simulate classical circuit? The following theorem gives an affirmative answer.

Theorem 1. *If $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ is computable by a circuit of size L , then we can compute f_{\oplus} by a reversible circuit with $L + m$ ancilla bits, and size $2L + m$.*

Proof. By construction. Given a circuit C computing f , let's construct a reversible circuit. For each gate AND, OR, and NOT in C , let's create a (fresh!) ancilla bit for the output wire. For example, if there is a gate in C which is the AND of wire x and y , then we introduce an ancilla bit z and a TOFFOLI gate as follows.



Given the input

$$(x_1, \dots, x_n, 0, \dots, 0),$$

after this procedure, we obtain a quantum circuit Q outputting

$$(x_1, \dots, x_n, f_1(x), \dots, f_m(x), \text{Garbage}, 0, \dots, 0),$$

where “Garbage” denotes by-products produced during the computation, which correspond to the intermediate wires in the original circuit, stored in ancilla registers.

Next, we copy the outputs $f_1(x), \dots, f_m(x)$ to the last m bits, which can be done by applying m CNOT gates, and thus we get

$$(x_1, \dots, x_n, f_1(x), \dots, f_m(x), \text{Garbage}, f_1(x), \dots, f_m(x)).$$

Finally, we need to clear the “garbage”. Here comes a very cute trick which will be often used in quantum computing. Let's apply Q^{-1} , which is obtained by reversing Q (both the gates and the order of the computation are reversed). Here, we take the advantage of reversibility, and it's easily checked that both QQ^{-1} and $Q^{-1}Q$ are the identity map. Now, we get the desired output:

$$(x_1, \dots, x_n, 0, \dots, 0, f_1(x), \dots, f_m(x)).$$

The number of ancilla bits used is $L + m$, and the number of quantum gates is $2L + m$. \square

Lecture 2

Scribe: Ian Alevy, University of Chicago.

Date: April 4, 2013

1.2 Probabilistic Computation

As a model of probabilistic computation we suppose that our Turing machine M is allowed to flip a coin at any time during the computation. A language $L \subset \{0,1\}^*$ will be identified with its *characteristic function* $f : \{0,1\}^* \rightarrow \{0,1\}$ such that $f(x) = 1$ if and only if $x \in L$. Now we can define probabilistic complexity classes.

Definition 2. The complexity class BPP (bounded-error probabilistic polynomial time) is the set of all languages L for which there exists a probabilistic Turing machine M such that

$$\begin{aligned}x \in L &\implies P[M \text{ accepts } x] \geq 2/3 \\x \notin L &\implies P[M \text{ accepts } x] \leq 1/3\end{aligned}$$

and M runs in polynomial time in $|x|$.

The "B" in the acronym refers to the assumption that the error probability, $p = P[M \text{ accepts } x]$ for $x \notin L$ (or vice versa), is bounded away from $1/2$. To be consistent we will always set $p = 1/3$. However, the choice of p is irrelevant as long as there is some fixed (or even an inversely polynomial in n) $\epsilon > 0$ such that $p = 1/2 - \epsilon$. One can use the Chernoff Bound to reduce the error probability dramatically by repeating the algorithm a small number of times. We can form a larger complexity class if we remove the restriction that p is bounded strictly below $1/2$.

Definition 3. The complexity class PP (probabilistic polynomial time) is the set of all languages recognizable with error probability $p < 1/2$, i.e., when we can achieve any advantage (even if arbitrarily small!) over a random coin toss.

We can simulate M with a deterministic Turing machine, M_d , if the machine has access to a binary string in which the results of coin tosses are stored. A new variable r is introduced into the characteristic function, $f(x,r)$ to represent the probabilistic dependence on the binary string. If the inputs are of the form $x \in \{0,1\}^m$ then we can represent probability distributions on inputs as vectors (p_x) of length 2^m , where p_x denotes the

probability of the binary string x . The two standard assumptions for probability distributions are $p_x \geq 0$ and $\sum_x p_x = 1$. Both the inputs and outputs to the circuits are random vectors. Circuits naturally act on probability distributions, and this is an example of a push-forward measure. We, however, will need a matrix representation of this action.

Any reversible computation is equivalent to the action of a permutation matrix on a probability vector. Permutation matrices have entries in the set $\{0, 1\}$ and exactly one nonzero entry in each row and each column. As an example consider the case $m = 1$ with $p_0 = p$ and $p_1 = q$. The NOT gate can be represented by its action as

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix} \rightarrow \begin{pmatrix} q \\ p \end{pmatrix}.$$

Likewise the CNOT gate is represented by

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} p_{00} \\ p_{01} \\ p_{10} \\ p_{11} \end{pmatrix} \rightarrow \begin{pmatrix} p_{00} \\ p_{01} \\ p_{11} \\ p_{10} \end{pmatrix}.$$

There is an analogy between this and the interpretation of (non-unitary!) quantum computation as an application of superoperators to arbitrary *density matrices* (probability distributions being just diagonal density matrices), but we will discuss this much later in the course.

Definition 4. A nonnegative matrix such that every column and every row sums to one is called a *doubly stochastic matrix*.

As an example, let us simulate in this formalism the natural instruction “with probability $1/2$ apply the CNOT gate, and with probability $1/2$ do not do anything”. With probability of $p = 1/2$ the gate is

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

and with probability $q = 1 - p = 1/2$ the gate is

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

The probabilistic CNOT is the normalized linear combination of these matrices,

$$\frac{1}{2} \left(\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \right) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \end{pmatrix}.$$

Notice that this is a convex combination of permutation matrices. In fact this is a characterization of doubly stochastic matrices.

Theorem 5 (Birkhoff-von Neumann). *A matrix is doubly stochastic if and only if it is a convex combination of permutation matrices.*

1.3 Crash Course in Linear Algebra

We identify $\mathbb{R}^{2^n} = \bigotimes_{i=1}^n \mathbb{R}^2$, where \bigotimes denotes the tensor product. In this course we use the computer science notion of the tensor product. If L is a vector space with basis $\{e_1, \dots, e_n\}$ and M is a vector space with basis $\{f_1, \dots, f_\ell\}$ then $L \otimes M$ is a vector space with basis $\{e_i \otimes f_j | i = 1, \dots, n, j = 1, \dots, \ell\}$. The existence of a natural basis makes the computer science definition simpler to work with than abstract mathematical notions. The standard basis for \mathbb{R}^2 is denoted $\{e_1, e_2\}$ and therefore the standard basis for \mathbb{R}^{2^n} is $\{\bigotimes_{i=1}^n e_{x_i} | x_i \in \{1, 2\}\}$.

The CNOT gate defined above is not sufficient because as written it only acts on 2 bits and we need the gates to act on all n bits. Using the tensor product notation we can decompose $\mathbb{R}^{2^n} = \mathbb{R}^{2^I} \otimes \mathbb{R}^{2^{\text{co}-I}}$, where $I \subseteq \{1, 2, \dots, n\}$ and $\text{co}-I = \{1, 2, \dots, n\} \setminus I$. Now we can write the CNOT gate acting on all of \mathbb{R}^{2^n} by defining the action on \mathbb{R}^{2^I} in the usual manner and the action on $\mathbb{R}^{2^{\text{co}-I}}$ as the identity.

To study quantum computation, we need to work in the *complex* vector space $\mathbb{C}^{2^n} = \bigotimes_{i=1}^n \mathbb{C}^2$ equipped with the inner product

$$\langle \alpha, \beta \rangle = \sum_x \alpha_x^* \beta_x$$

for vectors $\alpha = (\alpha_x | x \in \{0, 1\}^n)$ and $\beta = (\beta_y | y \in \{0, 1\}^n)$ in \mathbb{C}^{2^n} . Here α_x^* denotes the complex conjugate of α_x . For now we will work over finite dimensional vector spaces. In this case the inner product space is in fact simply a finite dimensional Hilbert space, \mathcal{H} .

For a linear operator T we write T^\dagger to denote the *conjugate transpose* of T obtained by taking the complex conjugate of every entry of the transpose

of T . This operation is also referred to as the *Hermitian conjugate* or *adjoint operator*. Notice that for permutation matrices we have $P^T P = I$, and, more generally, for real orthogonal matrices $U^T U = I$. Generalizing even further, a matrix with complex entries is called *unitary* if $U^\dagger U = I$. It is an exercise to verify that the set of all unitary matrices is a group under the operation of matrix multiplication. Every quantum computation can be represented by a unitary matrix, that's why unitary operators are of the utmost importance in this course.

Exercise 6. If U, V are unitary operators on the vector spaces L and M respectively, then $U \otimes V$ acts on $L \otimes M$. Show that the tensor product of unitary operators is unitary.

The *Pauli group* consists of the four unitary matrices

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

along with their multiples by scalars from $\{\pm 1, \pm i\}$. $I, \sigma_x, \sigma_y, \sigma_z$ form a linear basis for the set of all 2×2 matrices. This group is very important in physics, and we will often see its elements featured in our course.

The norm in \mathcal{H} is

$$|\alpha| = \langle \alpha, \alpha \rangle^{1/2} = \left(\sum_x |\alpha_x|^2 \right)^{1/2}.$$

A vector α is called *unit* if $|\alpha| = 1$. Notice that unitary matrices preserve unit vectors because for a unitary matrix U we have

$$\langle U\alpha, U\alpha \rangle = (U\alpha)^\dagger U\alpha = \alpha^\dagger U^\dagger U\alpha = \alpha^\dagger \alpha = |\alpha|^2.$$

This is analogous to the case of doubly stochastic matrices in which the property of being a probability distribution is preserved. A *pure state* is an arbitrary unit vector α , i.e. the one satisfying $\sum_x |\alpha_x|^2 = 1$. For the time being, pure states will be identified with the states of a quantum mechanical system. If α represents such a state, then $p_x = |\alpha_x|^2$ represents the probability that the system is found in the classical state x after a complete measurement. The quantity $|\alpha_x|$ is referred to as the *amplitude* of x . Unitary matrices preserve amplitudes, hence they also preserve pure states.

Exercise 7. A matrix U is unitary if and only if it preserves unit vectors.

Dirac's Notation

Dirac's notation presumably simplifies the notation for tensor products. We write

$$|x\rangle := \bigotimes_{i=1}^n e_{x_i}$$

and refer to this as a "ket-vector" or "ket". We write the inner product of two kets, $|\phi\rangle, |\psi\rangle$ as $\langle\phi|\psi\rangle$ and refer to $\langle\phi|$ as a "bra-vector" or "bra". This notation can be interpreted as $\langle\phi| = |\phi\rangle^\dagger$, where $|\phi\rangle$ is a column vector. To see how this notation helps us we note that

$$|0\rangle \otimes |1\rangle = |01\rangle,$$

where $e_0 = |0\rangle$, $e_1 = |1\rangle$, and $e_{01} = |01\rangle$. The tensor product operation is just concatenation, although we will sometimes resort to the unabridged \otimes notation when warranted. The application of a linear operator U to a vector is written as $U|\phi\rangle$. We also write

$$\langle\psi, U\phi\rangle = \langle\psi|U|\phi\rangle.$$

Notice that unitary operators (as, for that matter, any square matrix) have the property

$$\langle\psi|T^\dagger|\phi\rangle = \langle\phi|T|\psi\rangle^*$$

(remember that replacing bra with ket automatically conjugates entries). Etc. The best way to get used to Dirac's notation is to contemplate over one or two concrete calculations performed with its help.

The Hadamard gate, H , is one of the most important gates in quantum computing. It is a 2×2 matrix defined in terms of its action

$$\begin{aligned} H|0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ H|1\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \end{aligned}$$

This is represented by the unitary matrix

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

Lecture 3

Scribe: Young Kun Ko, University of Chicago.

Date: April 7, 2013

Geometrically, this can be seen as a reflexion: $H^2 = I$.

Theorem 8. *Every unitary operator can be realized as a product of unitary operators acting on 1 or 2 qubits.*

We will not prove this theorem as its approximate version (see Definition 11) turns out to be way more important.

Recall the definition of controlled operator in the classical world.

Definition 9. Let $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$. Then the controlled version of f , denoted $f_{\oplus} : \{0, 1\}^{n+m} \rightarrow \{0, 1\}^{n+m}$ acts as follows

$$(x_1, \dots, x_n, y_1, \dots, y_m) \mapsto (x_1, \dots, x_n, y_1 \oplus f_1(x), \dots, y_m \oplus f_m(x))$$

We can extend this definition to the quantum world.

Definition 10. Let U be a unitary operator. Then the controlled version of U , denoted $\Lambda^n(U)$ acts as follows

$$\Lambda^n(U) |x\rangle \otimes |y\rangle = \begin{cases} |x\rangle \otimes |y\rangle & \text{if } x_1 \dots x_n = 0 \\ |x\rangle \otimes U |y\rangle & \text{otherwise.} \end{cases}$$

Under the above definition, well-known gates can be seen as controlled operators.

$$\begin{aligned} \Lambda^1(\sigma_x) &= \text{CNOT} \\ \Lambda^2(\sigma_x) &= \text{TOFFOLI.} \end{aligned} \quad (\text{up to negation})$$

Definition 11. A set of gates Q is *universal* if every unitary operator on a fixed number of qubits can be approximated within ϵ using $\text{poly}(\log_2(1/\epsilon))$ number of gates in Q .

Definition 12. A language L is in BQP if there exists a polynomially sized circuit Q such that

- $x \in L \Rightarrow \mathbb{P}[Q \text{ accepts } x] \geq 2/3$
- $x \notin L \Rightarrow \mathbb{P}[Q \text{ accepts } x] \leq 1/3,$

where $\mathbb{P}[Q \text{ accepts } x] = \sum_{y=1} |\alpha_y|^2$ for $U_Q |x, 0^L\rangle = \sum_y \alpha_y |y\rangle$.

Theorem 13. *Universal bases exist:*

1. $\left\{ H, K = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, K^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}, \text{CNOT}, \text{TOFFOLI} \right\}$. Up to normalization, all entries are $\{\pm 1, \pm i\}$.
2. $\left\{ H, \text{CNOT}, T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \right\}$. Note that $T^2 = K$.

2 Early Quantum Algorithms

Sketched by Pratik Worah in 2011.
Scribe: Jue Xu, University of Chicago.
Date: January 16, 2018

2.1 Deutsch algorithm (1985)

2.1.1 Black-box model

Deutsch algorithm solves a kind of the most elementary problem "black-box model", which is also called *Oracle model* or *query model*. The **black-box function** in Deutsch algorithm is a one-bit Boolean function:

$$f : \{0, 1\} \rightarrow \{0, 1\}$$

the input is a bit 0 or 1 and the output is also a bit 0 or 1. We don't know the inside circuitry of this black-box function, but we can query it to obtain its output. It is easy to observe that there are four Boolean functions in one variable:

$$\begin{aligned} f_1(0) = 0, f_1(1) = 1; & \quad f_2(0) = 1, f_2(1) = 0; \\ f_3(0) = 0, f_3(1) = 0; & \quad f_4(0) = 1, f_4(1) = 1. \end{aligned}$$

According to the value of $f(0) \oplus f(1)$, these four functions can be classified as below:

- *Balanced functions:* $f_1(x) = x, f_2(x) = \bar{x} \iff f(0) \oplus f(1) = 1$, i.e., the output is related with the input
- *Constant functions:* $f_3(x) = 0, f_4(x) = 1 \iff f(0) \oplus f(1) = 0$, i.e., the output has no relation with the input

2.1.2 Ingredients of Deutsch Algorithm

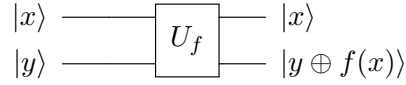
Now, the problem can be formulated as follows:

Input: A Boolean function $f : \{0, 1\} \rightarrow \{0, 1\}$ given as a black-box.

Problem: We want to determine the value of $f(0) \oplus f(1)$, equivalently, judge whether the function is constant or balanced.

The classical algorithm requires querying this black-box function twice to solve this problem, while, in the quantum world, Deutsch algorithm only needs one query! All ingredients we need to implement Deutsch algorithm are

- The initial state $|01\rangle$ with two qubits: the first qubit can be regarded as the target qubit and the second one is the control qubit, which are required by the reversibility of controlled function (discussed in Lecture 1).
- Hadamard gates H (introduced in Lecture 2) are used to apply *Discrete Fourier Transform* over \mathbb{Z}_2 (we will talk about this connection in more detail later), or in physical words, to generate interference.
- A black-box operator $U_f: |x, y\rangle \xrightarrow{U_f} |x, y \oplus f(x)\rangle$

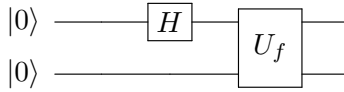


- A measurement operator: projector $P = |0\rangle\langle 0| + |1\rangle\langle 1|$ (again, the general theory of measurements will be discussed later in the course).

2.1.3 The First Try

Before giving the right algorithm, we will show an unsuccessful try and discuss why it doesn't work.

Now the initial state is $|00\rangle$. We could start with the following transformation:



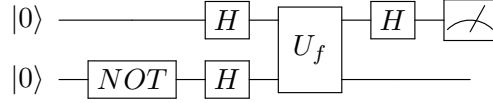
- Apply Hadamard gates to the first qubit of the initial state $|00\rangle$.
- Apply oracle operator U_f once.

$$\begin{aligned}
 |00\rangle &\xrightarrow{H \otimes I} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \\
 &\xrightarrow{U_f} \frac{1}{\sqrt{2}}(|0, f(0)\rangle + |1, f(1)\rangle).
 \end{aligned}$$

Now, information-theoretically we are in a good shape: states corresponding to our four functions are pairwise different. But where do we go from here? How to *extract* this information quantumly? It turns out that this is in fact impossible: angles between these states are sort of erratic while in order to be successful we want states leading to the same answer to coincide (well, up to a global phase change) and states leading to different answers to be orthogonal. With this guiding principle in mind, we can now fix our attempt: the key to success is to produce interference.

2.1.4 Successful Try: Interference

We modify our circuit by applying a NOT gate to the second qubit of the initial state $|00\rangle$.



That is, we created in the second register a new state $|\phi\rangle$ defined as follows:

$$|0\rangle \xrightarrow{\text{NOT}} \xrightarrow{H} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \stackrel{df}{=} |\phi\rangle.$$

This new state has the following property:

$$|x\rangle \otimes |\phi\rangle \xrightarrow{U_f} (-1)^{f(x)} |x\rangle \otimes |\phi\rangle, \quad x \in \{0, 1\}$$

which can be easily verified by direct calculation. This property will make a big difference in the result due to the magic of **interference**. Let's see it:

$$\begin{aligned} |01\rangle &\xrightarrow{H \otimes H} \frac{1}{\sqrt{2}}(|0\rangle|\phi\rangle + |1\rangle|\phi\rangle) \\ &\xrightarrow{U_f} \frac{1}{\sqrt{2}}[(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle] \otimes |\phi\rangle \\ &\xrightarrow{H \otimes I} \pm |f(0) \oplus f(1)\rangle \otimes |\phi\rangle. \end{aligned}$$

We can see that after applying U_f operator, the first qubit of the state depends on which category this black-box function belongs to

$$\frac{1}{\sqrt{2}}[(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle] = \begin{cases} \pm \frac{1}{\sqrt{2}}[|0\rangle + |1\rangle] \equiv \pm H|0\rangle, & f \text{ is balanced} \\ \pm \frac{1}{\sqrt{2}}[|0\rangle - |1\rangle] \equiv \pm H|1\rangle, & f \text{ is constant,} \end{cases}$$

and note that in accordance with our principle above, *these two states are orthogonal*. With the fact that $H^2 = I$, we can get an amazing final state by applying Hadamard gate to the first qubit

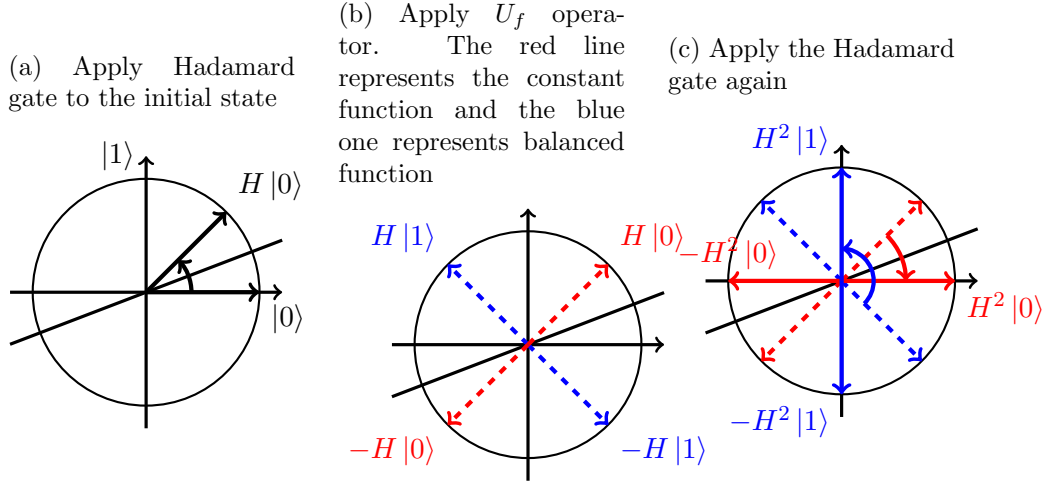
$$\xrightarrow{H \otimes I} \begin{cases} \pm H^2|0\rangle \otimes |\phi\rangle = \pm |0\rangle \otimes |\phi\rangle, & f \text{ is balanced} \\ \pm H^2|1\rangle \otimes |\phi\rangle = \pm |1\rangle \otimes |\phi\rangle, & f \text{ is constant,} \end{cases}$$

which can be succinctly written as

$$\pm |f(0) \oplus f(1)\rangle \otimes |\phi\rangle.$$

Now, we measure the first qubit in the basis $|0\rangle$ and $|1\rangle$. We will get $|0\rangle$ if and only if f is a constant function, otherwise, it is a balanced function.

