Part III — Quantum Computation

Definitions

Based on lectures by R. Jozsa

Notes taken by Dexter Chua

Michaelmas 2016

Quantum mechanical processes can be exploited to provide new modes of information processing that are beyond the capabilities of any classical computer. This leads to remarkable new kinds of algorithms (so-called quantum algorithms) that can offer a dramatically increased efficiency for the execution of some computational tasks. Notable examples include integer factorisation (and consequent efficient breaking of commonly used public key crypto systems) and database searching. In addition to such potential practical benefits, the study of quantum computation has great theoretical interest, combining concepts from computational complexity theory and quantum physics to provide striking fundamental insights into the nature of both disciplines.

The course will cover the following topics:

Notion of qubits, quantum logic gates, circuit model of quantum computation. Basic notions of quantum computational complexity, oracles, query complexity.

The quantum Fourier transform. Exposition of fundamental quantum algorithms including the Deutsch-Jozsa algorithm, Shor's factoring algorithm, Grovers searching algorithm.

A selection from the following further topics (and possibly others):

(i) Quantum teleportation and the measurement-based model of quantum computation;

(ii) Lower bounds on quantum query complexity;

(iii) Phase estimation and applications in quantum algorithms;

(iv) Quantum simulation for local hamiltonians.

Pre-requisites

It is desirable to have familiarity with the basic formalism of quantum mechanics especially in the simple context of finite dimensional state spaces (state vectors, Dirac notation, composite systems, unitary matrices, Born rule for quantum measurements). Prerequisite notes will be provided on the course webpage giving an account of the

necessary material including exercises on the use of notations and relevant calculational techniques of linear algebra. It would be desirable for you to look through this material at (or slightly before) the start of the course. Any encounter with basic ideas of classical theoretical computer science (complexity theory) would be helpful but is not essential.

Contents

0 Introduction

1 Classical computation theory

Definition (Input string)**.** An *input bit string* is a sequence of bits $x = i_1 i_2 \cdots i_n$, where each i_k is either 0 or 1. We write B_n for the set of all n-bit string, and $B = \bigcup_{n\in\mathbb{N}} B_n$. The *input size* is the length n. So in particular, if the input is regarded as an actual number, the size is not the number itself, but its logarithm.

Definition (Language)**.** A *language* is a subset $L \subseteq B$.

Definition (Decision problem)**.** Given a language L, the *decision problem* is to determine whether an arbitrary $x \in B$ is a member of L. The output is thus 1 bit of information, namely yes or no.

Definition (Computational model)**.** A *computational model* is a process with discrete steps (elementary computational steps), where each step requires a constant amount of effort/resources to implement.

Definition (Randomized/probabilistic computation)**.** This is the same as a usual computational model, but the process also has access to a string $r_1, r_2, r_3, \cdots$ of independent, uniform random bits. In this case, we will often require the answer/output to be correct with "suitably good" probability.

Definition (Complexity of a computational task (or an algorithm))**.** The *complexity* of a computational task or algorithm is the "consumption of resources as a function of input size n". The resources are usually the time

$$T(n) = \text{number of computational steps needed},$$

and space

$$Sp(n) = \text{number of memory/work space needed}.$$

In each case, we take the worse case input of a given size n.

Definition (Polynomial growth)**.** We say $T(n)$ *grows polynomially*, and write

$$T(n) = O(\text{poly}(n)) = O(n^k)$$

for some k, if there is some constant c, and some integer k and some integer n_0 such that $T(n) < cn^k$ for all $n > n_0$.

2 Quantum computation

Definition (Approximate universality)**.** A collection of gates is *approximately universal* if for any unitary matrix U and any $\varepsilon > 0$, there is some circuit $\tilde{U}$ built out of the collection of gates such that

$$\left\| U - \tilde{U} \right\| < \varepsilon.$$

In other words, we have

$$\sup_{\|\psi\|=1} \left\| U|\psi\rangle - \tilde{U}|\psi\rangle \right\| < \varepsilon,$$

where we take the usual norm on the vectors (any two norms are equivalent if the state space is finite dimensional, so it doesn't really matter).

Definition (**BQP**)**.** The complexity class **BQP** (bounded error, quantum polynomial time) is the class of all decision problems computable with polynomial quantum circuits with at least 2/3 probability of being correct.

3 Some quantum algorithms

3.1 Balanced vs constant problem

3.2 Quantum Fourier transform and periodicities

Definition (Quantum Fourier transform mod N)**.** Suppose we have an N-dimensional state space with basis $|0\rangle, |1\rangle, \cdots, |N-1\rangle$ labelled by $\mathbb{Z}/N\mathbb{Z}$. The *quantum Fourier transform mod N* is defined by

$$\mathsf{QFT} : |a\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{b=0}^{N-1} e^{2\pi i ab/N} |b\rangle .$$

The matrix entries are

$$[\mathsf{QFT}]_{ab} = \frac{1}{\sqrt{N}} \omega^{ab}, \quad \omega = e^{2\pi i/N},$$

where $a, b = 0, 1, \cdots, N-1$. We write QFT_n for the quantum Fourier transform mod n.

