UNIVERSITY OF LONDON

Imperial College of Science, Technology and Medicine
Department of Computing

Complexity Analysis and Semantics for Quantum Computation

Elham Kashefi

Thesis submitted in partial fulfilment of the
requirements for the degree of
Doctor of Philosophy
of the University of London
and the Diploma of Membership of Imperial College.

November 26, 2003

Abstract

This thesis focuses on two different aspects of quantum computation: quantum complexity and quantum semantics. In the first part, I study the quantum complexity mainly within the quantum query model. Together with my colleagues, I introduce a new framework for quantum query complexity, phrased in terms of the minimal oracle and analyse limits and strengths of this new model in comparison to their classical and quantum counterparts. Working within the query model I study quantum one-way functions. I show that in the quantum setting, the problem of the existence of a quantum one-way permutation can be reduced to the problem of constructing polynomial size networks for performing the specific task of the reflection about a sequence of states. Furthermore, I extend these results to the domain of the state and operator complexity. I show that if a quantum one-way function exists, then we can construct a sequence of so called “hard” states with the property that the reflection operators about those states are efficiently implementable.

In the second part, I study the extension of domain theory to the quantum setting and develop the semantics of quantum computation. By defining a quantum domain I introduce a rigourous definition of quantum computability for quantum states and operators. Furthermore I show that the denotational semantics of quantum computation has the same semantical structure as the denotational semantics of classical probabilistic computation introduced by Kozen. This could be considered as a foundation for designing functional programming languages for quantum computation. Finally, I continue with an abstract mathematical approach to study a general formalism for describing entanglement manipulation and introduce a new approach to derive a unique measure of entanglement for bipartite quantum pure states.

Acknowledgments

Many people have guided me with their knowledge and supported me with their love during the completion of this thesis. My deepest gratitude goes to my supervisor Vlatko Vedral who taught me many interesting things with the patience of a great teacher and helped me truly in many situations with the affection of a wonderful friend. I am grateful to Steffen van Bakel my co-supervisor, Chris Hankin and Peter Knight for their advice, support and provision of funding for my research.

Special thanks to my co-authors and collaborators, Konrad Banaszek, Jens Eisert, Adrian Kent, Harumichi Nishimura, Vlatko Vedral, Herbert Wiklicky, most of the work in this thesis has been inspired by our joint discussions.

I thank particulary my dear friend, officemate and collaborator, Ivette Fuentes-Guridi who introduced me first to this fascinating quantum world and passionately encouraged me to carry out my research in this area. Many thanks to my dear friends Angelo Carollo, Damian Markham and Jesús Rogel, their friendship, constant technical, scientific and moral support are much appreciated.

My research life have benefited enormously both scientifically and socially from meeting and discussing with the following people from all over the world: Scott Aaronson, Mauricio Alvarez-Manilla, Katia Babbar, Stephen Bartlett, Andrej Bauer, Almut Beige, Charles Bennett, Sougato Bose, Vasco Brattka, Adam Brazier, Hans. J. Briegel, Dan Brown, Hilary Carteret, Richard Cleve, Wim van Dam, Ivona Dragun, Christoph Dürr, Artur Ekert, Lindsay Errington, Ernesto Galvão, Lov Grover, Rudy Raymond Harry Putra, Reinhold Heckmann, Peter Høyer, Lawrence Ioannou, Richard Jozsa, Phillip Kaye, Julia Kempe, Viv Kendon, Koji Kobayashi, Marko Krznaric, Daniel Lidar, Seth Lloyd, Hoi-Kwong Lo, Frédéric Magniez, Istvan Maros,

Koji Maruyama, Keiji Matsumoto, Michele Mosca, Mio Murao, Rajagopal Nagarajan, Yasser Omar, Nick Ovenden, Jannis Pachos, Nikola Paunkovic, Carlos Perez, Luke Rallan, Robert Raussendorf, Caroline Rogers, Terry Rudolph, Barry Sanders, Miklos Santha, Marcelo Santos, Stefan Scheel, Christoph Simon, Fernando Souza, Alain Tapp, Shashank Virmani, John Watrous, Klaus Weihrauch, Christof Zalka.

Above all, the endless love and encouragements of Leily, Nezam, Maryam and Reza made it possible for me to follow my dreams.

Contents

1 Preliminary Materials **8**

1.1 Mathematical Structures 9

1.1.1 Linear Spaces 10

1.1.2 Measure and Probability 12

1.2 Quantum Mechanics 14

1.2.1 Hilbert Space Framework 14

1.2.2 Technical Developments 19

1.3 Quantum Computation 21

1.3.1 Quantum Turing Machine 22

1.3.2 Quantum Circuit Model 24

1.3.3 Complexity Analysis 27

1.3.4 Quantum Query Model 28

2 Quantum One-way Function **31**

2.1 Introduction 31

2.2 Worst Case Complexity 34

2.3 Average Case Complexity 41

2.4 Complexity Classes 52

2.5 State and Operator Complexity 54

2.6 Discussion 59

3 Quantum Oracle **61**

3.1 Introduction 61

3.2 Minimal Oracle 61
3.3 Promise Problems 66
3.4 Discussion 71

4 Quantum Domain Theory **73**
4.1 Introduction 73
4.2 Classical Domain Theory 74
4.2.1 Computability Analysis 79
4.2.2 Denotational Semantics 80
4.3 Quantum Setting 84
4.3.1 Computability Analysis 85
4.3.2 Denotational Semantics 93
4.4 Information Theory 96
4.5 Discussions 100