Figure 1: Geometrical interpretation of the transformation on the first qubit



Lectures 4-5

Scribe: Tatiana Orlova, University of Chicago.
Date: January 18 and 20, 2011

2.2 Deutsch-Josza algorithm (1992)

Deutsch-Josza algorithm is a generalization of Deutsch algorithm we studied in the previous lecture.

Suppose we are given a Boolean function in 2^n inputs

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

We consider two types of Boolean functions. First, a constant function, as in the previous lecture. Second, a *balanced* function, which is a function that outputs as many zeros as ones (2^{n-1} zeros and 2^{n-1} ones) over its input set. For example, in the simplest case $n = 2$, all Boolean functions are well defined by these two types, and, moreover, we are in the situation of Section 2.1.

Given $f(x)$, we want to determine whether it is constant or balanced. We can try to solve this problem deterministically. Clearly, if we make 2^{n-1} queries we might get unlucky and get all 0s or all 1s. Thus, we need to know at least $2^{n-1} + 1$ values of $f(x)$ to decide whether it is constant or balanced, which simply means too many queries! We would like to see if using quantum computation can help us significantly reduce the number of queries.

Similar to the Deutsch algorithm we will first prepare the state $|\phi\rangle$:

$$|0\rangle \xrightarrow{\text{NOT}} \xrightarrow{H} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \stackrel{df}{=} |\phi\rangle.$$

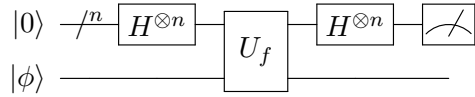
Recall, that the U_f -operator is defined by

$$U_f: |x, y\rangle \mapsto |x, y \oplus f(x)\rangle.$$

The initial state of the circuit is

$$|0^n\rangle |\phi\rangle.$$

We then perform the following transformations ($/^n$ stands for duplicating a state n times)



$$\begin{aligned} |0^n\rangle |\phi\rangle &\xrightarrow{H^{\otimes n}} \frac{1}{2^{n/2}} \sum_x |x\rangle |\phi\rangle \xrightarrow{U_f} \frac{1}{2^{n/2}} \sum_x |x\rangle \frac{|f(x)\rangle - |f(x) \oplus 1\rangle}{\sqrt{2}} \\ &= \frac{1}{2^{n/2}} \sum_x (-1)^{f(x)} |x\rangle |\phi\rangle, \end{aligned}$$

where the sum is over all all binary strings $x \in \{0, 1\}^n$. Note, that

$$(-1)^{f(x)} \frac{|f(x)\rangle - |f(x) \oplus 1\rangle}{\sqrt{2}} \equiv |\phi\rangle$$

regardless of $f(x)$.

The transformation

$$U_f^*: |x\rangle \mapsto (-1)^{f(x)} |x\rangle$$

changes the phase of any vector according to $f(x)$ and, unlike U_f does not use ancilla bits. So, we simply moved from one representation to another, and, in fact, in future we will often be using this alternate form of feeding f to our algorithms.

Discrete Fourier Transform

$$H^{\otimes n} |x\rangle = \frac{1}{2^{n/2}} \sum_y (-1)^{\langle x, y \rangle} |y\rangle,$$

where x, y are strings of length n , and $\langle x, y \rangle$ is ordinary inner product (over \mathbb{Z} or \mathbb{F}_2).

$$H^{\otimes n} \frac{1}{2^{n/2}} \sum_x (-1)^{f(x)} |x\rangle = \frac{1}{2^n} \sum_{x,y} (-1)^{f(x) + \langle x, y \rangle} |y\rangle.$$

We measure first n qubits and look at the coefficient α_0 corresponding to $|0^n\rangle$. The value $\alpha_0 \alpha_0^*$ can be interpreted as the probability of getting 0^n as the result of this measurement.

$$\alpha_0 = \frac{1}{2^n} \sum_x (-1)^{f(x)} = \begin{cases} \pm 1, & \text{if } f(x) \text{ is constant;} \\ 0, & \text{if } f(x) \text{ is balanced.} \end{cases}$$

Thus, the quantum Deutsch-Josza algorithm requires only a **single** query versus an **exponential** number of queries in classical deterministic case. It does not have this much advantage over the classical probabilistic computation, that provides a correct answer with probability $1 - \frac{1}{2^n}$ by using only $O(n)$ queries. This fact makes the result of this section a little bit less exciting :) In the next section we will discuss the first example of an exponential gap between classical and quantum computation.

2.3 Simon's algorithm (1994)

The problem can be formulated as follows.

Input: A function

$$f: \{0, 1\}^n \rightarrow \{0, 1\}^m,$$

where m can possibly be larger than n .

Promise: There exists $s \in \{0, 1\}^n$ such that $f(x) = f(y)$ if and only if $x = y$ or $x = y + s$. (See Figure 2 for an intuitive interpretation of s .)

Problem: We want to determine this value of s .

The Simon's problem is a part of the class of problems known as the *hidden subgroup problems*. In order to demonstrate the significance of the problems in this class we will give a few examples. Solving this problem for \mathbb{Z} , which is an infinite abelian group, will imply solving integer factorization problem. For the *Dihedral* group, the smallest non-abelian group, the result

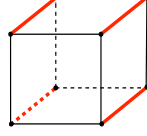


Figure 2: Function $f(x)$ has the same values on the edges that share the red direction.

is unknown. For the *symmetric* group the problem implies an efficient algorithm for the graph isomorphism problem. We will discuss all this in more details in Section 4.4 below.

The solution scheme will be extremely similar to what we already did. Now the initial state will be

$$|0^n, 0^m\rangle.$$

We then perform the following transformations:

$$\begin{aligned}
& \begin{array}{c} |0\rangle \xrightarrow{-/n} \boxed{H^{\otimes n}} \\ |0\rangle \xrightarrow{-/m} \end{array} \boxed{U_f} \boxed{H^{\otimes n}} \boxed{\text{Measurement}} \\
& |0^n, 0^m\rangle \xrightarrow{H^{\otimes n}} \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x, 0^m\rangle \xrightarrow{U_f} \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x, f(x)\rangle \xrightarrow{H^{\otimes n}} \\
& \xrightarrow{H^{\otimes n}} \frac{1}{2^n} \sum_{x,y \in \{0,1\}^n} (-1)^{\langle x,y \rangle} |y, f(x)\rangle \\
& = \frac{1}{2^n} \sum_{y \in \{0,1\}^n} \left(\sum_{x \in \{0,1\}^n} (-1)^{\langle x,y \rangle} |y, f(x)\rangle \right).
\end{aligned}$$

Note, that since $f(x)$ is not injective, terms $|y, f(x)\rangle$ in the final sum come in pairs and those with the opposite sign will cancel each other out. To show this, we consider coefficients of $|y, z\rangle$ for all $z \in \{0,1\}^m$, such that $f(x) = z$, and $f(x+s) = z$. We have

$$(-1)^{\langle x,y \rangle} + (-1)^{\langle x+s,y \rangle} = (-1)^{\langle x,y \rangle} (1 + (-1)^{\langle s,y \rangle}).$$

If $\langle s, y \rangle = 1$, i.e., s and y are not orthogonal then

$$(-1)^{\langle x, y \rangle} (1 + (-1)^{\langle s, y \rangle}) = 0.$$

At the end we have a uniform superposition $|y, z\rangle$, such that $y \perp s$. We then repeat the experiment from scratch multiple times to recover s .

The Simon's algorithm is a good example of “quantum magic” - *interference*, or cancelations. When we send our queries to $f(x)$, values that would appear with high probability might cancel out because of interference.

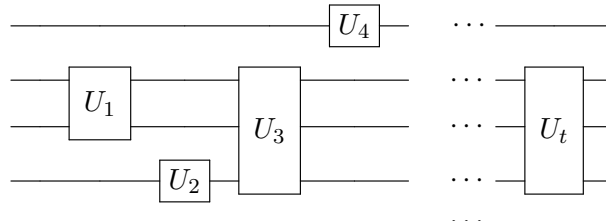
3 BQP \subseteq PP

We want to show that

$$BQP \subseteq PP.$$

One characterization of languages in PP is this: $L \in PP$ if and only if for all $x \in L$ we have $\#\{y: A(x, y)\} - \#\{y: R(x, y)\} > 0$, and for all $x \notin L$ we have $\#\{y: A(x, y)\} - \#\{y: R(x, y)\} < 0$, where $x, y \in \{0, 1\}^n$ are strings of polynomial range, and A, R are polynomially time computable. The way to think of this definition is that $\{y: A(x, y)\}$ and $\{y: R(x, y)\}$ are accepting and rejecting configurations respectively, and when y is such that neither of them happens (that typically will be the most frequent case), we toss a fair coin.

Consider a quantum circuit C with gates of the following types: $\{H, K, \text{CNOT}, \text{TOFFOLI}\}$. At every particular time it executes one quantum operator. Thus, the whole computation is the product of the quantum operators



$$U_t \dots U_2 U_1 |x, 0^L\rangle = \sum_y \alpha_y |y\rangle$$

We would like to compute coefficients α_y . To do this we sum up the corresponding positions in quantum operators

$$\alpha_y = \sum_{|z_1\rangle, \dots, |z_{t-1}\rangle} \langle y | U_t | z_{t-1} \rangle \langle z_{t-1} | U_{t-1} | z_{t-2} \rangle \dots \langle z_1 | U_1 | x_1, 0^L \rangle.$$

Note that if y and z_t are basis vectors then

$$\langle y|U_t|z_t\rangle = U_t[y, z_t].$$

The matrix entries of our operators, except for Hadamard gate, are from the set

$$\{0, \pm 1, \pm i\}.$$

So that the resulting values remain in this set, since it is closed under multiplication. Only the Hadamard gate creates nontrivial coefficients

$$\frac{1}{\sqrt{2}}\{\pm 1\}.$$

But the number of Hadamard gates in C is known in advance and does not depend on the input.

Denote this number by h . We have

$$\alpha_y = \sum_{|z_1\rangle, \dots, |z_{t-1}\rangle} \frac{1}{2^{h/2}} f(z_1, \dots, z_{t-1}, x, y),$$

where $f \in \{0, \pm 1, \pm i\}$ is efficiently computed. The amplitude is

$$\alpha_y \alpha_y^* = \frac{1}{2^h} \sum_{\substack{|z_1\rangle, \dots, |z_{t-1}\rangle \\ |z'_1\rangle, \dots, |z'_{t-1}\rangle}} f(z_1, \dots, z_{t-1}, x, y) f^*(z'_1, \dots, z'_{t-1}, x, y).$$

We have

$$\begin{aligned} \alpha_y \alpha_y^* = & \frac{1}{2^{h+1}} \left(\# \left\{ \vec{z}, \vec{z}' : f(z, x, y) f^*(z', x, y) = 1 \right\} \right. \\ & \left. - \# \left\{ \vec{z}, \vec{z}' : f(z, x, y) f^*(z', x, y) = -1 \right\} \right). \end{aligned}$$

This gives us the desired predicates A and R and hence finishes the proof.

4 Famous Quantum Algorithms

4.1 Grover's search algorithm (1996)

We begin with a simple case (when the solution is known to be unique) that, however, already contains all essential ideas. The general case is sketched below in Section 4.1.2.

Input: $f: [N] \rightarrow \{0, 1\}$, where $[N]$ stands for a domain of size N , which in particular could be binary strings of length $\log_2 N$.

Promise: There exists a unique w such that $f(w) = 1$.

Problem: We want to find this w .

Theorem 14. *There exists a quantum algorithm that performs search in time $O(\sqrt{N})$.*

We will see in Section 6.1 that this bound is tight.

4.1.1 A Geometrical Interpretation

Consider a standard superposition

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

Let $|w\rangle$ be the unknown unit vector we want to find. Define

$$|\psi_{\text{Bad}}\rangle \stackrel{\text{df}}{=} \frac{1}{\sqrt{N-1}} \sum_{x \neq w} |x\rangle.$$

The vectors $|\psi\rangle$ and $|\psi_{\text{Bad}}\rangle$ generate a plane (2-dimensional subspace). Clearly,

$$|\psi\rangle = \sqrt{\frac{N-1}{N}} |\psi_{\text{Bad}}\rangle + \frac{1}{\sqrt{N}} |w\rangle,$$

and the vector

$$|\bar{\psi}\rangle = \sqrt{\frac{N-1}{N}} |w\rangle - \frac{1}{\sqrt{N}} |\psi_{\text{Bad}}\rangle.$$

is orthogonal to $|\psi\rangle$. In other words, $\{|\psi\rangle, |\bar{\psi}\rangle\}$ is an orthogonal basis in the subspace generated by the vectors $|\psi\rangle$ and $|\psi_{\text{Bad}}\rangle$. Denote the angle between $|\psi\rangle$ and $|\psi_{\text{Bad}}\rangle$ by θ , then

$$|\psi_{\text{Bad}}\rangle = \cos \theta |\psi\rangle - \sin \theta |\bar{\psi}\rangle.$$

From the inner product $\langle \psi_{\text{Bad}} | \bar{\psi} \rangle$, we find

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

and thus

$$\theta \approx \frac{1}{\sqrt{N}}.$$

Consider two geometrical transformations:

1. Reflection on the line defined by $|\psi_{\text{Bad}}\rangle$ - this transformation corresponds to U_f^* ;
2. Reflection on the line defined by $|\psi\rangle$ - we denote this transformation by V .

Key Idea: Define a new transformation, **Grover Iterate**, $G \stackrel{\text{df}}{=} VU_f^*$. It is a composition of U_f^* and a reflection V , and results in a rotation by 2θ . If we apply this operator to a unit vector precisely $\lfloor \sqrt{N} \frac{\pi}{4} \rfloor$ times we will rotate this vector by $\approx \frac{\pi}{2}$. Thus, if we apply Grover Iterate $\lfloor \sqrt{N} \frac{\pi}{4} \rfloor$ times to $|\psi\rangle$ it will become “almost” $|w\rangle$.

4.1.2 Some Details

We now fill in the details. Here we assume that all objects from $[N] = \{1, 2, \dots, N\}$ are binary strings. We would like to construct a unitary operator V with the following properties

$$V|\psi\rangle = |\psi\rangle$$

and

$$V(|x\rangle - |y\rangle) = |y\rangle - |x\rangle.$$

We first apply DFT (or Hadamard matrices)

$$H^{\otimes n} |x\rangle = \frac{1}{2^{n/2}} \sum_y (-1)^{\langle x, y \rangle} |y\rangle$$

Let U_0 be a unitary operator such that

$$U_0 |0\rangle = |0\rangle$$

and

$$U_0 |x\rangle = -|x\rangle.$$

Thus, U_0 flips the phase of all non-zero vectors. We leave the explicit construction of U_0 as an exercise.

We now apply U_0 to the $H^{\otimes n} |x\rangle$ and obtain

$$U_0 H^{\otimes n} |x\rangle = \frac{1}{2^{n/2}} (2|0\rangle - \sum_y (-1)^{\langle y, x \rangle} |y\rangle)$$

Finally, we apply n Hadamard gates again

$$H^{\otimes n} U_0 H^{\otimes n} |x\rangle = \frac{1}{2^n} \left(2 |\psi\rangle - \sum_{y,z} (-1)^{\langle y,x \rangle} (-1)^{\langle y,z \rangle} |z\rangle \right).$$

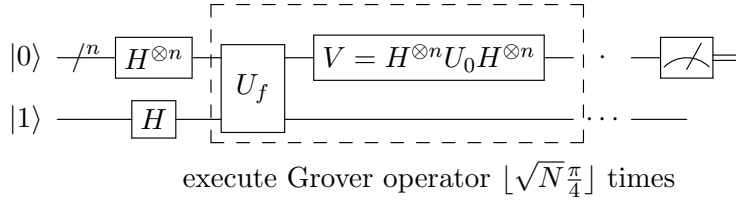
Terms where $|z\rangle \neq |x\rangle$ will cancel out. So, we have

$$H^{\otimes n} U_0 H^{\otimes n} |x\rangle = \frac{1}{2^n} (2 |\psi\rangle - 2^n |x\rangle) = -|x\rangle + \frac{2}{2^n} |\psi\rangle.$$

Let us check

$$H^{\otimes n} U_0 H^{\otimes n} |\psi\rangle = (-|\psi\rangle + 2 |\psi\rangle) = |\psi\rangle$$

$$H^{\otimes n} U_0 H^{\otimes n} (|x\rangle - |y\rangle) = |y\rangle - |x\rangle.$$



Now we briefly discuss what to do when there is more than one solution, and, moreover, their number

$$\ell = \#\{w : f(w) = 1\}$$

is not even known in advance.

But let us assume for a second that ℓ is known. What should we do? It depends on the value of ℓ . If ℓ is really large ($\ell \geq 10^{-2}N$), we apply the probabilistic algorithm. If ℓ is small ($\ell < 10^{-2}N$) then we apply a straightforward generalization. We replace w in $(|w\rangle, |\psi\rangle)$ with the sum of all good values of w . It is easy to check that

$$\sin \theta = \sqrt{\frac{\ell}{N}}.$$

We then apply Grover Iterate $\lfloor \sqrt{\frac{N}{\ell}} \frac{\pi}{4} \rfloor$ times.

In case the value of ℓ is not known, we fix a constant $C = 1.001 > 1$ and assume $\ell = 1, \lceil C \rceil, \lceil C^2 \rceil, \dots, \lceil C^t \rceil$, where $t = O(\log N)$. We then iterate Grover operator $\lfloor \sqrt{\frac{N}{\ell}} \frac{\pi}{4} \rfloor$ times for each value of ℓ and then distinguish

between good and bad answers. Clearly, as ℓ gets larger the number of iterations gets smaller,

$$\sqrt{\frac{N}{1}} + \sqrt{\frac{N}{C}} + \sqrt{\frac{N}{C^2}} + \dots = O(\sqrt{N}).$$

Even if we do not know the real value of ℓ , in one of our $O(\log N)$ experiments it will be guessed with sufficiently good accuracy.

Lectures 6-7

Scribe: Denis Pankratov, University of Chicago.

Date: January 25 and 27, 2011

4.2 Factoring: Shor's Algorithm

4.2.1 Reductions

In the *factoring problem* we are given a composite integer $N \in \mathbb{Z}$, and we are tasked with finding a nontrivial factor of N . Note that if we can solve this problem, we can completely factor any integer N in at most $\log N$ steps. Recall that primality testing is in P , as well as computing gcd of two numbers, so we assume that these procedures are readily available to us. First we show how to reduce factoring to the *order finding problem*: given $N, a \in \mathbb{N}$ such that $(N, a) = 1$, find minimum $r \in \mathbb{N}$ such that $r > 0$ and $a^r \equiv 1 \pmod{N}$. In the rest of this section on Shor's Algorithm, the notation r, a and N will be always used in this sense.