3.3 Shor's algorithm

3.4 Search problems and Grover's algorithm

Definition (Verifier)**.** Suppose we have a language $L \subseteq B^*$, where

$$B^* = \bigcup_{n \in \mathbb{N}} B_n$$

is the set of all bit strings.

A *verifier* for L is a computation $V(w, c)$ with two inputs w, c such that

(i) V halts on all inputs.

(ii) If $w \in L$, then for *some* c, $V(w, c)$ halts with "accept".

(iii) If $w \notin L$, then for *all* c, $V(w, c)$ halts with "reject".

A *polynomial time verifier* is a V that runs in polynomial time in $|w|$ (not $|w| + |c|$!).

Definition (Non-deterministic polynomial time problem)**. NP** is the class of languages that have polynomial time verifiers.

3.5 Amplitude amplification

4 Measurement-based quantum computing

Notation. We write

$$|\pm_\alpha\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm e^{-i\alpha}|1\rangle).$$

In particular, we have

$$|\pm_0\rangle = |\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$$

Then

$$\mathcal{B}(\alpha) = \{|+_\alpha\rangle, |-_\alpha\rangle\}$$

is an orthonormal basis. We have 1-qubit gates

$$\mathsf{J}(\alpha) = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & e^{i\alpha}\\ 1 & -e^{i\alpha}\end{pmatrix} = \mathsf{HP}(\alpha),$$

where

$$\mathsf{H} = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, \quad \mathsf{P}(\alpha) = \begin{pmatrix}1 & 0\\ 0 & e^{i\alpha}\end{pmatrix}.$$

We also have the "Pauli gates"

$$\mathsf{X} = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}, \quad \mathsf{Z} = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix} = \mathsf{P}(\pi)$$

We also have the 2-qubit gates

$$\mathsf{E} = \mathsf{CZ} = \mathrm{diag}(1, 1, 1, -1).$$

We also have 1-qubit measurements

$$\mathsf{M}_i(\alpha) = \text{measurement of qubit } i \text{ in basis } \mathcal{B}(\alpha).$$

The outcome $|+_\alpha\rangle$ is denoted 0 and the outcome $|-_\alpha\rangle$ is denoted 1.

We also have $\mathsf{M}_i(\mathsf{Z})$, which is measurement of qubit i in the standard basis $\{|0\rangle, |1\rangle\}$.

Finally, we have the notion of a graph state. Suppose we have an undirected graph $G = (V, E)$ with vertices V and edges E with no self-loops and at most one edge between two vertices, we can define the *graph state* $|\psi_G\rangle$ that is a state of $|V|$ qubits as follows: for each vertex $i \in V$, introduce a qubit $|+\rangle_i$. For each edge $e : i \to j$, we apply E_{ij} (i.e. E operating on the qubits i and j). Since all these E_{ij} commute, the order does not matter.

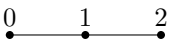
Example. If G_1 is

then we have

$$|\psi_{G_1}\rangle = E_{12}|+\rangle_1|+\rangle_2 = \frac{1}{2}[|00\rangle + |01\rangle + |10\rangle - |11\rangle],$$

and this is an entangled state.

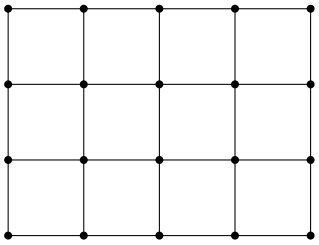
If G_2 is



then we have

$$|\psi_{G_2}\rangle = E_{12}E_{23}\,|+\rangle_1\,|+\rangle_2\,|+\rangle_3\,.$$

A *cluster state* is a graph state $|\psi_G\rangle$ for G being a rectangular $2D$ grid.



5 Phase estimation algorithm

6 Hamiltonian simulation

Definition (Operator norm)**.** The *operator norm* of an operator A is

$$\|A\| = \max_{\||\psi\rangle\|=1} \|A\,|\psi\rangle\| \,.$$

If A is diagonalizable, then this is the maximum eigenvalue of A.

Definition (k-local Hamiltonian)**.** We say a Hamiltonian H is k-local (for k a fixed constant) on n qubits if

$$H = \sum_{j=1}^{m} H_j,$$

where each H_j acts on at most k qubits (not necessarily adjacent), i.e. we can write

$$H_j = \tilde{H}_j \otimes I,$$

where $\tilde{H}_j$ acts on some k qubits, and I acts on all other qubits as the identity.

Notation. For a matrix X, we write

$$X + O(\varepsilon)$$

for $X + E$ with $\|E\| = O(\varepsilon)$.