Suppose $N = p_1^{n_1} p_2^{n_2} \dots p_t^{n_t}$ for some primes p_i . We may assume that none of the p_i is 2, otherwise simply divide N by 2 as many times as possible. Furthermore we can check if N has only one prime divisor, i.e., if $N = p^z$ for some odd prime p and integer z , since in this case $z \leq \log N$ and we simply need to check that $N^{1/z}$ is an integer prime. Hence, in the rest we assume that N has all odd prime factors, and at least two different primes. We have $\mathbb{Z}_N^* \cong \mathbb{Z}_{p_1^{n_1}}^* \times \dots \times \mathbb{Z}_{p_t^{n_t}}^*$ and $\mathbb{Z}_{p_i^{n_i}}^* \cong \mathbb{Z}_{p_i^{n_i-1}(p_i-1)}$. Therefore if $a \in \mathbb{Z}_N^*$ we can write it as $a = (a_1, \dots, a_t)$ and $\text{ord}(a) = \text{lcm}(\text{ord}(a_1), \dots, \text{ord}(a_t))$. Now, assume that we have access to an order finding black box B , which on input N, a outputs the minimal r such that $a^r \equiv 1 \pmod{N}$. For an integer a chosen from \mathbb{Z}_N uniformly at random $P(B(N, a) \text{ is even}) \geq 1/2$. Keep picking a until we get an even r , i.e., $r = 2s$ for some $s \in \mathbb{Z}$. (Observe that if we accidentally pick a such that $(a, N) \neq 1$ we are done). Then we have $a^{2s} \equiv 1 \pmod{N}$ and $(a^s - 1)(a^s + 1) \equiv 0 \pmod{N}$. $a^s \not\equiv 1 \pmod{N}$ since r is minimal, and if either $(a^s - 1, N)$ or $(a^s + 1, N)$ is a nontrivial factor of N we

are done. The only problem occurs when $a^s \equiv -1 \pmod{N}$, but probability that this happens is at most $1/2^{t-1}$ (this is where we use the fact that N is not a prime power). This completes the reduction.

Instead of solving the order finding problem directly, we will develop a quantum algorithm to output a rational σ such that for some k (unknown to us), we have

$$\left| \sigma - \frac{k}{r} \right| < \frac{1}{2N^2}. \quad (1)$$

Claim 15. *Once we have σ (as above) we can reconstruct r .*

Proof. Take $\sigma - \lfloor \sigma \rfloor$, invert it and repeat to get the continued fraction expansion of σ :

$$\sigma = n_0 + \frac{1}{n_1 + \frac{1}{n_2 + \frac{1}{\dots}}}.$$

Note that since $\sigma \in \mathbb{Q}$, the above procedure converges. If we truncate the continued fractions of σ at some level we obtain k/r (see e.g. [1, Theorem A4.16]). \square

Claim 16. *There is at most one pair k', r' (in the reduced form) satisfying equation (1).*

Proof. Suppose that we have k', r' such that

$$\left| \sigma - \frac{k'}{r'} \right| < \frac{1}{2N^2}. \quad (2)$$

Then from (1) and (2) we obtain

$$\left| \frac{kr' - k'r}{rr'} \right| < \frac{1}{N^2}.$$

And consequently,

$$|kr' - k'r| < \frac{rr'}{N^2} \leq 1.$$

Since k, k', r , and r' are integers, we have $kr' = k'r$. It follows that $k = k'$ and $r = r'$, since $(k', r') = (k, r) = 1$. \square

4.2.2 Linear Algebra

First, we review some background from Linear Algebra.

Definition 17. A matrix $H \in M_n(\mathbb{C})$ is called *Hermitian* if $H = H^\dagger$.

Definition 18. A matrix $U \in M_n(\mathbb{C})$ is called *unitary* if $U^\dagger = U^{-1}$.

The above two notions are special cases of the following.

Definition 19. A matrix $A \in M_n(\mathbb{C})$ is called *normal* if it commutes with its adjoint, i.e. $AA^\dagger = A^\dagger A$.

Theorem 20 (Spectral Decomposition Theorem). *Any normal matrix A has a decomposition*

$$A = P\Lambda P^\dagger, \quad (3)$$

where P is unitary and Λ is diagonal.

Observe that Spectral Decomposition Theorem implies that the eigenvalues of Hermitian matrices are real, and eigenvalues of unitary matrices lie on a unit circle in the complex plane.

Given $N \leq 2^n$ and $a < N$ with $(a, N) = 1$, define operator U_a as follows.

$$U_a : |x\rangle \mapsto \begin{cases} |xa \bmod N\rangle & \text{if } x \in [0, N-1] \\ |x\rangle & \text{if } x \geq N \end{cases}$$

where $x \in \{0, 1\}^n$.

Observe that U_a is a permutation and is clearly computable in polynomial time. Since, U_a^r is an identity operator, all eigenvalues of U_a are r^{th} roots of unity, i.e., of the form $e^{2\pi i k/r}$. Now, we describe some of the eigenvectors of U_a that we will need for Shor's Algorithm (all others are obtained in a similar way by shifting this formula to cosets of the subgroup in \mathbb{Z}_N generated by a).

$$|u_k\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i k s/r} |a^s \bmod N\rangle$$

It is straightforward to check that $U_a|u_k\rangle = e^{2\pi i k/r}|u_k\rangle$. The rest of Shor's Algorithm splits into two parts:

Part 1: If we have eigenvectors $|u_k\rangle$, what do we do with them?

Part 2: How do we get vectors $|u_k\rangle$?

4.2.3 Part 1: Phase Estimation Algorithm

Consider a more general setting: given a unitary matrix U and an eigenvector $|\psi\rangle$ such that $U|\psi\rangle = e^{2\pi i\omega}|\psi\rangle$ for some $\omega \in \mathbb{R}$, estimate ω to arbitrary accuracy (think: $\omega = k/r, |\psi\rangle = |u_k\rangle, U = U_a$). It turns out we won't be able to solve it for arbitrary unitary operators. We need one more condition on U , which will come naturally as we develop the algorithm.

If U is a unitary operator acting on $|y\rangle$ define its *controlled version*, denoted by $c-U$ by

$$\begin{aligned} c-U|0\rangle|y\rangle &= |0\rangle|y\rangle \\ c-U|1\rangle|y\rangle &= |1\rangle U|y\rangle. \end{aligned}$$

Note that this generalizes previously defined notion of a controlled- f operator (in which case $c-U$ is simply a permutation matrix).

Observation 21. *If U is computable by a small circuit then $c-U$ is also computable by a small circuit.*

Proof. Note that for any two unitary operators U, V , we have $c-UV = (c-U)(c-V)$. Since our basis is universal, we can introduce new more complicated gates (controlled version of gates in the basis) and produce the desired circuit with a small increase in size. \square

Now, we want to generalize it further and construct $c-U^x$, which given $x \in \{0, 1\}^t$ (interpreted as an integer in binary) computes

$$c-U^x : |x\rangle|y\rangle \mapsto |x\rangle U^x|y\rangle.$$

The circuit shown in Figure 3 achieves this task (of all the gates shown on this picture, we keep those that correspond to 1 in the binary expansion of x).

Observe that in order for the above circuit to be small, we need U^{2^t} be efficiently computable. This is the additional requirement on the unitary matrix U we mentioned at the beginning of the section. Observe that in our case, $U_a^{2^t}|x\rangle = |xa^{2^t} \bmod N\rangle$ can be efficiently computed using the repeated squaring algorithm.

We will need one last ingredient for the phase estimation algorithm. It is the *quantum Fourier transform*, defined as follows.

$$\text{QFT}_m : |x\rangle \mapsto \frac{1}{\sqrt{m}} \sum_{y=0}^{m-1} e^{2\pi i xy/m} |y\rangle,$$

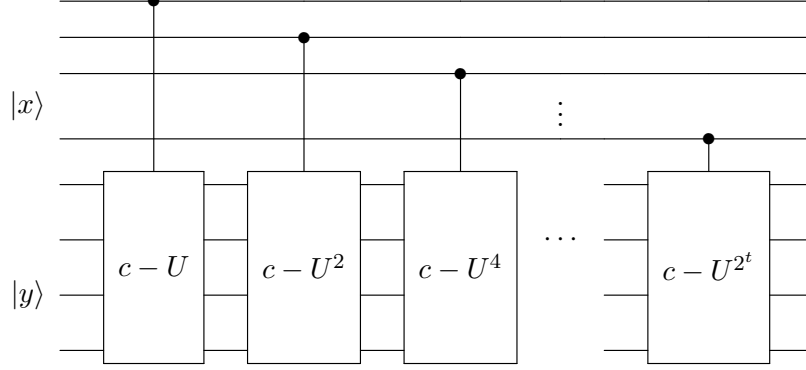


Figure 3: Circuit to compute $c - U^x$.

where $x, y \in \mathbb{Z}_m$ and $m \gg N$. It is easy to check that the inverse of this operator is defined in the following manner.

$$\text{QFT}_m^{-1} : |x\rangle \mapsto \frac{1}{\sqrt{m}} \sum_{y=0}^{m-1} e^{-2\pi i xy/m} |y\rangle.$$

In these notes we are omitting how to prepare QFT_m .

The circuit representing the phase estimation algorithm is shown in Figure 4.

Performing the computation, we obtain

$$\begin{aligned} |0\rangle|\psi\rangle &\mapsto \frac{1}{\sqrt{m}} \sum_{x=0}^{m-1} |x\rangle|\psi\rangle \\ &\mapsto \frac{1}{\sqrt{m}} \sum_x |x\rangle U^x |\psi\rangle \\ &= \frac{1}{\sqrt{m}} \sum_x e^{2\pi i \omega x} |x\rangle|\psi\rangle. \end{aligned}$$

Finally, we obtain

$$|0\rangle|\psi\rangle \mapsto \frac{1}{m} \sum_{x,y \in \mathbb{Z}_m} e^{2\pi i \omega x - 2\pi i xy/m} |y\rangle|\psi\rangle. \quad (4)$$

We measure the first register. Let $p(y)$ be the amplitude in front of $|y\rangle|\psi\rangle$, i.e.,

$$p(y) = \frac{1}{m} \sum_x e^{2\pi i x(\omega - y/m)}.$$

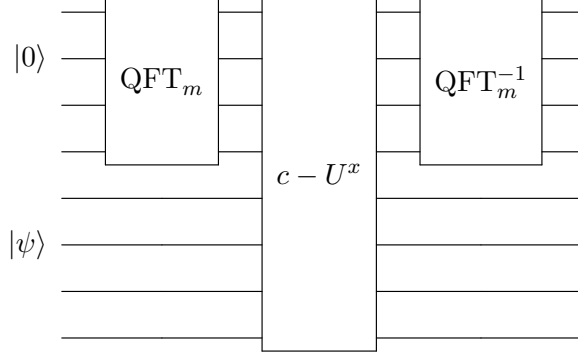


Figure 4: Phase estimation algorithm.

Clearly, there exists at least one y such that $|\omega - y/m| \leq 1/2m$. We provide an informal argument that for such y we have $p(y) > 0.1$ (an analytical expression in a closed form can be found e.g. in [2, Section 7.1.1]). Assume that $0 \leq \omega - y/m \leq 1/2m$ and consider a unit circle on the complex plane. Each term in the expression for $p(y)$ represents a unit vector rotated counter clockwise by an angle $2\pi(\omega - y/m)$. So after m steps, we'll move by an angle at most π . If it is in fact less than $\pi/2$, then the average of the terms will have a large real part. If it is greater than $\pi/2$, then the average of the terms will have a large imaginary part.

By choosing $m = N^2$ we obtain the desired result.

4.2.4 Part 2: How to Construct $|u_k\rangle$?

Recall, that the eigenvectors of interest to us are of the form

$$|u_k\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{2\pi i s k / r} |a^s \bmod N\rangle.$$

We cannot construct individual vectors $|u_k\rangle$, but we can construct their uniform superposition $1/\sqrt{r} \sum_{k=0}^{r-1} |u_k\rangle = |1\rangle$. We will get a uniform superposition of expressions in (4). There will be no cancellation between different values of k , because we measure only the first register. So we can use the Phase Estimation Algorithm with $|\psi\rangle = |1\rangle$, and we will get an estimate of a phase for a random value of k , but this is all we need for our purposes.

4.3 Discrete Logarithm

Another problem from the domain of cryptography is a so-called “Discrete Logarithm problem”: given $N, a, b \in \mathbb{N}$ with N a prime, find t such that $a^t \equiv b \pmod{N}$. Unlike cryptosystems based on factoring (when there is a small amount of “trapdoor” information allowing legitimate users to use it), this problem is believed to be hard even for Boolean circuits (that is, “no trapdoor” property).

To solve this problem efficiently on a quantum computer we will apply Shor’s order-finding algorithm (even though in this case the order can be easily computed as $r = N - 1$). Here we have two operators

$$U_a : |x\rangle \mapsto |xa \pmod{N}\rangle, \text{ and } U_b : |x\rangle \mapsto |xb \pmod{N}\rangle.$$

Note that if we apply order-finding algorithm to both U_a and U_b acting on a specific vector then we get a good estimate $\sigma \approx k/(N-1)$ and $\sigma' \approx \ell/(N-1)$. Since $b \equiv a^t \pmod{N}$ we have $U_b = U_a^t$ and $\ell = kt$. Consequently we can estimate $t \approx \sigma'/\sigma$. The only problem is that if we apply U_b after U_a we lose the vector $|u_k\rangle$. The solution to this problem is to apply U_a and U_b in parallel.

There is a physical justification for the validity of this argument. Namely, we can measure information partially. The part we measure gets destroyed, but we can continue with the rest as if nothing happened. This idea will be developed later.

The validity of the circuit solving the discrete logarithm problem can be also confirmed with the following direct computation:

$$|0\rangle|\psi\rangle|0\rangle \mapsto \frac{1}{m^2} \sum_{x_1, x_2, y_1, y_2 \in \mathbb{Z}} e^{2\pi i \omega_1 x_1 - 2\pi i x_1 y_1 / m} e^{2\pi i \omega_2 x_2 - 2\pi i x_2 y_2 / m} |y_1\rangle|\psi\rangle|y_2\rangle.$$

Here, $\omega_1 = k/(N-1)$ and $\omega_2 = kt/(N-1)$. Then the amplitude in front of $|y_1\rangle|\psi\rangle|y_2\rangle$ is

$$p(y_1, y_2) = \frac{1}{m^2} \sum_{x_1, x_2} p(y_1) p(y_2),$$

where $p(y_1)$ and $p(y_2)$ are as in Shor’s algorithm. Thus if $p(y_1) > 0.1$ and $p(y_2) > 0.1$ then $p(y_1, y_2) > 0.01$, so we can measure y_1, y_2 , take rounded value of y_2/y_1 as t , check if it works, and repeat if needed.

4.4 Hidden Subgroup Problem

The problems solved by Simon’s Algorithm, Shor’s Algorithm, and Discrete Logarithm Algorithm can be phrased as instances of a more general *Hidden*

Subgroup Problem.

Simon's Algorithm Given a finite abelian group $G = \mathbb{Z}_2^k$ and some function f such that for a subgroup $H \leq G$ of index 2 we have $f(x) = f(y)$ if and only if $x \in yH$, the goal is to find H . (yH denotes the y -coset of H).

Shor's Algorithm Given $G = \mathbb{Z}$ and $f(x) = a^x \bmod N$ the goal is to find a hidden subgroup $H = r\mathbb{Z}$. Again, $f(x) = f(y)$ if and only if $x \in yH$.

Discrete Logarithm Given $G = \mathbb{Z}_r \times \mathbb{Z}_r$ and $f(x, y) = a^x b^y \bmod N$, the goal is to find a hidden subgroup $H = \{(x, y) \mid x + ty = 0\}$ generated by $(-t, 1)$.

Theorem 22 (Kitaev, 95). *If G is a finitely generated abelian group and $H \leq G$ is a finite index subgroup then Hidden Subgroup Problem (HSP) for G is solvable by a polytime quantum algorithm.*

We will not give a proof of this theorem, it can be found e.g. in [3, Section 13.8].

A major open problem is to solve HSP for non-abelian groups. The progress for non-abelian case has been rather limited, but the motivation for studying non-abelian case is quite compelling. There are two great applications.

4.4.1 First Application - Symmetric Group

If HSP were solved for S_n (symmetric group of order $n!$), we could solve the graph isomorphism problem as follows. Given graphs G_1 and G_2 , each on n variables, consider group S_{2n} . For $\sigma \in S_{2n}$ define $f(\sigma) = \sigma(G_1 \cup G_2)$, i.e., σ acts on the vertices of $G_1 \cup G_2$ by permuting them. Then $f(\sigma_1) = f(\sigma_2)$ if and only if $\sigma_2^{-1}\sigma_1 \in \text{Aut}(G_1 \cup G_2)$. Once we know $\text{Aut}(G_1 \cup G_2)$ (say, by a list of generators L) we can decide if two graphs are isomorphic by checking whether there exists a permutation in L that moves all vertices from G_1 to G_2 .

4.4.2 Second Application - Dihedral Group

Dihedral group, denoted D_{2n} , is defined as a group of symmetries of a regular n -gon. The order of D_{2n} is $2n$. Let r be a counter clockwise rotation by $2\pi/n$ counter clockwise, and s be a reflection through vertex 1 and $n/2$ if n is even or the center of the opposite edge if n is odd. Then D_{2n} consists of r^i

and sr^i for $0 \leq i \leq n-1$. In a sense, D_{2n} is very close to being abelian, since $[D_{2n} : \mathbb{Z}_n] = 2$. Observe that D_{2n} contains many involutions (subgroups of order 2), and it is not known if one can detect a subgroup of order 2.

Shortest Vector Problem (SVP) in lattices $\mathbb{Z}^n \subset \mathbb{R}^n$ is to find a shortest non-zero vector in a lattice, i.e. $\min\{|v| \mid v \in \mathbb{Z}^n \setminus \{0\}\}$. Ajtai and Dwork [4] showed how to create a public-key cryptographic system whose security could be proven using only worst-case hardness of a certain version of SVP. This was the first result that used worst-case hardness to create secure systems. However, if you can solve HSP (in a sense) for D_{2n} then you can break SVP (almost) [5]. Let us describe one of the technicalities here.

Most of the current approaches to HSP for non-abelian groups use the operator U_f only via the following algorithm known as the *coset sampling algorithm* (that is a reasonable assumption due to the absolutely generic nature of the function f). Consider $f : G \rightarrow Z$, where G is a group and Z is an arbitrary set, $|Z| = N$, and a subgroup $H \leq G$, $L = |H|$. Since $U_f : |x, 0\rangle \mapsto |x, f(x)\rangle$, we have $U_f : 1/\sqrt{N} \sum_x |x, 0\rangle \mapsto 1/\sqrt{N} \sum_x |x, f(x)\rangle$. We “measure the second register” and obtain value of $f(x) = y$, we then continue the computation. Intuitively, we expect to obtain a uniform superposition of all x in a “random” coset of H , i.e. $1/\sqrt{L} \sum_{f(x)=y} |x\rangle$.

In the next section we show how to make these notions precise.

Lecture 8

Scribe: Kenley Pelzer, University of Chicago.

Date: February 1, 2011

5 Quantum Probability

Deficiencies of the current formalism of unitary operators:

1. Probability distributions (dealing with randomness) over pure states need to be considered.

2. A problem with building quantum computers is the issues with noise and decoherence; we need a way to describe quantum noise (because no system is completely isolated from the environment). The unitary model is not up to the challenge; we need to consider mixed states.

3. We need to consider partial measurement (tracing out).

We start with a set of unitary (pure) states and their probabilities:

$$(p_1, |\psi_1\rangle), (p_2, |\psi_2\rangle), \dots$$

Each $|\psi\rangle$ is also an exponential sum:

$$|\psi\rangle = \sum_x \alpha_x |x\rangle.$$

This is messy, so we want something more concise: there is an invariant that we can work with.

If two (possibly mixed) states have the same invariant, they are physically indistinguishable (in our world, this means that they are *computationally* indistinguishable).

A density matrix is such an invariant.

$|\psi\rangle|\psi\rangle \leftarrow$ tensor product (unit vector in a larger space)

$\langle\psi|\psi\rangle \leftarrow$ scalar product

$\rho_\psi = |\psi\rangle\langle\psi| =$ density operator for state ψ (also called the “outer product”)

$$\rho_\psi(x, y) = \alpha_x \alpha_y^* \text{ if } |\psi\rangle = \sum_x \alpha_x |x\rangle.$$

Three important properties of density matrices:

- (a) Any density matrix is Hermitian.
- (b) Trace of a density matrix is 1.
- (c) A density matrix is positive semidefinite (its eigenvalues are non-negative).

Definition: A density matrix is any square matrix that satisfies all of the conditions (a)-(c) listed above.

If we take a convex combination of density matrices, we get another density matrix.

So we can sum density matrices with corresponding probabilities:

$$\rho = p_1 |\psi_1\rangle\langle\psi_1| + \dots + p_t |\psi_t\rangle\langle\psi_t|.$$

ρ is a density matrix. We apply a unitary operation on both sides:

$$U \rho U^\dagger = U |\psi\rangle\langle\psi| U^\dagger = |U\psi\rangle\langle U\psi| \text{ (by definition of bra-ket rules)}$$

We can do many more great things now, like half of a unitary operation, giving

$$\rho \mapsto \frac{1}{2}\rho + \frac{1}{2}U\rho U^\dagger.$$

Another thing is *depolarization at the rate η* defined as

$$\mathcal{E}_\eta(\rho) = (1 - \eta)\rho + \frac{\eta}{N}I_N,$$

where I_N is the identity matrix.

Many more noise channels can be found in [1, Section 8.3]; some of them will also be considered in Section 8 below.

Another way to create the identity matrix is to say that each state occurs with probability $\frac{1}{N}$ if there are N states. This is the mathematical equivalent of a “completely depolarized” (or “totally random”) state.

Two things to NOT mix up:

$$|x_1\rangle, p = \frac{1}{N}, |x_2\rangle, p = \frac{1}{N}, \dots, |x_N\rangle, p = \frac{1}{N} \text{ with density matrix } \frac{1}{N}I_N$$

versus the *uniform superposition*

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

with density matrix

$$\frac{1}{N} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

If you apply a measurement immediately, these two density matrices give equivalent results, but after applying a unitary operator, we get very different results (as we have already seen many times before).

5.1 “Tracing out” or “partial measurement”

Take the pure state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Upon measuring (and then discarding) the second register, we intuitively should get the mixed state

$$\left(|0\rangle, \frac{1}{2}\right), \left(|1\rangle, \frac{1}{2}\right).$$

In the vector space $\mathcal{H}_A \otimes \mathcal{H}_B$,

$$\psi = \sum_{a,b} \alpha_{ab} |a\rangle |b\rangle$$

$$\sum_{a,b} |\alpha_{ab}|^2 = 1.$$

Denoting $\sum_a |\alpha_{ab}|^2$ by p_b , with probability p_b we get

$$\frac{1}{\sqrt{p_b}} \sum_a \alpha_{ab} |a\rangle.$$

$$\sum_b \left[p_b \sum_{a_1, a_2} \left(\frac{1}{p_b} \alpha_{a_1, b} \alpha_{a_2, b}^* |a_1\rangle \langle a_2| \right) \right] =$$

This corresponds to the density matrix

$$\sum_{a_1, a_2, b} \alpha_{a_1 b} \alpha_{a_2 b}^* |a_1\rangle \langle a_2|$$

$$= \sum_{a_1, a_2, b_1, b_2} (\alpha_{a_1 b_1} \alpha_{a_2 b_2}^* |a_1\rangle \langle a_2| \langle b_1| b_2\rangle).$$

Thus, $\text{Tr}_B (|a_1\rangle \langle a_2| \otimes |b_1\rangle \langle b_2|)$ “should” be defined as

$$|a_1\rangle \langle a_2| \cdot \langle b_2| b_1\rangle.$$

This operation is called “tracing out”. It is a good exercise to check that this operator indeed takes density matrices to density matrices.

5.2 Superoperators

All examples of quantum operations we have seen so far share the following properties: they are linear operators that act on matrices, take matrices of one size to matrices of another (possibly, different) size and take density matrices to density matrices. This is “almost” the right definition of a *superoperator* or an operator “physically realizable” in nature, for the completely right one see e.g. [1, Section 8.2.4]. We will see one more (and sometimes more useful) definition in Section 8.

A superoperator is not necessarily reversible.

Lecture 9

Scribes: Kenley Pelzer and Tatiana Orlova, University of Chicago.

Date: February 8, 2011

If you want to measure noise, we need to know distance between two different states.

In unitary world, pure states are just unit vectors, so ”distance” is angle between them (there’s hardly any other choice).

Probability distributions:

Statistical difference (ℓ_1) is represented by a diagonal matrix:

$$\begin{pmatrix} p_1 & 0 & 0 & 0 \\ 0 & p_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & p_n \end{pmatrix} - \begin{pmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & q_n \end{pmatrix} = \begin{pmatrix} a_1 & 0 & 0 & 0 \\ 0 & a_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & a_n \end{pmatrix}.$$

Given an event E , the difference between the probability of the event E in two different distributions can be bounded by:

$$|P_p(E) - P_q(E)| \leq \frac{1}{2} \sum_{i=1}^N |a_i|$$

This is the statistical distance between two distributions; note that we must take absolute value since

$$\sum_{i=1}^N a_i = 0.$$

Assume now that ρ and σ are density matrices. It would be tempting to define the *trace distance* between them simply as

$$D(\rho, \sigma) = \text{Tr}(|\rho - \sigma|),$$

but we want to be able to measure in an arbitrary basis. Thus, we let

$$D(\rho, \sigma) = \max_U \text{Tr}(|U(\rho - \sigma)U^\dagger|),$$

where the maximum is taken over all unitary matrices U .

Theorem 23. *If T is an arbitrary superoperator, then for any pair of density matrices:*

$$D(T(\rho), T(\sigma)) \leq D(\rho, \sigma).$$

The proof is omitted and can be found e.g. in [1, Section 9.2.1].

6 Quantum Complexity Theory: black-box model

In the general black-box problem, we are typically given a function $f : [N] \rightarrow \{0, 1\}$. While the set $[N]$ can actually be of arbitrary nature, in many interesting cases it is comprised of binary strings. To commemorate this fact, we use lower case letters x, y, z etc. for its elements (and we will typically represent the function f by its truth-table $X = (X_1, \dots, X_N)$, where $X_x = f(x)$).

6.1 Hybrid method: optimality of Grover's search

In the search problem we want to find $x \in \{1, 2, \dots, N\}$ such that $f(x) = 1$. We have shown that Grover's search algorithm solves this problem by making $O(\sqrt{N})$ queries to the black-box U_f (see Theorem 14 in Section 4). We now show that this result is the best possible.

Theorem 24. *Grover's search algorithm is optimal, i.e., every quantum black-box search algorithm requires $\Omega(\sqrt{N})$ queries.*

The proof of the above theorem follows directly from the same lower bound for the corresponding decision problem. We now state the decision problem and then prove the lower bound for it.

Let $X \stackrel{df}{=} (X_1, \dots, X_N)$, where $X_x \in \{0, 1\}$, such that $X_x = f(x)$ for all $x \in [N]$. We will denote by X_0 an all-zero string, i.e. the string (X_1, \dots, X_N) , such that $X_x = 0$ for all $x \in [N]$, and by \mathbf{X}_x the string (X_1, \dots, X_N) , such that $X_y = \begin{cases} 1, & \text{if } y = x; \\ 0, & \text{otherwise} \end{cases}$. In other words, \mathbf{X}_x is the string that contains precisely one 1 in the x^{th} place. We want to compute the following function

$$F(X) \stackrel{df}{=} \begin{cases} 0, & \text{if } X \equiv X_0; \\ 1, & \text{if } X \in \{\mathbf{X}_x\}_{x \in [N]}. \end{cases}$$

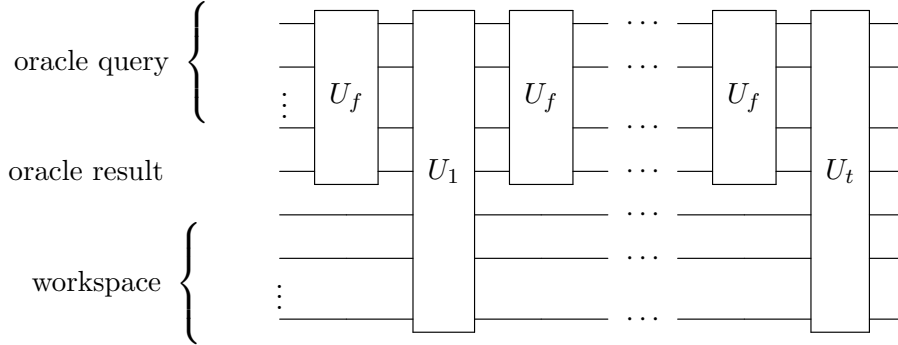


Figure 5: Quantum circuit for black-box query computation.

Theorem 25. *Computing $F(X)$ requires $\Omega(\sqrt{N})$ queries to the black box U_f .*

Proof. Let

$$|\psi_j^x\rangle \stackrel{df}{=} U_j U_{\mathbf{X}_x} U_{j-1} \dots U_1 U_{\mathbf{X}_x} |\psi\rangle,$$

and

$$|\psi_j\rangle \stackrel{df}{=} U_j U_{j-1} \dots U_1 |\psi\rangle$$

(note that U_{X_0} is the identity operator).

We want to look at the distance $\| |\psi_j^x\rangle - |\psi_j\rangle \|$ (where $\|\cdot\|$ stands for Euclidean norm). For $j = 0$ this distance is 0. When j goes from 0 to t , the distance must change from 0 to 1% for any fixed x as our circuit must be able to distinguish between X_0 and \mathbf{X}_x . We want to prove that the distance cannot change by more than a certain amount that depends on t . This will bound the number of times we have to apply the operator U_f in order to successfully solve the problem.

First of all, since unitary operators preserve distances, we can assume w.l.o.g. that U_{j+1} is the identity operator which implies $|\psi_{j+1}\rangle = |\psi_j\rangle$. Let $|\psi_{j+1}\rangle = |\psi_j\rangle = \sum_y \alpha_{y,j} |y\rangle |\phi_y\rangle$, where $\alpha_{y,j} \geq 0$, $\sum_y \alpha_{y,j}^2 = 1$ and $\|\phi_y\| = 1$. On Figure 5, $|y\rangle$ corresponds to the first register, and $|\phi_y\rangle$ is the combination of oracle result and workspace. Also, let $|\psi_j^x\rangle = \sum_y \alpha_{y,j}^x |y\rangle |\phi_y^x\rangle$, and $|\psi_{j+1}^x\rangle = \sum_y \alpha_{y,j+1}^x |y\rangle |\tilde{\phi}_y^x\rangle$, where again $\alpha_{y,j}^x \geq 0$, coefficients do not change, i.e. $\alpha_{y,j+1}^x = \alpha_{y,j}^x$ for all y , and $|\tilde{\phi}_y^x\rangle = |\phi_y^x\rangle$ unless $y = x$. For $y = x$, we have no control over what happens to $|\tilde{\phi}_x^x\rangle$ when changing from the state $|\psi_j^x\rangle$ to $|\psi_{j+1}^x\rangle$, except for the fact that the length is preserved (and it will be important).

The first obvious idea to try is the triangle inequality

$$\| |\phi_{j+1}\rangle - |\phi_{j+1}^x\rangle \| \leq \| |\phi_j\rangle - |\phi_j^x\rangle \| + \| |\phi_j^x\rangle - |\phi_{j+1}^x\rangle \| \leq \| |\phi_j\rangle - |\phi_j^x\rangle \| + 2\alpha_{x,j}^x.$$

That is good, but for certain reasons that will become clear later, we would like the above inequality to depend only on values $|\alpha_{x,j}|$ (without the superscript x). For this purpose we split our Hilbert space \mathcal{L} into a direct sum of two subspaces

$$\mathcal{L} = \left(\bigoplus_{y \neq x} \mathcal{L}_y \right) \oplus \mathcal{L}_x.$$

In the subspace $\bigoplus_{y \neq x} \mathcal{L}_y$ all operators act identically. In the subspace \mathcal{L}_x we have

$$\| \alpha_{x,j} |\phi_x\rangle - \alpha_{x,j}^x |\phi_x^x\rangle \| + \alpha_{x,j} \geq \alpha_{x,j}^x \geq \| \alpha_{x,j} |\phi_x\rangle - \alpha_{x,j+1}^x |\tilde{\phi}_x^x\rangle \| - \alpha_{x,j}.$$

(We did use for this calculation that $\alpha_{x,j+1}^x = \alpha_{x,j}^x$!) This gives us the desired bound in \mathcal{L}

$$\| |\psi_{j+1}\rangle - |\psi_{j+1}^x\rangle \| \leq \| |\psi_j\rangle - |\psi_j^x\rangle \| + 2\alpha_{x,j}$$

(An exercise!) Summing this over all x and taking into account the Cauchy-Schwartz inequality $\sum_x \alpha_{x,j} \leq \sqrt{N \cdot \sum_x \alpha_{x,j}^2} = \sqrt{N}$, we get the estimate

$$\sum_x \| |\psi_{j+1}\rangle - |\psi_{j+1}^x\rangle \| \leq \sum_x \| |\psi_j\rangle - |\psi_j^x\rangle \| + 2\sqrt{N}.$$

Then, by induction,

$$\sum_x \| |\psi_t^x\rangle - |\psi^x\rangle \| \leq 2t\sqrt{N}.$$

Since the left-hand side, as we already observed, is $\Omega(N)$, the proof is completed. \square

Lectures 10 and 11

Scribe: Philip Reinhold, University of Chicago

Date: February 10 and 15, 2011

6.2 Quantum Query Complexity vs. Other Complexity Measures

While Simon's Algorithm demonstrates the feasibility of exponential quantum speedup in black-box query complexity for at least one problem, another approach has shown that for another class of problems, the best one can achieve is a polynomial speedup. Namely, Simon's problem dealt with a predicate which was only defined on certain inputs, specifically that the input function f satisfied $\exists s \forall x \forall y (f(x) = f(y) \leftrightarrow x = y \vee x \oplus y = s)$. In the case that the input to Simon's algorithm did not satisfy this promise, the output is not well defined. We will now see that when we forbid the last feature, the situation changes dramatically.

Definition 26. A property $F : \{0,1\}^N \rightarrow \{0,1\}$ is *total* if its output is defined for all inputs $\{0,1\}^N$.

Definition 27. For a property F under the black-box model, $D(F)$ is the *deterministic complexity*, i.e., the number of calls to the black box that must be made (in the worst-case) with a deterministic classical algorithm to determine the property.

Note that in this definition we do not count the internal work of the algorithm, only the number of queries.

Definition 28. For a property F under the black-box model, $Q_2(F)$ is the *bounded-error quantum complexity*, i.e., the number of calls to the black box that must be made with a quantum computer such that the probability of returning the correct answer is at least $2/3$.

We have the following theorem relating these two (and a few other) measures.

Theorem 29. For any total property F , $D(F) \leq O(Q_2(F)^6)$.

Proof. We have

$$\begin{aligned} \deg(F) &\leq D(F) \leq O(C^{(1)}(F)\text{bs}(F)) \\ &\leq O(s(F)\text{bs}(F)^2) \\ &\leq O(\text{bs}(F)^3) \\ &\leq O(\widetilde{\deg(F)}^6) \\ &\leq O(Q_2(F)^6). \end{aligned}$$

□

Fleshing out the content of this proof will be the following supporting concepts and theorems; for historical attributions of all these pieces see e.g. the survey [6].

Definition 30. A *one-certificate* for $F : \{0, 1\}^N \rightarrow \{0, 1\}$ is an assignment $c : S \rightarrow \{0, 1\}$ for some $S \subseteq [N]$ such that for all inputs X that are consistent with c , $F(X) = 1$. An input to F , X is consistent with c iff $\forall i \in S (X_i = c_i)$.

The *one-certificate complexity*, $C^{(1)}(F)$ is the minimum value such that for all inputs X on which $F(X)$ is true, there exists a certificate c such that X is consistent with c and $|c| \leq C^{(1)}(F)$.

Definition 31. A function $F : \{0, 1\}^N \rightarrow \{0, 1\}$ is *sensitive* on $B \subseteq [N]$ for input X iff flipping the values X_v for $v \in B$ flips the output of F , i.e., $F(X) \neq F(X \oplus B)$. Let the *block sensitivity* of F , $\text{bs}(F)$ be the size of the largest set of **pairwise disjoint** non-empty blocks B_i such that for some input X , F is sensitive on B_i for X , for i from 1 to $\text{bs}(F)$.

Theorem 32. $D(F) \leq C^{(1)}(F)\text{bs}(F)$.

Proof. We show this by presenting an algorithm whose steps are based on a single certificate (that is, use at most $C^{(1)}$ queries), which converges on an answer after at most $\text{bs}(F)$ steps.

At each stage we pick a certificate $c : S \rightarrow \{0, 1\}$ of size at most $C^{(1)}(F)$ which is consistent with those X_i already queried (if there is no consistent c , output 0 and stop). We then query X_v for all previously unqueried $v \in S$. If X is consistent with c , output 1 and stop. If we have not terminated after $\text{bs}(F)$ steps, pick any input Y consistent with those X_v queried, and output $F(Y)$.

The claim is that for any two inputs Y, Y' as above we necessarily have $F(Y) = F(Y')$ (and thus in particular $F(Y) = F(X)$). If we assume not, we can show that there are disjoint subsets of the input on which F is sensitive, B_1, B_2, \dots, B_{b+1} , where $b = \text{bs}(F)$.

Let c_i for $i \in [b]$ be the certificates whose indices were queried in the algorithm. Let Y, Y' be as above; assume w.l.o.g. that $F(Y) = 0, F(Y') = 1$. Let the certificate for Y' be c_{b+1} . Let B_i for $i \in [b+1]$ be the set of variables on which c_i and Y disagree. Then, $\forall i (F(Y \oplus B_i) = 1)$, which shows that F is sensitive on B_i at the input Y . To show that these sets are pairwise disjoint, consider two certificates used, c_i and c_j , where $i < j$. For all variables $v \in B_i$, $X_v = Y_v \neq c_i(v)$. However, having queried v in step i , the certificate c_j would be picked to be consistent on v , so even if v is in the domain of c_j , then $X_v = c_j(v)$. Hence, in any case we have $v \notin B_j$. Therefore, B_i s form a disjoint set of $b+1$ blocks on which F is sensitive at the input y , which is a contradiction. \square

Definition 33. The *sensitivity* of F , $s(F)$ is the number of variables of the input X on which a flip guarantees a flip in $F(X)$. It is equivalent to the block sensitivity with the additional restriction that $|B_i| = 1$, so $s(F) \leq \text{bs}(F)$.

Theorem 34. $C^{(1)}(F) \leq O(s(F)\text{bs}(F))$.

Proof. Let $\text{bs}_X(F) \leq \text{bs}(F)$ be the size of a maximal set of disjoint blocks B_i such that F is sensitive for X on all B_i . Furthermore, let these blocks be minimal in $|B_i|$. It follows that $c : \bigcup_i B_i \rightarrow \{0, 1\}$, $c(i) = X_i$, is a certificate for X , since if not, there would be another sensitive block $B_{\text{bs}_X(F)+1}$ defined as those input variables not in $\bigcup_i B_i$ on which X is still sensitive. Since these blocks are minimally sized, $X \oplus B_i$ must be sensitive on v for all $v \in B_i$, so

$|B_i| \leq s_{X \oplus B_i}(F) \leq s(F)$. Hence

$$|c| = \sum_{i=1}^{\text{bs}(F)} |B_i| \leq s(F) \text{bs}(F).$$

□

It is a big open problem to determine whether $s(F)$ and $\text{bs}(F)$ are always polynomially related; for a comprehensive survey on this problem see [7].

The next theorem requires the symmetrization technique. Let $p : \mathbb{R}^N \rightarrow \mathbb{R}$ be a multilinear polynomial. Let a permutation $\pi \in S_N$ be a rearrangement of the variables composing the input to p , i.e., $\pi(X) = (X_{\pi_1}, X_{\pi_2}, \dots, X_{\pi_N})$. The *symmetrization* of p is the average of p over all permutations of the inputs, i.e.

$$p^{\text{sym}}(X) = \frac{\sum_{\pi \in S_n} p(\pi(X))}{N!}.$$

Lemma 35. $p^{\text{sym}}(X)$ can be equivalently (for $X \in \{0, 1\}^N$) written as a single-variate polynomial $q(|X|)$.

Proof. Let $p(X) : \mathbb{R}^N \rightarrow \mathbb{R}$ be a multilinear polynomial. Let P_j denote the sum of all products of j input variables X_i . There are $\binom{N}{j}$ terms in this sum. Since p^{sym} is symmetrical, it can be written as

$$p^{\text{sym}}(X) = a_0 + \sum_{i=1}^N a_i P_i.$$

On inputs $X \in \{0, 1\}^N$, the only terms contributing to the sum V_i are those which are 1. With $|X| \equiv \sum_i X_i$, there are $\binom{|X|}{i}$ such terms, leaving P_j with the same value. Thus

$$p^{\text{sym}}(X) = q(|X|) \equiv a_0 + \sum_i a_i \binom{|X|}{i}.$$

□

Definition 36. The *approximate degree* of F , $\widetilde{\deg}(F)$ is the smallest degree of a multi-linear polynomial which approximates F . More formally

$$\widetilde{\deg}(F) = \min_p \left\{ \deg(p) \mid \forall x \in \{0, 1\}^N, |p(x) - F(x)| \leq \frac{1}{3} \right\}.$$

Furthermore, this theorem relies on a result of Ehlich, Zeller, Rivlin and Cheney.

Lemma 37. *If a polynomial p is bounded, i.e., $\forall i \in [N], b_1 \leq p(i) \leq b_2$, and $\exists x \in [0, 1](|p'(x)| \geq c)$ then*

$$\deg(p) \geq \sqrt{\frac{cN}{c + b_2 - b_1}}.$$

Theorem 38. $bs(F) \leq O(\widetilde{\deg}(F)^2)$.

Proof. Let p be a polynomial that approximates F , and let B_i be a family of $bs(F) = b$ blocks on which F is sensitive. Let $Y = (Y_1, \dots, Y_b)$ be a b -variate variable. For some input X where $F(X) = 0$, define $Z = (Z_1, \dots, Z_N)$ such that $Z_v = X_v \oplus Y_j$ if $v \in B_j$, and $Z_v = X_v$ if $v \notin B_1 \cup \dots \cup B_b$ (thus, in particular, when $Y = \vec{0}$, $Z = X$). Define $q(Y) = p(Z)$, making $q(Y)$ a b -variate polynomial of degree $\deg(p)$.

Note that since p is bounded in $\{0, 1\}^N$, so is q (in $\{0, 1\}^b$). Furthermore

$$|q(\vec{0}) - 0| = |p(X) - F(X)| \leq 1/3$$

(by the definition of $\widetilde{\deg}(F)$), and for any input Y with Hamming weight $|Y| = 1$, we have

$$|q(Y) - 1| = |p(X \oplus B_i) - F(X \oplus B_i)| \leq 1/3,$$

since $F(X)$ flips when flipping block B_i .

Let $r(|Y|) = q^{\text{sym}}(Y)$ be the single-variate polynomial obtained from symmetrizing q as in Lemma 35. Since r is obtained as an average over the possible inputs of q , the aforementioned properties translate to r , namely $r(n) \in [0, 1]$ ($n \in \{0, 1, \dots, N\}$), $r(0) \leq 1/3$, $r(1) \geq 2/3$. By the mean value theorem we have $|p'(x)| \geq 1/3$ for some x in $[0, 1]$.

Using this value as c , and the $[-1/3, 4/3]$ bound for r in Lemma 37 we have $\sqrt{\frac{\frac{1}{3}b}{\frac{1}{3} + 14/3 - (-1/3)}} = \sqrt{\frac{b}{6}} \leq \deg(r) \leq \deg(p)$. So $bs(F) \leq O(\widetilde{\deg}(F)^2)$. \square

Theorem 39. $\widetilde{\deg}(F) \leq O(Q_2(F))$.

Proof. Write a quantum circuit as an alternating series of arbitrary unitary transformations (U_j) and queries to the input X (U_X), see Figure 5. The output of this circuit can be written as $\sum_{k \in K} \alpha_k^X |k\rangle$ where K is the set of possible output strings. We first claim that for any fixed k , α_k^X can be

written as a multilinear polynomial in the variables $X = (X_1, \dots, X_N)$ with $\deg(\alpha_k^X) \leq Q_2(F)$.

Write the state of the circuit just after applying the j^{th} query to X as $|\psi_j\rangle$. Then $|\psi_{j+1}\rangle = U_j U_X |\psi_j\rangle$ (cf. the proof of Theorem 25). Write without loss of generality that the query U_X maps $|k\rangle = |y, b, z\rangle$ to $|y, b \oplus X_y, z\rangle$. Then we have the formula

$$U_X \left(\sum_{y,b,z} \alpha_{y,b,z}^X |y, b, z\rangle \right) = \sum_{y,b,z} (\alpha_{y,b,z}^X (1 - X_y) + \alpha_{y,b \oplus 1, z}^X X_y) |y, b, z\rangle,$$

which implies that the degree of the polynomials $\alpha_{y,b,z}^X$ can increase by at most 1 from applying the operator U_X . And the unitary transformation U_j does not depend on X at all, therefore it will output a polynomial of the same degree as its input.

It follows that the amplitudes α_t^X at the t^{th} step will be a polynomial of degree at most t . The probability of observing any basic state is $p_k = |\alpha_k^X|^2$, a polynomial of degree at most $2t$. \square

Remark 1. Whether the bound of Theorem 29 can be improved is a major open problem. Some of the intermediate steps in this chain of inequalities are known to be tight, and some are not, we again refer to the survey [6] for further details.

Remark 2. The bound in Theorem 39 is called *polynomial method*. This is our second technique for proving quantum lower bounds in the black-box model; we will review one more in the next lecture.

Lectures 12 and 13

Scribe: Pooya Hatami, University of Chicago

Date: February 16 and 17, 2011

6.3 Ambainis's Adversary Method

Let $F : \{0, 1\}^N \rightarrow \{0, 1\}$ be a Boolean function. Consider sets $\mathcal{X}, \mathcal{Y} \subseteq \{0, 1\}^N$ such that:

- $\forall X \in \mathcal{X} : F(X) = 1$, and
- $\forall Y \in \mathcal{Y} : F(Y) = 0$.

Let $R \subseteq \mathcal{X} \times \mathcal{Y}$ be a binary relation. Defining b and b' as follows

$$b = \max_{X \in \mathcal{X}, z \in [n]} |\{Y \mid R(X, Y) \wedge X_z \neq Y_z\}|,$$

$$b' = \max_{Y \in \mathcal{Y}, z \in [n]} |\{X | R(X, Y) \wedge X_z \neq Y_z\}|,$$

we have the following theorem.

Theorem 40 (Ambainis).

$$Q_2(F) \geq \Omega\left(\frac{|R|}{\sqrt{|\mathcal{X}||\mathcal{Y}|bb'}}\right).$$

Proof. Let $|\psi_j^X\rangle$ be the state just after the j^{th} call to the oracle when the computation is led by X . Similar to what we have seen in the Hybrid Method (Theorem 25) and Polynomial Method (Theorem 39), we have

$$|\psi_j^X\rangle = \sum_z \alpha_{z,j}^X |z\rangle |\phi_{z,j}^X\rangle.$$

We will study the following sum

$$W_j := \sum_{(X,Y) \in R} |\langle \psi_j^X | \psi_j^Y \rangle|.$$

Notice that $W_0 = |R|$.

Suppose the algorithm \mathcal{A} has the property that for any input Z the probability of guessing the correct answer is at least $1 - \epsilon$. This means that the final stage of \mathcal{A} can correctly distinguish $|\psi_t^X\rangle$ from $|\psi_t^Y\rangle$ for any X, Y with $F(X) \neq F(Y)$ (in particular, when $(X, Y) \in R$) with probability at least $1 - \epsilon$. Intuitively, it should be clear that it implies that the states $|\psi_t^X\rangle$ and $|\psi_t^Y\rangle$ can not be too close to each other; the following theorem makes this intuition precise.

Theorem 41 ([2, Theorem 9.2.1]). *Any procedure that on input $|\psi_Z\rangle$ guesses whether $Z = X$ or $Z = Y$ will guess correctly with probability at most $1 - \epsilon = \frac{1}{2} + \frac{1}{2}\sqrt{1 - \delta^2}$, where $\delta = |\langle \psi_X | \psi_Y \rangle|$. This probability is achievable by an optimal measurement.*

By the above theorem we know that we must have

$$|\langle \psi_t^X | \psi_t^Y \rangle| = \delta \leq 2\sqrt{\epsilon(1 - \epsilon)},$$

and thus $W_t \leq 2\sqrt{\epsilon(1 - \epsilon)}|R|$, which for $\epsilon < 1/2$ is less than $|R|$. Therefore it suffices to prove that $W_{j+1} \geq W_j - 2\sqrt{|\mathcal{X}||\mathcal{Y}|bb'}$.

Lemma 42. $W_{j+1} \geq W_j - 2\sqrt{|\mathcal{X}||\mathcal{Y}|bb'}$.

Proof. From the definition of W_j we know that

$$|W_{j+1} - W_j| \leq \sum_{(X,Y) \in R} |\langle \psi_{j+1}^X | \psi_{j+1}^Y \rangle - \langle \psi_j^X | \psi_j^Y \rangle|.$$

We also know that

$$|\psi_{j+1}^X\rangle = |\psi_j^X\rangle - 2 \sum_{z: X_z=1} \alpha_{z,j}^X |z\rangle |\phi_{z,j}^X\rangle,$$

and

$$|\psi_{j+1}^Y\rangle = |\psi_j^Y\rangle - 2 \sum_{z: Y_z=1} \alpha_{z,j}^Y |z\rangle |\phi_{z,j}^Y\rangle$$

(as in the similar argument on page 40, we assume w.l.o.g. that the interlacing constant operator U_{j+1} is identity). Note that unlike the Hybrid proof (see page 40), we do use here the specific form of the operators U_X, U_Y , although I suspect it can be avoided.

It follows that

$$|\langle \psi_{j+1}^X | \psi_{j+1}^Y \rangle - \langle \psi_j^X | \psi_j^Y \rangle| \leq 2 \sum_{z: X_z \neq Y_z} |\alpha_{z,j}^X| |\alpha_{z,j}^Y| |\langle \phi_{z,j}^X | \phi_{z,j}^Y \rangle|.$$

Thus it suffices to bound

$$2 \sum_{(X,Y) \in R} \sum_{z: X_z \neq Y_z} |\alpha_{z,j}^X| |\alpha_{z,j}^Y| \leq 2\sqrt{|\mathcal{X}||\mathcal{Y}|bb'}.$$

We know that

$$2|\alpha_{z,j}^X| |\alpha_{z,j}^Y| \leq r|\alpha_{z,j}^X|^2 + \frac{1}{r}|\alpha_{z,j}^Y|^2,$$

thus providing

$$\begin{aligned} & 2 \sum_{(X,Y) \in R} \sum_{z: X_z \neq Y_z} |\alpha_{z,j}^X| |\alpha_{z,j}^Y| \\ & \leq r \sum_{(X,Y) \in R} \sum_{z: X_z \neq Y_z} |\alpha_{z,j}^X|^2 + \frac{1}{r} \sum_{(X,Y) \in R} \sum_{z: X_z \neq Y_z} |\alpha_{z,j}^Y|^2 \leq rb|\mathcal{X}| + \frac{1}{r}b'|\mathcal{Y}|. \end{aligned}$$

Finally choosing $r = \sqrt{\frac{b'|\mathcal{Y}|}{b|\mathcal{X}|}}$ finishes the proof. \square

This also concludes the proof of Theorem 40. \square

6.4 Quantum Query Complexity and Formula Size

Let $F : \{0, 1\}^N \rightarrow \{0, 1\}$ be a Boolean function. It is known that

$$L(F) \geq \Omega\left(\frac{|R|^2}{|\mathcal{X}||\mathcal{Y}|}\right), \quad (5)$$

where $L(F)$ is the formula size of F , and X, Y and R are as previously defined. It is also easy to see that $Q_2(F) \leq L(F)$ (by induction on $L(F)$). Theorem 40 in the case when $b = b' = 1$, implies that

$$Q_2(F) \geq \Omega\left(\frac{|R|}{\sqrt{|\mathcal{X}||\mathcal{Y}|}}\right). \quad (6)$$

Inequalities (5) and (6) lead to the belief that $Q_2(F) \leq O(L(F)^{1/2})$. Grover's search algorithm corresponds to the case when F is a single OR function, therefore this conjecture can be viewed as a far far reaching generalization of his result.

In a recent breakthrough, Ambainis et. al. [8] almost proved this by showing that

$$Q_2(F) \leq L(F)^{\frac{1}{2}+o(1)}.$$

7 Quantum Communication Complexity

Suppose we have two parties Alice and Bob and a Boolean function $F : \{0, 1\}^N \times \{0, 1\}^N \rightarrow \{0, 1\}$. Consider a setting in which Alice has a Boolean string $X \in \{0, 1\}^N$ and Bob has a Boolean string $Y \in \{0, 1\}^N$, and their goal is to compute the value of $F(X, Y)$ by communicating as few bits as possible. Alice and Bob agree on a communication protocol beforehand. Having received inputs, they communicate in accordance with the protocol. At the end of the communication one of the parties declares the value of the function F . The cost of the protocol is the number of bits exchanged on the worst-case input.

Definition 43. The deterministic communication complexity of F , denoted by $C(F)$, is the cost of an optimal communication protocol computing F .

The topic of classical communication complexity was introduced and first studied by Andrew Yao in 1979 [9]. The following is a simple observation which is implied by the definition of deterministic communication complexity.

Observation 44. Let $F : \{0,1\}^N \times \{0,1\}^N \rightarrow \{0,1\}$ be a non-trivial Boolean function, meaning that it depends on all its variables. Then we have

$$C(F) \leq N.$$

Definition 45. For a Boolean function $F : \{0,1\}^N \times \{0,1\}^N \rightarrow \{0,1\}$, the communication matrix of F is a $\{0,1\}^N$ by $\{0,1\}^N$ matrix, denoted by M_F , where $M_F(X, Y) = F(X, Y)$.

We have the following rank lower bound for $C(F)$ due to Mehlhorn and Schmidt [10].

Theorem 46 (Mehlhorn and Schmidt [10]). *For any Boolean function $F : \{0,1\}^N \times \{0,1\}^N \rightarrow \{0,1\}$ we have*

$$C(F) \geq \log_2 \text{rk}(M_F).$$

Definition 47. Define $\text{EQ}_N : \{0,1\}^N \times \{0,1\}^N \rightarrow \{0,1\}$ to be the Boolean function where $\text{EQ}_N(X, Y) = 1$ if $X = Y$, and $\text{EQ}_N(X, Y) = 0$ otherwise.

We have the following lower bound on the communication complexity of the EQ_N function immediately following from Theorem 46.

Observation 48. *We have*

$$C(\text{EQ}_N) \geq N.$$

7.1 Probabilistic Communication Complexity

For a Boolean function $F : \{0,1\}^N \times \{0,1\}^N \rightarrow \{0,1\}$, let the two-way error probabilistic communication complexity of F be denoted by $C_2(F)$. Formally, it is defined similarly to $C(F)$, with the difference that the protocol is allowed to use random coins and is allowed to err with probability $\leq 1/3$ for each input (X, Y) . Then we have the following upper bound on the probabilistic communication complexity of the function EQ_N .

Theorem 49 (Rabin and Yao).

$$C_2(\text{EQ}_N) \leq O(\log N).$$

Proof. Let $p \geq 3N$ be a prime number which is only slightly greater than $3N$. Let E be an encoding of Boolean strings by low-degree polynomials over \mathbb{F}_p . The exact choice of this encoding does not matter, so we simply let $E(X) = \sum_{i=1}^N X_i \xi^i \in \mathbb{F}_p[\xi]$. Consider the following probabilistic protocol:

1. Alice chooses $z \in \mathbb{F}_p$ at random and sends $(z, E(X)(z))$ to Bob, where $E(X)(z) = \sum_{i=1}^N X_i z^i$.
2. Bob checks if $E(X)(z) = E(Y)(z)$, and outputs 1 if and only if it is the case.

If $X = Y$, then Bob always computes the correct value of $EQ(X, Y)$ at the last step. If $X \neq Y$, then $E(X)$ and $E(Y)$ differ for at least $2p/3$ out of p possibilities since their difference is a non-zero polynomial of degree $\leq N \leq p/3$ and thus can have at most $p/3$ roots in \mathbb{F}_p . So the probability that $E(X)$ and $E(Y)$ differ in the z -th coordinate and hence Bob correctly computes the value of $EQ(X, Y)$ is at least $\frac{2}{3}$, as desired.

It is easy to see that the number of bits transmitted in this protocol is $O(\log N)$. \square

7.2 Quantum Communication Complexity

Almost 15 years elapsed before the same pioneer, Andrew Yao, thought of asking how the situation of communication complexity might change in the quantum computation world [11].

Let \mathcal{X} and \mathcal{Y} be two sets, and $F : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ be a Boolean function. We will be working with $\mathcal{H}_A \otimes \mathcal{C} \otimes \mathcal{H}_B$, where \mathcal{H}_A , \mathcal{H}_B and \mathcal{C} are Hilbert spaces, which represent Alice's work space, Bob's work space, and communication channel respectively. A quantum communication protocol is defined as follows:

$$(I_A \otimes U_{Y,t})(U_{X,t} \otimes I_B) \cdots (I_A \otimes U_{Y,1})(U_{X,1} \otimes I_B)|\phi_0\rangle, \quad (7)$$

where $U_{X,i}$ are *arbitrary* unitary operators on $\mathcal{H}_A \otimes \mathcal{C}$ that depend only on Alice's input X , and $U_{Y,i}$ are described dually. The cost of the protocol is $t \cdot \log_2 \dim(\mathcal{C})$; the idea behind this measure is that we have t rounds of communication, with $\log_2 \dim(\mathcal{C})$ qubits sent in each of them. The quantum communication complexity of a function F (again, with error probability $1/3$) is equal to the cost of the most efficient quantum protocol to compute F and is denoted by $QC_2(F)$.

There have been different models of quantum communication complexity. While almost all of them are essentially equivalent, one important distinction that does not have any analogue in the black-box model is that of *prior entanglement*, depending on whether the initial vector $|\phi_0\rangle$ in (7) is arbitrarily entangled or simply has the form $|0^a\rangle \otimes |0^b\rangle \otimes |0^c\rangle$.

The straightforward analogue of Theorem 29 can not be true since Observation 48 and Theorem 49 already imply an exponential separation between

$C(F)$ and $C_2(F)$. Thus, a sensible thing to ask is if $C_2(F)$ and $QC_2(F)$ are polynomially related. For *partial* functions this is known to be not true [12], and for total functions this is a major open problem:

Conjecture 50. $C_2(F) \leq QC_2(F)^{O(1)}$ for totally defined functions F .

In the rest of this block we will discuss this conjecture for a natural class of total functions F where progress is being made, and that might be more tractable than the general case.

Definition 51. For two functions $F : \{0, 1\}^N \rightarrow \{0, 1\}$, and $g : \{0, 1\} \times \{0, 1\} \rightarrow \{0, 1\}$, the function $F \circ g^N$ is defined as follows

$$F \circ g^N := F(g(X_1, Y_1), g(X_2, Y_2), \dots, g(X_N, Y_N)).$$

The $F \circ g^N$ functions are called *block-composed functions* and are widely studied. Many authors also consider even more general case, when each block consists not of a single (qu)bit, but of a constant number of them. For the purpose of our discussion, however, the one-qubit case suffices.

Following are well-studied examples of block-composed functions:

1. $\text{EQ}(X, Y) = \bigwedge_z (X_z = Y_z)$.
2. $\text{IP}(X, Y) = \bigoplus_z (X_z \wedge Y_z)$.
3. $\text{DISJ}(X, Y) = \bigvee_z (X_z \wedge Y_z)$.

The following simple theorem relates quantum communication complexity of block-composed functions to the quantum complexity measure Q_2 .

Theorem 52 (Buhrman, Cleve and Wigderson 98 [13]). *We have*

$$QC_2(F \circ g^N) \leq O(Q_2(F) \log N).$$

Proof. In this case we will have a communication channel of dimension $O(N)$ (that is, representable by $O(\log N)$ qubits), and only Alice will have a work space. We will use an efficient black box protocol to compute F , and during the process, for every query

$$U_g : |x, s, 0\rangle \mapsto |x, s \oplus g(X_x, Y_x), 0\rangle,$$

Alice will compute U_g in her work space after providing X_x to Bob and getting back $g(X_x, Y_x)$ from Bob in the communication channel.

- $|x, s, 0\rangle|0, 0, 0\rangle \mapsto (\text{Alice}) |x, s, 0\rangle|X_x, x, 0\rangle;$
- $|x, s, 0\rangle|X_x, x, 0\rangle \mapsto (\text{Bob}) |x, s, 0\rangle|X_x, x, g(X_x, Y_x)\rangle;$
- $|x, s, 0\rangle|X_x, x, g(X_x, Y_x)\rangle \mapsto (\text{Alice}) |x, s, g(X_x, Y_x)\rangle|X_x, x, g(X_x, Y_x)\rangle;$
- $|x, s, g(X_x, Y_x)\rangle|X_x, x, g(X_x, Y_x)\rangle \mapsto (\text{Alice}) |x, s \oplus g(X_x, Y_x), g(X_x, Y_x)\rangle|X_x, x, g(X_x, Y_x)\rangle;$
- $|x, s \oplus g(X_x, Y_x), g(X_x, Y_x)\rangle|X_x, x, g(X_x, Y_x)\rangle \mapsto |x, s \oplus g(X_x, Y_x), 0\rangle|0, 0, 0\rangle,$

where the last step can be done by the Garbage Removal Lemma (Theorem 1). The cost of the protocol is $O(Q_2(F) \log N)$. \square

The following conjecture (that we purposely state in a little bit loose form) states that we cannot in fact do much better than that:

Conjecture 53. There is no better way to compute block-composed functions other than computing G 's in parallel and computing F of the outputs at the end.

Razborov confirmed the above conjecture for the case when F is a symmetric Boolean function [14]. Note that we also trivially have the classical analogue of Theorem 52: $C_2(F \circ g^N) \leq O(R_2(F))$, where $R_2(F)$ is the *randomized* decision-tree complexity of the predicate F . Since $R_2(F)$ and $Q_2(F)$ are polynomially related by Theorem 29, Conjecture 53 does imply Conjecture 50 for block-composed functions.

Lectures 14,15

Scribe: Pratik Worah, University of Chicago.

Date: 22, 23 February, 2011

Now we study a technique (discrepancy method) for lower bounds on QC_2 . We conclude with a sketch of some ideas involved in more advanced proofs based on generalizing the discrepancy method.

Note that we will work under the context described in Section 7 where the input is simply $|0^a, 0^b, 0^c\rangle$ and no prior entanglement is present¹. The output is written to the channel in the end. As in Definition 45, given a function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ denote by M_F ($M_F(X, Y) = F(X, Y)$) the communication matrix of F in the quantum model². More precisely,

¹Most of the material, however, can be generalized to the case with prior entanglement as well – see [14, Remark 4] for details.

²Let $|\mathcal{X}| = |\mathcal{Y}| = \mathcal{N}$, and so $N = \log_2 \mathcal{N}$.

Definition 54. The *quantum communication complexity with bounded-error probability* of a function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ denoted by $QC_2(F)$ is the cost of the most efficient quantum protocol (cf. Section 7) to compute F such that the probability of computing the correct answer is at least $\frac{2}{3}$.

We will denote by P_F the matrix of probabilities of acceptance for F (i.e., $P_F(X, Y)$ is the probability that the protocol accepts). Then, in the matrix notation, our acceptance condition can be written as³ $\ell_\infty(M_F - P_F) \leq \frac{1}{3}$.

7.3 Decomposition of quantum protocols

We now assume that the channel has only one qubit. This simplifies the expressions in the next theorem, and it is well-known that it does not restrict the power of the model much. The following observation abstracts out the structure of P_F .

Theorem 55. *Given $F : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ and P_F as before, we have*

$$P_F = \sum_{i=1}^{2^{2QC_2(F)}} R_i,$$

where R_i are real rank 1 matrices of the form $R_i = C_i D_i^T$ (we write D^T instead of D^\dagger for real matrices) such that $\ell_\infty(C), \ell_\infty(D) \leq 1$.

Proof. We start by proving a lemma regarding the structure of the quantum state.

Lemma 56. *The final state of a t step quantum communication protocol (7) is expressible as*

$$\sum_{c \in \{0,1\}} \sum_{i=1}^{2^t} A_{ci}(X) \otimes |c\rangle \otimes B_{ci}(Y), \quad (8)$$

where A_i, B_i are (complex) vectors with ℓ_2 norm ≤ 1 .

Proof. The proof will be by induction on the length of the protocol t . In the base case $t = 0$ the vectors A_{c1}, B_{c1} are unit vectors so the lemma is true.

Suppose the lemma holds for $t - 1$ steps and suppose that Alice applied $(U_{X,t} \otimes I_B)$. Since $U_{X,t}$ is unitary it preserves ℓ_2 norm, so taking projections implies $\exists A_{ci0}, A_{ci1}$ with ℓ_2 norm at most that of A_{ci} such that

$$(U_{X,t} \otimes I_B)(|A_{ci}\rangle|c\rangle)|B_{ci}\rangle = (|A_{ci0}\rangle|0\rangle + |A_{ci1}\rangle|1\rangle)|B_{ci}\rangle. \quad (9)$$

³ $\ell_\infty(A) = \max_{i,j} |A(i, j)|$.

Therefore the number of terms in our sum increases by at most a factor of 2. Observe that this suffices to prove the lemma. \square

Given Lemma 56, we can calculate P_F as follows (assuming $|1\rangle$ is accepting).

$$P_F(X, Y) = \sum_{i,j=1}^{2^t} \langle A_{1i}(X), A_{1j}^*(X) \rangle \langle B_{1i}(Y), B_{1j}^*(Y) \rangle.$$

Therefore P_F has the form $\sum_{i,j=1}^{2^t} C_{i,j} D_{i,j}^T$. Since $\|A_{1i}\|_2, \|B_{1i}\|_2 \leq 1$, C, D (viewed simply as vectors of length 2^{2t}) will have ℓ_∞ norm ≤ 1 . Moreover, $\text{rk}(CD) \leq \min(\text{rk}(C), \text{rk}(D))$ implies $\text{rk}(R_i) \leq 1$. Hence the proof follows. \square

As a corollary, we obtain the log rank bound for QC (zero-error version of QC_2).

Corollary 57. $QC(F) \geq \Omega(\log \text{rk}(M_F))$.

In the following, we discuss methods for obtaining lower bounds on $QC_2(F)$.

7.4 Lower bound for $QC_2(\text{IP}_2)$

We now change M_F to a ± 1 valued matrix while keeping $\ell_\infty(M_F - P_F)$ bounded by at most a small enough constant say ϵ . This will also require replacing P_F with $J_N - 2P_F$, where J_N is an all-one matrix. But this linear transformation does not affect the validity of Theorem 55, up to a small multiplicative increase in the number of terms, and this is the only property of P_F we are going to use (in particular, we will not need that it is non-negative).

Definition 58. Define the *Frobenius product* of two matrices A and B by $\langle A, B \rangle := \sum_{i,j} A_{ij}^* B_{ij}$.

Let us write M_F as $P_F + \Delta$ where $\ell_\infty(\Delta) \leq \epsilon$. Observe that

$$\mathcal{N}^2 = \langle M_F, M_F \rangle = \langle M_F, \Delta \rangle + \langle M_F, P_F \rangle. \quad (10)$$

Since the first term in the RHS is at most $\epsilon \mathcal{N}^2$, the second term has to be $\Omega(\mathcal{N}^2)$. Theorem 55 gives (for $QC_2(F) = k$)

$$\langle M_F, P_F \rangle = \sum_{i=1}^{2^{2k}} \langle M_F, R_i \rangle = \sum_{i=1}^{2^{2k}} \langle M_F, C_i D_i^T \rangle. \quad (11)$$

Note that for a single term in the last sum

$$\langle M_F, CD^T \rangle = \sum_{j,k=1}^{\mathcal{N}} M_F(j,k)(CD^T)(j,k) = \sum_{j,k=1}^{\mathcal{N}} C_j M_F(j,k) D_k = C^T M_F D.$$

Since $\ell_\infty(C), \ell_\infty(D) \leq 1$, we have $\|C\|, \|D\| \leq \sqrt{\mathcal{N}}$ (recall that $\|\cdot\|$ stands for the ℓ_2 -norm). Therefore by definition of spectral norm (that we will also denote simply by $\|\cdot\|$), $\|C^T M_F D\| \leq \mathcal{N} \|M_F\|$.

Hence returning to the original equation (11),

$$\langle M_F, P_F \rangle \leq \sum_{i=1}^{2^{2k}} \|C_i^T M D_i\| \leq 2^{2k} \mathcal{N} \|M_F\|.$$

Since, as we observed above, $\langle M_F, P_F \rangle \geq \Omega(\mathcal{N}^2)$, we conclude

$$2^{2k} \geq \Omega\left(\frac{\mathcal{N}}{\|M_F\|}\right). \quad (12)$$

The lower bound method above is known as the *discrepancy* method. In summary we showed that for a relation like (10) to hold with small $\ell_\infty(\Delta)$, $\langle M_F, P_F \rangle$ must be large. Using this fact and the properties of our quantum model we obtained the desired lower bound.

As an example, consider the function $F = \text{IP}_2$ i.e., $F(X, Y) = \bigoplus_{z=1}^N (X_z \wedge Y_z)$.

Observation 59. For $F = \text{IP}_2$, $\|M_F\| = \sqrt{\mathcal{N}}$.

Proof. The inner product matrix M_F is an orthogonal matrix, up to a normalizing factor (specifically a Hadamard matrix). Therefore all its eigenvalues are $\sqrt{\mathcal{N}}$ in absolute value. \square

The above discussion therefore implies:

Theorem 60 ([15]). $QC_2(\text{IP}_2) = \Omega(N)$.

As an aside, the following is an open problem in this area (see [16, Section 8] for more details).

Conjecture 61. If $F \in AC^0$ then $\|M\|$ is large for any large submatrix M of M_F .

A consequence of this would be that the “naive” discrepancy bound (12) *provably* does not work for functions in AC^0 . In the next two subsections we discuss its generalizations that can do the job.

7.5 Lower bound for $QC_2(\text{DISJ})$

In this subsection the aim is to study lower bounds on $QC_2(F \circ \wedge^N)$ for block-composed functions (cf. later half of Section 7) with symmetric F .

Tight estimates of the approximate degree (see Definition 36) of symmetric Boolean functions were obtained by Paturi [17] in terms of Γ , a quantity which depends on whether F changes values near $\frac{N}{2}$ or far from $\frac{N}{2}$. For brevity, we identify F with its univariate representation $[0, 1 \dots, N] \rightarrow \{0, 1\}$ (cf. Lemma 35).

Definition 62. $\Gamma_0(F)$ and $\Gamma_1(F)$ are defined as follows:

$$\Gamma_0(F) := \max \left\{ k \mid 1 \leq k \leq \frac{N}{2}, F(k) \neq F(k-1) \right\}$$

;

$$\Gamma_1(F) := \max \left\{ n - k \mid \frac{N}{2} \leq k \leq N, F(k) \neq F(k+1) \right\}.$$

Razborov [14] proved the following bound for symmetric F .

Theorem 63. $QC_2(F \circ \wedge^N) = \tilde{\Theta}(\sqrt{N\Gamma_0(F)} + \Gamma_1(F))$.

In case of F being the disjointness predicate (i.e., $F = \vee$) we have $\Gamma_0 = 1$ and $\Gamma_1 = 0$ so $QC_2(\text{DISJ}) = \tilde{\Omega}(\sqrt{N})$ [14]. The following discussion briefly gives the ideas involved in this proof.

Definition 64. The *trace norm* of a real symmetric matrix A is defined as $\|A\|_{\text{tr}} = \sum_{i=1}^n |\lambda_i(A)|$.

This can also be defined for general matrices; one would replace eigenvalues by singular values. But the alternative characterization of the above definition given below covers this case as well.

Observation 65. For an arbitrary matrix A

$$\|A\|_{\text{tr}} = \max_B \{ \langle A, B \rangle \mid \|B\| = 1 \}.$$

Proof. (for symmetric matrices) Since trace of a matrix (denoted Tr) and spectral norm are invariant under conjugate transforms and since $\langle A, B \rangle = \text{Tr}(A^\dagger B)$ we can diagonalize A (which is symmetric) to obtain

$$\langle A, B \rangle = \text{Tr} \begin{pmatrix} \lambda_1(A)B_{11} & \cdots & \lambda_1(A)B_{N1} \\ \vdots & \cdots & \vdots \\ \lambda_N(A)B_{1N} & \cdots & \lambda_N(A)B_{NN} \end{pmatrix}.$$

Now $\|B\| \leq 1$ implies $|B_{ii}| \leq 1$ in any orthogonal basis, in particular in the one chosen above that diagonalizes A . Therefore by Definition 64, $\text{LHS} \geq \text{RHS}$ in the statement above. Note that $\text{LHS} = \text{RHS}$ when B is a diagonal matrix (in our basis) with non-zero entries appropriately chosen from the set $\{\pm 1\}$. Hence the proof follows. \square

As a by-side remark (included mostly for educational purposes), this is a partial case of the following general paradigm.

Definition 66. The *dual norm* denoted $\|\cdot\|_*$ of a norm $\|\cdot\|$ is defined as

$$\|A\|_* = \sup\{\langle A, B \rangle \mid \|B\| \leq 1\}.$$

The observation above implies that the spectral norm and trace norm are dual norms. In general, $\text{Tr}(A^*B) \leq \|A\|_* \|B\|$ so in particular we have

$$\langle M_F, P_F \rangle \leq \|P_F\|_{\text{tr}} \|M_F\|.$$

Hence if we upper bound the trace norm of P and spectral norm of F then we can derive a contradiction to the decomposition in (10).

Unfortunately, this simple approach does not suffice for the disjointness function as its spectral norm

$$\|M_F\|$$

is large. Instead, [14] introduced the *approximate trace norm*

$$\|A\|_{\tilde{\text{tr}}} := \min\{\|B\|_{\text{tr}} \mid \ell_\infty(A - B) \leq \epsilon\}.$$

The following is not hard to prove.

Theorem 67 ([14]). *We have*

$$QC_2(F) = \Omega\left(\log \frac{\|M_F\|_{\tilde{\text{tr}}}}{\mathcal{N}}\right).$$

All that remains is to develop methods for bounding $\|M_F\|_{\tilde{\text{tr}}}$ from below, and this innocent-looking task turned out to be rather difficult.

7.6 Generalizations of the discrepancy method

These ideas were gradually developed and used in [11, 15, 18]; the exposition below follows [14, Section 5.2].

Let μ be a $\mathcal{N} \times \mathcal{N}$ matrix so that

$$\langle M_F, \mu \rangle = \langle P, \mu \rangle + \langle \Delta, \mu \rangle$$

and such that $\langle \Delta, \mu \rangle \leq \ell_\infty(\Delta) \ell_1(\mu) \leq c\mathcal{N}^2$. Earlier we had $\mu = M_F$ in the normal discrepancy method (so that $\ell_1(M_F) = \mathcal{N}^2$), but now we are free to choose μ subject to the following constraints:

1. $\ell_1(\mu) \leq 1$.
2. $\langle M_F, \mu \rangle \geq \frac{2}{3}\mathcal{N}^2$.
3. $\|\mu\|$ is as small as possible.

[14] erroneously claimed that no such μ can exist when $F = \text{DISJ}$ and developed instead another method of “multi-dimensional” discrepancy. We can not go into much details here, but the general idea is to test M_F not against a single matrix μ , but against a whole (finite) family of such matrices.

Using his *pattern matrix method*, Sherstov [19] showed that in fact a single μ with the desired properties exist, that resulted in another proof of Theorem 63. While simpler than the original one, it is still too complicated to be included here.

Note that even more proof methods using different norms with desirable properties are known. Linial and Shraibman [20] use the norm γ_2 defined as follows.

Definition 68. Given matrix A , let

$$\gamma_2(A) = \min_{XY=A} \left(\max_{\|x\|_2=1} \ell_\infty(Xx) \cdot \max_{\|y\|_1=1} \|Yy\|_2 \right).$$

It can be shown that $\gamma_2(A) \geq \|A\|_{\text{tr}}$ (therefore it maybe easier to lower bound than the trace norm) but γ_2 is not invariant under conjugation. [20] uses γ_2 norm (and its many variants) to obtain quantum communication complexity lower bounds (including weaker lower bounds for DISJ).

7.7 Direct products

Sometimes it is possible to save resources by solving many instances of a problem together as opposed to solving each instance naively. Multiplying

two $n \times n$ matrices provides an example - a matrix-vector multiplication takes n^2 operations, and thus one might expect that multiplying a matrix by n independent vectors should take n^3 operations. However, using fast matrix multiplication algorithms [21] one can solve the same problem in $o(n^3)$ operations. Of course most of the times one is not so lucky and then it is a challenge to prove that naive solving of the separate instances is the best that can be done (cf. Conjecture 53). Such theorems are known as *direct product theorems*. But the following is still open.

Question 69. Given $F_i : \mathcal{X}_i \times \mathcal{Y}_i \rightarrow \{0,1\}$ for $i = 1, \dots, t$, define in a natural manner the function $(F_1 \times \dots \times F_t)(\mathcal{X}_1 \times \dots \times \mathcal{Y}_t) \rightarrow \{0,1\}^t$. Is $QC_2(F_1 \times \dots \times F_t) \simeq \sum_{i=1}^t QC_2(F_i)$?

Note that a similar direct product result is known for the γ_2 norm [22].

Lecture16

Scribe: Youlian Simidjijski, University of Chicago.

Date: February 24, 2011

8 Quantum Error-Correcting Codes

Goal: Given a single qubit, we wish to preserve it in the presence of noise. We introduce noise into our system by applying various superoperators to the density matrix corresponding to our input. Three such superoperators corresponding to particular noisy channels are given below (here $\eta > 0$ is a real parameter called *noise rate*).

Depolarizing channel $\mathcal{E}_\eta(\rho) = (1 - \eta)\rho + \frac{\eta}{2}I_2$.

Bit flip channel $\mathcal{E}_\eta(\rho) = (1 - \eta)\rho + \eta \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \rho \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

Phase shift channel $\mathcal{E}_\eta(\rho) = (1 - \eta)\rho + \eta \begin{pmatrix} 1 & 0 \\ 0 & \alpha \end{pmatrix} \rho \begin{pmatrix} 1 & 0 \\ 0 & \alpha^* \end{pmatrix}$ for some $\alpha \in S^1$. We will be using the case where $\alpha = -1$, so that this becomes the phase flip channel.

8.1 Operator-sum representation of superoperators

Every physically realizable superoperator has the normal form:

$$T(\rho) = \sum_k E_k \rho E_k^\dagger, \text{ satisfying } \sum_k E_k^\dagger E_k = I$$

(the three noisy channels considered above make a very good example). Operators $\{E_k\}$ are called *operation elements*.

The condition that $\sum_k E_k^\dagger E_k = I$ stems from our earlier requirements on superoperators. Namely from the requirement that $\forall \rho, \text{tr}(\rho) = \text{tr}(T(\rho))$. This requirement implies

$$\text{tr}(T(\rho)) = \text{tr} \left(\sum_k E_k \rho E_k^\dagger \right) = \sum_k \text{tr}(E_k \rho E_k^\dagger) = \sum_k \text{tr}(E_k^\dagger E_k \rho) = \text{tr} \left(\sum_k (E_k^\dagger E_k) \rho \right),$$

and $\text{tr}(\rho) = \text{tr}(\sum_k (E_k^\dagger E_k) \rho)$ holds for all ρ if and only if $\sum_k E_k^\dagger E_k = I$.

8.2 Projective Measurements

Given a Hilbert space $\mathcal{H} = \mathcal{C}_1 \oplus \mathcal{C}_2 \oplus \dots \oplus \mathcal{C}_\ell$, consider projections P_1, P_2, \dots, P_ℓ onto \mathcal{C}_i . Projections satisfy the properties

1. $P_i^2 = P_i = P_i P_i^\dagger$;
2. $\sum_k P_k^\dagger P_k = \sum_k P_k = I$.

Thus, we have a superoperator mapping $\rho \mapsto \sum_k P_k^\dagger \rho P_k$, which is called *projective measurement*.

8.3 Quantum Information Theory

Definition 70. Given a state ρ and noisy channel \mathcal{E}_η , a *Recovery Operator*, \mathcal{R} is a superoperator satisfying $\mathcal{R}(\mathcal{E}_\eta(\rho)) = \rho$ for all ρ .

Given a general noisy channel (that is, an arbitrary superoperator), it is not possible to find such an \mathcal{R} . One good reason is that the trace distance inequality (23) on the noise operator yields $D(\mathcal{E}_\eta(\rho), \mathcal{E}_\eta(\rho')) \leq D(\rho, \rho')$, and if the inequality is strict, then inversion is not possible by a superoperator. However, we can hope to recover against specific classes of errors.

8.3.1 Error Correcting Codes in Classical Information Theory

Repetition codes: Suppose that a bit x gets flipped during transmission with probability $p < .5$, independently of all others. If we first map $x \mapsto (x, x, x)$ before transmission, and we reconstruct by majority vote, then the probability of failed reconstruction is $3p^2 - 2p^3 (\ll p)$. This cannot be trivially implemented in quantum computation due to the no cloning theorem (There

is no superoperator T which sends $|\phi\rangle\langle\phi|$ to $|\phi\rangle\langle\phi|\otimes|\phi\rangle\langle\phi|$. One of the many possible proofs is based on Theorem 23).

Most codes in classical information theory are linear codes. The setup for such schemes is as follows. Let $n > m$, and consider $\mathcal{C} \subset \{0,1\}^n$. We encode $\{0,1\}^m$ in \mathcal{C} by an injective mapping, typically including introducing some redundancy for decoding. We use a similar scheme for the quantum case.

Let $\mathcal{C} \subset \mathcal{H}$ be defined over a field of characteristic 0, and let $\dim(\mathcal{C}) = 2$, since we are only interested in single qubits. We will encode ρ in \mathcal{C} .

8.3.2 Correcting Against Quantum Bit Flip

Suppose that we are given a pure state on one qubit, $\phi = a|0\rangle + b|1\rangle$. We first map ϕ to a pure state on three qubits by sending $\phi \mapsto a|000\rangle + b|111\rangle$. Note that since these states are entangled, we have not violated the no cloning theorem. We now send each qubit across the channel \mathcal{E} . This yields a mixed state, ρ , because of the possibility of bit flips.

Consider now the projection operators P_0, P_1, P_2 , and P_3 , which project onto the subspaces in \mathcal{H} that are generated by $(|000\rangle, |111\rangle)$, $(|100\rangle, |011\rangle)$, $(|010\rangle, |101\rangle)$, and $(|001\rangle, |110\rangle)$ respectively. Then the map $\rho \mapsto \sum_{k=0}^3 (U_k P_k \rho P_k^\dagger U_k^\dagger)$ (where U_k is the bit flip operator corresponding to the error detected by a given P_k) defines a superoperator which decodes ρ with probability $3p^2 - 2p^3$ as in the classical case.

Physically, we have applied the projective measurement (see Section 8.2) followed by taking so-called *syndrome* represented in our case by the operator U_k (Note that the whole point of this procedure is that the operators U_k are *different* for different subspaces). For these reasons, this decoding procedure is sometimes called *syndrome measurement*. We will see in the proof of Theorem 72 that it is actually universal.

8.3.3 Correcting Against Quantum Phase Flip

Once we can correct against quantum bit flip, correcting against quantum phase flip is easy by first transforming into the Hadamard basis, given by $(\frac{1}{\sqrt{2}})(|0\rangle + |1\rangle)$ and $(\frac{1}{\sqrt{2}})(|0\rangle - |1\rangle)$. Phase shift in the standard basis is bit flip in the Hadamard basis, so there is no work to be done.

8.3.4 Correcting Against Simultaneous Bit and Phase Flip

With the bit flip and phase flip correcting techniques in hand, composing the two codes immediately yields a 9-qubit code that prevents against a channel that could perform both bit flips and phase flips. In addition to correcting bit and phase flip errors, Shor's code actually corrects against arbitrary errors on a single qubit. For more information about this particular code, consult [1, Section 10.2].

Now we develop a *general* mathematical technique (called *discretization of errors*) that allows us to reduce error-correcting of a huge (typically continuous) class of errors to correcting a finite set of errors, in many cases possessing a nice structure.

Lecture 17

Scribes: Youlian Simidjiski and David Kim, University of Chicago.

Date: March 1, 2011 and May 23, 2013

8.4 Conditions for the Recovery Operator

Recall that given a quantum code $\mathcal{C} \subset \mathcal{H}$, we say a noise operator \mathcal{E} is recoverable if there is a recovery operator \mathcal{R} such that for all density matrices ρ generated within \mathcal{C} , $\mathcal{R}(\mathcal{E}(\rho)) = \rho$.

Question 71. *What are necessary and sufficient conditions for the existence of a recovery operator \mathcal{R} ?*

We will ultimately show that given a channel $\mathcal{E}(\rho) = \sum_k E_k \rho E_k^\dagger$ and an encoding scheme mapping our input space onto $\mathcal{C} \subset \mathcal{H}$ as above, then the following theorem holds:

Theorem 72 (Quantum Error Correction Conditions). *Let \mathcal{E} be a noise operator with operation elements $\{E_k\}$, and let \mathcal{C} be a quantum code. If P is the projection operator onto \mathcal{C} , then a recovery operator \mathcal{R} exists iff $PE_i^\dagger E_j P = \alpha_{ij} P$ for some (Hermitian) matrix with entries α_{ij} .*

We will prove two more theorems first. Assume we have a density matrix generated from an ensemble of pure states $\{p_i, |\phi_i\rangle\}$, $\sum_i p_i = 1$, such that

$$\rho_\phi = \sum_i p_i |\phi_i\rangle \langle \phi_i|.$$

Assume we have another ensemble of pure states $\{q_j, |\psi_j\rangle\}$, $\sum_j q_j = 1$, such that it generates

$$\rho_\psi = \sum_j q_j |\psi_j\rangle \langle \psi_j|.$$

With slight abuse, we introduce a new notation: let $|\widetilde{\phi}_i\rangle = \sqrt{p_i}|\phi_i\rangle$ for $i = 1, \dots, n$, and $|\widetilde{\psi}_j\rangle = \sqrt{q_j}|\psi_j\rangle$ for $j = 1, \dots, m$. Then it is easy to see that

$$\rho_\phi = \sum_i p_i |\phi_i\rangle \langle \phi_i| = \sum_i |\widetilde{\phi}_i\rangle \langle \widetilde{\phi}_i|;$$

$$\rho_\psi = \sum_j q_j |\psi_j\rangle \langle \psi_j| = \sum_j |\widetilde{\psi}_j\rangle \langle \widetilde{\psi}_j|.$$

If we assume that the two density matrices are the same ($\rho_\phi = \rho_\psi$), what can we say about these states? Assume that $n = m$, as we can simply pad any one of the sets with pure states having 0 probabilities.

Theorem 73. *Given two ensembles of pure states $\{p_i, |\phi_i\rangle\}$ and $\{q_j, |\psi_j\rangle\}$, they generate the same density matrix $\rho_\phi = \rho_\psi$ iff there exists a unitary matrix U such that $|\widetilde{\psi}_j\rangle = \sum_i u_{ij}|\widetilde{\phi}_i\rangle$.*

Proof. We form two matrices $A = [|\widetilde{\phi}_1\rangle \dots |\widetilde{\phi}_n\rangle]$, and $B = [|\widetilde{\psi}_1\rangle \dots |\widetilde{\psi}_n\rangle]$, where the (normalized) pure states form the columns. Then $\rho_\phi = \sum_i |\phi_i\rangle \langle \phi_i| = AA^\dagger$, and $\rho_\psi = BB^\dagger$. We want to show that they are equal iff there exists a unitary U such that $B = AU$.

Assume such a U exists. Then $BB^\dagger = (AU)(U^\dagger A^\dagger) = AA^\dagger$, as U is unitary.

Conversely, assume that $AA^\dagger = BB^\dagger$. Then both are Hermitian, non-negative, and diagonalizable. Without loss of generality, we can assume that

$$AA^\dagger = BB^\dagger = \begin{pmatrix} d_1 & & \\ & \dots & \\ & & d_n \end{pmatrix}, d_i \geq 0. \text{ Thinking in terms of geometry, for}$$

non-zero d_i 's the rows of A must be orthogonal, each with length $\sqrt{d_i}$. The same argument applies to B as well. So both sets of row vectors can be normalized to form orthonormal bases, and we can have a unitary U such that $B = AU$. \square

For the next theorem, we make the same assumption $m = n$.

Theorem 74 (Unitary Invariants). *Two superoperators $\rho \mapsto \sum_{k=1}^n E_k \rho E_k^\dagger$ and $\rho \mapsto \sum_{\ell=1}^n F_\ell \rho F_\ell^\dagger$ in operator-sum form are equal iff there exists a unitary matrix U such that $F_\ell = \sum_k u_{k\ell} E_k$.*

Proof. Given U as above, then

$$\begin{aligned}
\sum_{\ell} F_{\ell} \rho F_{\ell}^{\dagger} &= \sum_{\ell, k, k'} u_{k\ell} E_k \rho E_{k'}^{\dagger} u_{k'\ell}^* \\
&= \sum_{k, k'} E_k \rho E_{k'}^{\dagger} \sum_{\ell} u_{k\ell} u_{k'\ell}^* \\
&= \sum_{k, k'} E_k \rho E_{k'}^{\dagger} \delta_{kk'} \\
&= \sum_k E_k \rho E_k^{\dagger},
\end{aligned}$$

where $u_{k\ell}^*$ is the term k, ℓ in U^{\dagger} , and $\sum_{k, k'} u_{k, \ell} u_{k', \ell}^* = \delta_{kk'}$ because U is unitary.

Conversely, assume we have the equality for all ρ , that is

$$\mathcal{E}(\rho) = \sum_{\ell} F_{\ell} \rho F_{\ell}^{\dagger} = \sum_k E_k \rho E_k^{\dagger}.$$

Let \mathcal{H} be the Hilbert space on which these operators act, and assume we have a basis $\{|x\rangle\}$. Let $\dim(\mathcal{H}) = N$ and $p_k = \frac{1}{N} \text{Tr}(E_k^{\dagger} E_k)$ (so that $\sum_k p_k = 1$). Define maximally entangled states $|\phi_k\rangle$ in $\mathcal{H} \otimes \mathcal{H}$ by

$$|\phi_k\rangle = \frac{1}{\sqrt{p_k}} |\widetilde{\phi_k}\rangle,$$

where

$$|\widetilde{\phi_k}\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle E_k |x\rangle.$$

Take the probability distribution $(p_k, |\phi_k\rangle)$ on these states. Then this ensemble will generate the density matrix

$$\begin{aligned}
\sum_k |\widetilde{\phi_k}\rangle \langle \widetilde{\phi_k}| &= \frac{1}{N} \sum_k \left(\sum_x |x\rangle E_k |x\rangle \right) \left(\sum_y \langle y| \langle y| E_k^{\dagger} \right) \\
&= \frac{1}{N} \sum_{x, y} \left(|x\rangle \langle y| \otimes \sum_k E_k |x\rangle \langle y| E_k^{\dagger} \right) \\
&= \frac{1}{N} \sum_{x, y} |x\rangle \langle y| \otimes \mathcal{E}(|x\rangle \langle y|).
\end{aligned}$$

We can interpret this as taking the operator to every position in the n by n matrix.

Let also $|\widetilde{\psi}_\ell\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle F_\ell |x\rangle$. Then a similar calculation applied to F_ℓ gives the same density matrix, so $\exists U$ such that $|\widetilde{\psi}_\ell\rangle = \sum_k u_{k\ell} |\widetilde{\phi}_k\rangle$ by the previous theorem. So we have $\exists U$ such that

$$\sum_x |x\rangle F_\ell |x\rangle = \sum_x |x\rangle \sum_k u_{k\ell} E_k |x\rangle.$$

They will have to cancel out on each of the basis vectors, and thus, we have $F_\ell = \sum_k u_{k\ell} E_k$ with the same unitary U . \square

Let A be any square matrix. Consider $A^\dagger A$. This is clearly Hermitian, positive-semidefinite, and so we can diagonalize $A^\dagger A$ and define $\sqrt{A^\dagger A}$ by

$$\sqrt{A^\dagger A} = \begin{pmatrix} \sqrt{\lambda_1} & 0 & \cdots & 0 \\ 0 & \sqrt{\lambda_2} & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & \cdots & 0 & \sqrt{\lambda_N} \end{pmatrix}.$$

Lemma 75 (Polar Decomposition). *Given a matrix A , there exists a unitary U such that $A = U\sqrt{A^\dagger A}$. Thus, any matrix can be decomposed as the product of a hermitian matrix and a unitary matrix.*

Proof. A proof of the Polar Decomposition Lemma can be found in a standard linear algebra text, or in [1, Theorem 2.3]. \square

Proof of Theorem 72. First we show that the conditions are necessary. Suppose we have a noise operator \mathcal{E} with operational elements $\{E_i\}$ and a recovery operator \mathcal{R} which has operation elements $\{F_k\}$. If P is the projector onto \mathcal{C} , the quantum encoding space, then we have

$$\sum_{i,k} F_k E_i P \rho P E_i^\dagger F_k^\dagger = P \rho P. \quad (13)$$

Let $Q = I - P$ be the projection onto the subspace \mathcal{C}^\perp orthogonal to \mathcal{C} . Due to (13), the two sets of operational elements $\{F_k E_i P, Q\}$ and $\{P, Q\}$ (with necessary paddings) define the same superoperator, namely the projective measurement corresponding to the decomposition $\mathcal{H} = \mathcal{C} \oplus \mathcal{C}^\perp$. If we apply Theorem 74, we have $F_k E_i P = cP + dQ$. We can simply multiply both sides by Q on the right to see that $d = 0$. This gives us $F_k E_i P \sim P$. For every other j , we also have $F_k E_j P \sim P$. If we now sum $P E_i^\dagger F_k^\dagger F_k E_j P \sim P$ over all possible values of k , then we have $P E_i^\dagger E_j P \sim P$ as desired, as $\{F_k^\dagger F_k\}$ sum to I . It is obvious that the constant factors are Hermitian.

We now show the sufficiency of the conditions. First apply Theorem 74 in order to see that we can choose an equivalent set of operation elements so that the matrix α is diagonal with non-negative entries. We will assume that α is diagonal for the remainder of the proof.

Now observe that by the polar decomposition,

$$E_k P = U_k \sqrt{P E_k^\dagger E_k P} = \sqrt{\alpha_{k,k}} U_k P,$$

where we have applied the fact that $P^2 = P$.

Now consider $P_k = U_k P U_k^\dagger$, the projection onto $U_k \mathcal{C}$, the image subspace of $E_k P$. Using the fact that the matrix α is diagonal, we can conclude that $U_\ell \mathcal{C}$ and $U_k \mathcal{C}$ are pairwise orthogonal, as

$$P_k P_\ell = U_k P U_k^\dagger U_\ell P U_\ell^\dagger \sim U_k (P E_k^\dagger E_\ell P) U_\ell^\dagger = 0.$$

So, the image spaces of P_k and P_ℓ are orthogonal because P_k and P_ℓ are projectors.

Thus, we can decompose \mathcal{E} into $\sum_k P_k + Q$, where the Q portion projects onto parts of \mathcal{H} that \mathcal{C} does not map into under the error map \mathcal{E} .

We can now recover our original ρ by the syndrome measurement using operators P_k and U_k^\dagger . Mathematically:

$$\begin{aligned} \mathcal{R}(\mathcal{E}(\rho)) &= \sum_{j,k} U_k^\dagger P_k E_j P \rho P E_j^\dagger P_k U_k \\ &= \sum_{j,k} U_k^\dagger (U_k P U_k^\dagger) E_j P \rho P E_j^\dagger (U_k P U_k^\dagger) U_k \\ &= \sum_{j,k} P U_k^\dagger E_j P \rho P E_j^\dagger U_k P \\ &= \frac{\sum_{j,k} (P E_k^\dagger E_j P) \rho (P E_j^\dagger E_k P)}{\alpha_{k,k}} \\ &= \frac{\sum_k (\alpha_{k,k} P) \rho (\alpha_{k,k} P)}{\alpha_{k,k}} \\ &= \sum_k \alpha_{k,k} P \rho P. \end{aligned}$$

and since $\rho \in \mathcal{C}$, and P is the projector onto \mathcal{C} , we reach our conclusion that $\mathcal{R}(\mathcal{E}(\rho)) \sim \rho$. This completes the proof that our conditions are sufficient for \mathcal{R} to exist. \square

Lecture 18

Scribe: Olga Medrano Martín del Campo, University of Chicago.

Date: March 25th, 2021

8.5 Stabilizer Codes

8.5.1 Definition and some Examples

We consider first the Pauli Group G_1 , which is an abstract group generated by the 2×2 Pauli matrices

$$G_1 = \langle X, Y, Z \rangle = \{I, X, Y, Z\} \cdot \{\pm 1\} \cdot \{\pm i\}. \quad (14)$$

Since X, Y, Z are matrices, it is fruitful to also think of G_1 as a group of linear operators acting in a single qubit Hilbert space. We can generalize this notion to Hilbert spaces with n qubits, in which the following linear operators

$$X_i, Y_i, Z_i, i = 1, 2, \dots, n$$

are tensor products of Pauli matrices and identity matrices, and act only on the i th qubit among n of them. That way, we can define the Pauli group G_n based on its generators:

$$G_n = \langle X_i, Y_i, Z_i, i = 1, 2, \dots, n \rangle$$

Remark. The groups G_n and G_1^n are *not* isomorphic as abstract groups! In particular, we have that $X_1 Y_1 Z_1$ acts by multiplication by $-i$, so it is a constant matrix which is not equal to the identity matrix. Thus in G_n we e.g. have the relation $X_1 Y_1 Z_1 = X_2 Y_2 Z_2$ that obviously does not hold in G_1^n .

Definition 76. Given a subgroup $S \leq G_n$ of the n -th Pauli Group, we obtain a corresponding stabilizer code

$$V_S = \text{Stab}(S) := \{v \in \mathcal{H} : \forall g \in S, gv = v\}.$$

Example. In a 2 qubit Hilbert space, we compute $\text{Stab}(Z_1 Z_2)$, the stabilizer code for $Z_1 Z_2$. Since for all $|ab\rangle$ ($a, b \in \{0, 1\}$) we have

$$Z_1 Z_2(|ab\rangle) = (-1)^{a+b} \cdot |ab\rangle,$$

then, in terms of the basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ of \mathcal{H}_2 , the signs of the coefficients for $|00\rangle, |11\rangle$ would remain unchanged, whereas the signs of $|01\rangle, |10\rangle$ would be flipped. With this we can tell that

$$\text{Stab}(Z_1 Z_2) = \text{Span}(|00\rangle, |11\rangle).$$

Example. In the same Hilbert space, we compute $Stab(X_1X_2)$. The following describes the action of X_1X_2 in \mathcal{H}_2 with respect to the basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$:

$$\begin{aligned} X_1X_2(\alpha) &= X_1X_2(\alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle) = \\ &\alpha_{11}|00\rangle + \alpha_{10}|01\rangle + \alpha_{01}|10\rangle + \alpha_{00}|11\rangle; \end{aligned}$$

then, a necessary and sufficient condition for α to be in the stabilizer code is the matching of the pairs of coefficients $\alpha_{00} = \alpha_{11}$ and $\alpha_{10} = \alpha_{01}$. Thus,

$$Stab(X_1X_2) = Span(|00\rangle + |11\rangle, |01\rangle + |10\rangle).$$

Example. In a 3 qubit Hilbert space, we compute $Stab(Z_1Z_2, Z_2Z_3, Z_1Z_3)$. The action of Z_1Z_2 with respect to the basis $\{|abc\rangle, a, b, c \in \{0, 1\}\}$ is given by:

$$Z_1Z_2 \left(\sum_{a,b,c \in \{0,1\}} \alpha_{abc} |abc\rangle \right) = \sum_{a,b,c \in \{0,1\}} (-1)^{a+b} \alpha_{abc} |abc\rangle$$

then, for α to be in the stabilizer code we must have $\alpha_{abc} = -\alpha_{abc} = 0$ whenever $a+b$ is odd. With a similar observation in the other two operators Z_2Z_3 and Z_1Z_3 , each of the basis elements which are not $|000\rangle$ or $|111\rangle$ must have zero coefficient. Conversely, these two basis elements are in the stabilizer code, hence

$$Stab(Z_1Z_2, Z_1Z_3, Z_2Z_3) = Span(|000\rangle, |111\rangle);$$

note the above is the bit flip code, which was mentioned in previous lectures.

Exercise 77. Show that in a 3 qubit Hilbert space, $Stab(X_1X_2, X_2X_3, X_1X_3)$ is the phase flip code.

Example. In a 2 qubit Hilbert space, we compute $Stab(\langle X_1X_2, Z_1Z_2 \rangle)$. This stabilizer code has to be contained in both

$$Stab(X_1X_2) = Span(|00\rangle + |11\rangle, |01\rangle + |10\rangle)$$

and

$$Stab(Z_1Z_2) = Span(|00\rangle, |11\rangle).$$

Then, for α to be in the stabilizer code, it must be in the span of $|00\rangle + |11\rangle$. Conversely, this vector is in the stabilizer code of both operators, thus in the stabilizer code of the subgroup generated by them:

$$Stab(\langle X_1X_2, Z_1Z_2 \rangle) = Span(|00\rangle + |11\rangle).$$

Remark. Shor's 9 qubit code is the stabilizer of a subgroup of G_9 ; can you tell which subgroup this is?

8.5.2 Conditions on Pauli subgroups

With the same setting as before, $S \leq \mathcal{H}^{2^n}$, one observation we do before anything else is that, whenever $-I_{2^n} \in S$, the study of this group becomes a bit pointless to our interest, because the stabilizer code is just the trivial subspace. Indeed, for any $|v\rangle \in \mathcal{H}^{2^n}$,

$$|v\rangle \in \text{Stab}(S) \Rightarrow |v\rangle = -|v\rangle \Rightarrow |v\rangle = 0$$

Then, an assumption from now on will be that $-I_{2^n} \notin S$. The next observation gives us a very rich property of our subgroups:

Lemma 78. *For every element g of the Pauli Group G_n , $g^2 = \pm 1$.*

Proof. Since the subset of generators $\{X_i\}$ acts independently on each of the n qubits, and the same happens for $\{Y_i\}, \{Z_i\}$, it will suffice to prove this statement for $n = 1$. This readily follows from the relations

$$X^2 = Y^2 = Z^2 = I$$

and the description (14) of G_1 . □

Recalling the assumption that our subgroup S does not contain $-I_{2^n}$, we must have $g^2 = 1$ for any $g \in S$. An implication of this is that S is itself an Abelian group. Indeed, for any $g, h \in S$:

$$\begin{aligned} g^{-1}h^{-1}gh &= ghgh && \text{because } g = g^{-1} \text{ and } h = h^{-1} \\ &= (gh)^2 \\ &= 1 && \text{thus } gh = hg. \end{aligned}$$

S being an Abelian and finite group, it must be a product of cyclic groups, and since every element has order 1 or 2, then

$$S \simeq \mathbb{Z}_2^{n-k} \text{ for some } 0 < k \leq n.$$

In fact, we have the following very useful property of S :

Theorem 79. *If $S \leq G_n$ is a Pauli Subgroup as above, then $\dim(V_S) = 2^k$.*

Remark. Note how this statement fits the examples we worked on in the previous part of this lecture:

- In Example 1, $n = 2$ and $n - k = 1$, so $\dim(\text{Stab}(Z_1 Z_2)) = 2^k = 2^1 = 2$.
- In Example 2, $n = 2$ and $n - k = 1$, so $\dim(\text{Stab}(X_1 X_2)) = 2^k = 2^1 = 2$.
- In Example 3, it might seem that, $n = 3$ and $n - k = 3$ so we should get the dimension of the stabilizer to be $2^0 = 1$. This is however not the case, because the three generators $Z_1 Z_2, Z_2 Z_3, Z_1 Z_3$ are dependent, so $n - k = 2$ and $\dim(\text{Stab}(Z_1 Z_2)) = 2^1 = 2$.
- In Example 4, $n = 2$ and $n - k = 2$ because the two generators are independent. Then, the dimension of the stabilizer is $2^0 = 1$.

8.5.3 Error Correcting properties of V_S

We aim to answer the question which asks what elements of G_n can we correct against by the stabilizer code. This is an important question, because once we can correct against certain Pauli matrices, by discreditation of error we can correct against several others.

In the same setting, for a Pauli subgroup $S \leq G_n$ let's define

$$N(S) := \{g \in G_n : g^{-1} S g = S\}$$

$$C(S) := \{g \in G_n : g^{-1} h g = h \quad \forall h \in S\}$$

which we will call, respectively, the *normalizer* and the *centralizer* of S . Note that the second object is smaller because it imposes a stricter condition, which is pointwise fixation of the elements of S . Also, conjugation by any element of S does not alter S . So, we have (since S is Abelian)

$$S \subseteq C(S) \subseteq N(S).$$

Just by using that $g^2 = \pm 1$ for all $g \in G_n$, we get

$$h^{-1} g^{-1} h g = \pm h g h g = \pm 1 \quad \Rightarrow \quad g^{-1} h g = \pm h;$$

But both h and $-h$ can not be in the subgroup by the assumption that $-I_{2^n} \notin S$, which implies $g^{-1} h g = h$ for every element of the normalizer of S . Therefore,

$$C(S) = N(S).$$

Theorem 80. Let $\{E_i\}$ be a collection of Pauli errors, $E_i \in G_n$, and let

$$P : \mathcal{H}^{2^n} \rightarrow V_S$$

be the projector into the stabilizer code. Then we have $PE_i^\dagger E_j P \sim P$ whenever either of these two conditions hold:

- a) $E_i^\dagger E_j \in S$;
- b) $E_i^\dagger E_j \notin N(S)$;

as a consequence, if for all i, j one of the two conditions above hold, then it is possible to find an error correction against these Pauli errors.

Part of the strength of this theorem is that by using it, we can cover a wide family of codes. For instance, a good way to check that Shor's 9 qubit code can be corrected against any single qubit error is to check the above condition to hold on any pair of Pauli matrices which generate this code. In order to prove this theorem, we will need to prove first the following Lemma:

Lemma 81. The operator Q on \mathcal{H}^{2^n} defined as follows

$$Q := \frac{1}{|S|} \sum_{g \in S} g$$

is equal to the projector $P : \mathcal{H}^{2^n} \rightarrow V_S$.

Proof. It suffices to show $Q|_{V_S} = id_{V_S}$, and $Q|_{V_S^\perp} = 0$. We show each of these identities.

If $v \in V_S$, then $gv = v$ for all $g \in S$. Then,

$$Qv = \left(\frac{1}{|S|} \sum_{g \in S} g \right) v = \frac{1}{|S|} \sum_{g \in S} gv = \frac{1}{|S|} \sum_{g \in S} v = v = id_{V_S} v$$

If $v \in V_S^\perp$, then $\langle v, V_S \rangle = 0$, so for every $g \in S$ we have

$$\langle gV_S, gv \rangle = \langle V_S, gv \rangle = 0.$$

If we take the average over all $g \in S$ of the above expression, then we get that $\langle V_S, Qv \rangle = 0$, namely that $Qv \in V_S^\perp$. But also $Qv \in V_S$, because

$$\forall \tilde{g} \in S, \quad \tilde{g}(Qv) = \frac{1}{|S|} \sum_{g \in S} \tilde{g}gv = \frac{1}{|S|} \sum_{g \in S} gv = Qv.$$

Therefore, it has to be that $Qv = 0$, proving that $Qv = 0$ for all $v \in V_S^\perp$ and thus our lemma. \square

Proof. We have E_i and E_j satisfy one of the two conditions a), b), and our goal is to show that $PE_i^\dagger E_j P$ is a scalar multiple of P .

a) If $E_i^\dagger E_j \in S$, then for every $v \in \mathcal{H}^{2^n}$ we have the following

$$(PE_i^\dagger E_j P)v = PE_i^\dagger E_j(Pv) = P(Pv) = Pv$$

because $Pv \in V_S$ has to be unchanged by $E_i^\dagger E_j$, and $P^2 = P$. This shows that $PE_i^\dagger E_j P = P$.

b) If $h := E_i^\dagger E_j \notin N(S)$, then let us consider the following double sum:

$$PhP = \frac{1}{|S|^2} \sum_{g_1, g_2 \in S} g_1 h g_2.$$

Now, we recall that, by the properties of Pauli groups, any two elements either anticommute or commute. But $h \notin N(S) = C(S)$ implies that there exists some element $g_* \in S$ such that it anticommutes with h :

$$h^{-1}g_*h = -g_* \quad \Leftrightarrow \quad g_*h = -hg_*$$

Using this element we can apply a useful trick which is often seen in the mathematical field of Representation Theory, and essentially reads as $g_*S = S = Sg_*$:

$$\begin{aligned} PhP &= \frac{1}{|S|^2} \sum_{g_1, g_2 \in S} g_1 h g_2 \\ &= \frac{1}{|S|^2} \sum_{g_1, g_2 \in S} (g_1 g_*) h g_2 \\ &= \frac{1}{|S|^2} \sum_{g_1, g_2 \in S} g_1 (g_* h) g_2 \\ &= \frac{1}{|S|^2} \sum_{g_1, g_2 \in S} g_1 (-h g_*) g_2 \\ &= -\frac{1}{|S|^2} \sum_{g_1, g_2 \in S} g_1 h (g_* g_2) \\ &= -PhP \end{aligned}$$

All in all, we showed that $PhP = -PhP = 0$, which is still a scalar multiple of the projector P .

□

Lecture 19

Scribe: Olga Medrano Martín del Campo, University of Chicago.

Date: March 25th, 2021

9 Extra Material: Quantum Interactive Proofs

Interactive proof classes combine two important notions, which are those of the class BPP and the class NP. The first class, we recall, is defined upon the acceptance conditions: L is a language in BPP if there exists polynomially computable $f(x, y)$ such that if the string r is chosen at random,

$$\begin{cases} \Pr[f(x, r) = 1] \geq \frac{2}{3} & \text{when } x \in L \\ \Pr[f(x, r) = 1] \leq \frac{1}{3} & \text{when } x \notin L \end{cases};$$

The class NP , we also recall is defined upon the acceptance conditions: L is a language in NP if there exists polynomially computable $f(x, y)$ such that for every $x \in L$ there exists a certificate s , with its length polynomial in that of x , such that $f(x, s) = 1$. Next, we will explain how to combine these two notions.

9.1 Classical Merlin-Arthur Proofs

The setting is the following:

- Merlin is an advisor of Arthur, and his goal is to convince Arthur that $x \in L$, where L is a language;
- Merlin gives a certificate s to Arthur, supposedly to be a proof that $x \in L$;
- Merlin is not trustworthy, so the proof might be wrong. To decide whether he accepts it or not, Arthur tosses a coin, and based on the r he decides whether to believe in Merlin or not.

Acceptance conditions: We say that a language L is in MA (as in *Merlin Arthur*) if there exists a polynomially time computable function $f(x, r, s)$ such that

$$\begin{cases} \text{if } x \in L, \Rightarrow \exists s : \Pr[f(x, r, s) = 1] \geq \frac{2}{3} \\ \text{if } x \notin L, \Rightarrow \forall s : \Pr[f(x, r, s) = 1] \leq \frac{1}{3} \end{cases}$$

9.2 Quantum Merlin-Arthur Proofs

The setting is similar to our classical case, except now instead of a function we need a quantum circuit, and instead of a proof we require a state $|\phi\rangle$.

Acceptance conditions: We say a language L is in QMA (*Quantum-Merlin-Arthur*) if there exists a quantum circuit Q acting on the composite Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_M$ such that

$$\begin{cases} \text{if } x \in L, \Rightarrow \exists |\phi\rangle : Pr[Q(|x\rangle \otimes |\phi\rangle) = 1] \geq \frac{2}{3} \\ \text{if } x \notin L, \Rightarrow \forall |\phi\rangle : Pr[Q(|x\rangle \otimes |\phi\rangle) = 1] \leq \frac{1}{3} \end{cases}$$

In other words, when the word is in the language, Merlin will be able to convince Arthur with high probability, but if the word isn't in the language, no matter what Merlin does, it will be unlikely that Arthur will accept Merlin's advice.

Note that in the definition above, ϕ is not specified to be a pure or mixed state. It can be either; Merlin can have a mixed state which can be represented by a probability distribution among states:

$$(p_1, |\phi_1\rangle), \dots, (p_t, |\phi_t\rangle),$$

so that a mixed state would be a convex linear combination of pure states; by linearity of the quantum circuit in the ϕ_i 's, being able to choose one probabilistic distribution implies some ϕ_i can be chosen such that the above conditions hold. Note how we do not need to introduce a random element within the Quantum circuit; the randomness component of this process is hardware into Q itself.

9.2.1 Two Examples

Example: 2-local Hamiltonian. We can have a large Hamiltonian H given by a $2^n \times 2^n$ Hermitian matrix, which, as in the Second Postulate of Quantum Mechanics describes the evolution of our isolated quantum system. 2-locality means that H can be explicitly written as a sum $H = \sum_i H_i$ in which every H_i is a Hermitian matrix depending on ≤ 2 qubits. Thus, H has a succinct representation even if its size is huge.

Then, we have a promise problem through which we want to find out whether H is positive semi definite or far from it. More precisely, we want

to answer the following:

$$\mathcal{A} = \begin{cases} \text{Yes, if } H \text{ is far from being positive semi definite:} & \lambda_{\min} \leq -1 \\ \text{No, if } H \text{ is positive semi definite:} & \lambda_{\min} \geq 0 \end{cases}$$

Theorem: 2-local Hamiltonian is in QMA (and moreover, is QMA -complete).

The idea is simply to replace, like in Schrödinger's equation, the Hermitian matrix H with the unitary matrix $Q = e^{-itH}$, where t is a small parameter. This (or rather its approximation that we are able to achieve) will be Arthur's verification procedure. Negative "energies" λ will translate into eigenvectors $|\phi\rangle$ of Q with eigenvalues $\mu = e^{-it\lambda}$ that will satisfy, as long as t is chosen wisely, $\text{Im}(\mu) > 0$. The honest Merlin will simply report such an eigenvector $|\phi\rangle$.

The actual implementation of these ideas is a bit technical and beyond the scope of this course.

Example 2: *Group non-membership.* The setting is the following:

- G is a finite group, represented as a group of permutations.
- We have a subgroup $H = \langle h_1, \dots, h_k \rangle \leq G$.
- The question being asked is whether a given $g \in G$ is in this subgroup H .

Proving that $g \in H$ is in NP is easy: slightly cheating, a proof of this would only be an expression of g as a word, or product, in the elements h_1, \dots, h_k .

Now, our setting is, that quantum Merlin also wants to be able to prove to Arthur the opposite, that is $g \notin H$. To do this, we first consider the uniform superposition of all the elements in the subgroup:

$$|H\rangle = \frac{1}{\sqrt{|H|}} \sum_{h \in H} |h\rangle,$$

We also consider the unitary operator U_g which permutes the elements of the group, in the same way as in Shor's Factoring algorithm:

$$U_g : |h\rangle \mapsto |hg\rangle,$$

and its controlled version, acting as U_g only when the first control bit is 1:

$$U_g^c : \begin{cases} |0h\rangle \mapsto |0h\rangle \\ |1h\rangle \mapsto |1(hg)\rangle \end{cases}$$

Then, we have the following action of U_g^c on $|+\rangle \otimes |H\rangle$, where we recall, $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$:

$$|+\rangle \otimes |H\rangle \mapsto \frac{|0H\rangle + |1Hg\rangle}{\sqrt{2}}$$

And then, applying the Hadamard gate to the above state, we get

$$\frac{|0\rangle}{2} \otimes (|H\rangle + |Hg\rangle) + \frac{|1\rangle}{2} \otimes (|H\rangle - |Hg\rangle).$$

If we measure the first qubit, we have two possibilities yielding us different outcomes:

- $g \in H$ implies $Hg = H$, so the right summand is zero and the first qubit has measurement probability of 1 at 0.
- $g \notin H$ implies that the states $|H\rangle$ and $|Hg\rangle$ are orthogonal. Then, $P(0) = P(1) = 0.5$; namely, the probabilities of obtaining 0 or 1 in the measurement of our first qubit are equal.

If we ever come across a measurement of the first qubit to be 1, that has to mean $g \notin H$. Conversely, $g \notin H$ will be accurately displayed by our measurement with probability 1/2.

This solution works as long as Merlin will give us the state $|H\rangle$ (that we are incapable of producing by ourselves). However, Merlin can be dishonest and *cheat*, that is attempt to give us a state $|\phi\rangle$ which is far from $|H\rangle$. Showing that our problem is in QMA amounts to showing how we can catch him. Again, we are only able to outline here some rough ideas of the proof.

If Merlin is *reasonably* honest, that is provides us with a state $|\phi\rangle$ that is close enough to $|H\rangle$, then we are still in a good shape as 1 will come up with high probability. It is also relatively easy to design a simple test to catch Merlin cheating but the problem is that our test will be guaranteed to succeed with relatively low probability (of order $1/k$). Then the No-Cloning Theorem provides Merlin with more possibilities of cheating – he knows we may not repeat our test! The solution is to request *Merlin* to do this for us and to come up not only with a single certificate $|\phi\rangle$, but with several copies of it $|\phi\rangle \otimes |\phi\rangle \otimes \dots \otimes |\phi\rangle$. Then, we would need to perform two additional tests, namely:

- *Non-entanglement test*, to make sure that the certificates provided by Merlin are not completely or nearly completely entangled.

- *Consistency test*, which with high probability detects if Merlin tries to feed up with a composite state $|\phi_1\rangle \otimes \dots \otimes |\phi_k\rangle$ in which many pairs $|\phi_i\rangle, |\phi_j\rangle$ are not close to each other.

And, as always, the catch is that we do not tell Merlin in advance which one of this bunch of tests we are going to perform (remember No-Cloning!) but decide only after he commits himself to the certificate, by flipping a (quantum) coin.

Acknowledgement

I wish to express my sincere gratitude to all scribes in 2011, 2013 and 2021 whose effort and contributions made this project possible. My thanks are also due to Leonardo Coregliano for the thorough proof checking he did in 2015 that has greatly helped to improve the presentation.

References

- [1] M. A. Nielsen and I. L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2000.
- [2] R. Laflamme P. Kaye and M. Mosca. *An Introduction to Quantum Computing*. Oxford University Press, 2007.
- [3] A. Kitaev, A. Shen, and M. Vyalii. *Classical and quantum computation*. American Math. Society, 2002. Extended version of a book originally published in Russian.
- [4] Ajtai M and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the 29th ACM STOC*, pages 284–293, 1997.
- [5] O. Regev. Quantum computation and lattice problems. *SIAM Journal on Computing*, 33(3):738–760, 2004.
- [6] H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288:21–43, 2002.
- [7] P. Hatami, R. Kulkarni, and D. Pankratov. Variations on the sensitivity conjecture. *Theory of Computing Library, Graduate Surveys*, 4:1–27, 2011.

- [8] A. Ambainis, A. M. Childs, B. W. Reichardt, R. Spalek, and S. Zhang. Any AND-OR formula of size N can be evaluated in time $N^{1/2+o(1)}$ on a quantum computer. *SIAM Journal on Computing*, 39(6):2513–2530, 2010.
- [9] A. Yao. Some complexity questions related to distributive computing. In *Proceedings of the 11th ACM Symposium on the Theory of Computing*, pages 209–213, New York, 1979. ACM Press.
- [10] K. Mehlhorn and E. M. Schmidt. Las Vegas is better than determinism in VLSI and distributive computing. In *Proceedings of the 14th ACM Symposium on the Theory of Computing*, pages 330–337, New York, 1982. ACM Press.
- [11] A. Yao. Quantum circuit complexity. In *Proceedings of the 34th IEEE Symposium on Foundations of Computer Science*, pages 352–361, Los Alamitos, 1993. IEEE Computer Society.
- [12] R. Raz. Exponential separation of quantum and classical communication complexity. In *Proceedings of the 31st ACM STOC*, pages 358–367, 1999.
- [13] H. Buhrman, R. Cleve, and A. Wigderson. Quantum vs. classical communication and computation. In *Proceedings of the 30th ACM Symposium on the Theory of Computing*, pages 63–86, New York, 1998. ACM Press. Preliminary version available at quant-ph/9802040.
- [14] A. Razborov. Quantum communication complexity of symmetric predicates. *Izvestiya: Mathematics*, 67(1):145–159, 2003.
- [15] I. Kremer. Quantum communication. Master’s thesis, Hebrew University, Jerusalem, 1995.
- [16] A. Razborov and A. Sherstov. The sign-rank of AC^0 . *SIAM Journal on Computing*, 39(5):1833–1855, 2010.
- [17] R. Paturi. On the degree of polynomials that approximate symmetric Boolean functions. In *Proceedings of the 24th ACM Symposium on the Theory of Computing*, pages 468–474, New York, 1992. ACM Press.
- [18] H. Klauck. Lower bounds for quantum communication complexity. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, pages 288–297, Los Alamitos, 2001. IEEE Computer Society. Preliminary version available at quant-ph/0106160.

- [19] A. Sherstov. *Lower bounds in Communication Complexity and Learning Theory via Analytic Methods*. PhD thesis, University of Texas at Austin, 2009.
- [20] Nati Linial and Adi Shraibman. Lower bounds in communication complexity based on factorization norms. *Random Structures and Algorithms*, 34(3):368–394, 2009.
- [21] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *J. Symbolic Computation*, 9(3):251–280, 1990.
- [22] T. Lee, A. Shraibman, and R. Spalek. A direct product theorem for discrepancy. In *IEEE Conference on Computational Complexity*, pages 71–80, 2008.