5 Axiomatic Information Theory **101**
5.1 Introduction 101
5.2 Formal Theory 102
5.3 Thermodynamics 105
5.4 Entanglement Manipulation 106
5.5 Elementary Classical Information 109
5.6 Discussion 109
References 111

List of Publications

- **On quantum one-way permutations.** E. Kashefi, H. Nishimura and V. Vedral, *Quantum Information and Computation*, **5**, 379, 2002.
- **Physical reversibility and one-way functions.** E. Kashefi and V. Vedral, *Proceedings of QCMC'02 – The Sixth International Conference on Quantum Communication, Measurement and Computing*, Boston, 2002.
- **A comparison of quantum oracles.** E. Kashefi, A. Kent, V. Vedral and K. Banaszek, *Phys. Rev. A*, **65**, 05304, 2002.
- **Quantum Domain Theory - Definitions and Applications.** E. Kashefi, *Proceedings of CCA'03 – The International Conference on Computability and Complexity Analysis*, Cincinnati, 2003.
- **Uniqueness of entanglement measure and thermodynamics.** V. Vedral and E. Kashefi, *Phys. Rev. Lett.*, **89**, 037903, 2002.
- **A unified axiomatic approach to information content of physical states.** V.Vedral and E. Kashefi, *Proceedings of QCMC'02 – The Sixth International Conference on Quantum Communication, Measurement and Computing*, Boston, 2002.

1

Preliminary Materials

The topic of this thesis lies in the new and rapidly growing field of quantum computing, which explores connections between physics and computing in general. Quantum information processing is a cross-disciplinary field and is of great importance from both a fundamental, as well as technological perspective [79]. From the fundamental perspective we have deepened our understanding of the relationship between physics, information and computation in general, and have also gained a deeper understanding of the fundamental aspects of quantum theory - non-locality and entanglement in particular [105]. From the technological perspective we have manipulated larger and larger quantum systems and obtained powerful practical applications in the domain of communication and cryptography such as the unconditionally secure quantum cryptography (key exchange) and quantum teleportation [19, 42, 13].

Historically, the greater potential of the quantum computer was first realised by Feynman, who noted that quantum systems appear to be exponentially hard to simulate with classical computers [45]. He speculated that, therefore, quantum computers could potentially be much more powerful than their classical counterparts. This intuition has been proven to be correct for some tasks, such as factoring large numbers and searching unstructured databases. Every computer is fundamentally a physical system, and any computation is just a physical process undergone by

this system. Quantum physics is the most accurate way of describing physical systems and their behaviour in general. Encoding information into quantum systems and processing it according to the laws of quantum physics results in new features which do not exist in the classical computation.

Large scale quantum computation is still hypothetical. However, Moore's law[1] predicts that technology will reach the level where the quantum effects become important in near future. Parallel to this there is a growing effort to build quantum computers by manipulating larger numbers of quantum systems. Steady progress has now led to ion trap quantum computers with 4 qubits [101], Nuclear Magnetic Resonance (NMR) schemes with 7 qubits [65, 26] and realistic proposals for quantum computing in solid state environments [70]. Simple quantum algorithms such as the Deutsch-Jozsa algorithm [34] or quantum database search algorithms [53] have been experimentally demonstrated in NMR schemes and further progress towards higher numbers of qubits (10) seems likely in the foreseeable future.

Either way, we will enter the quantum realm where every aspect of computing, including storing information, loading and running of programs and reading the output will be governed by laws of quantum physics which are completely different from those of classical physics. Therefore there is a great need for theoretical study of quantum computation. The aim of this thesis is to study the quantum effects on computational complexity and semantics of computation.

In this chapter we present all the required preliminary materials for this thesis. First we briefly review the mathematical structures which we will refer to later in this thesis. Subsequently, we describe the mathematical foundation of quantum mechanics and finally, we discuss the basis of the theory of quantum computation.

1.1 Mathematical Structures

We will use the notation and terminology of the following books: *Measure Theory* by Halmos [55]; *Probability Theory* by Feller [44] and Chung [27]; *Linear Analysis*

[1]Gordon Moore, one of the founders of the Intel, observed in mid 1960's that the memory capacity of a typical chip doubles roughly every eighteen months while its physical size remains the same.

by Dunford and Schwartz [37]. In addition Thirring [100] is an excellent introduction to *Mathematical Physics*.

We begin by recollecting the basic definitions and theorems in linear spaces.

1.1.1 Linear Spaces

The linear spaces are the mathematical structure of quantum mechanics as we will describe in the next section.

Definition 1 *A* vector space *over complex numbers* $\mathbb{C}$ *is a set* V *equipped with a* sum operator $V \times V \to V : (u, v) \mapsto u + v = v + u$ *and a* scaler product $V \times \mathbb{C} \to V : (u, \alpha) \mapsto \alpha u$ *such that the following conditions are satisfied:*

(i) $(V, +)$ *is an Abelian group.*

(ii) $\alpha_1(\alpha_2 v) = (\alpha_1 \alpha_2)v$.

(iii) $\alpha(u + v) = \alpha u + \alpha v$.

(iv) $(\alpha_1 + \alpha_2)v = \alpha_1 v + \alpha_2 v$.

(v) $1v = v$.

(iv) $1v = v$.

A subset $V_1 \subset V$ *which is also a vector space is called a* subspace *of* V.

By the axiom of choice, it is always possible to find a *Hamel basis* $\{e_\gamma\}, \gamma \in \mathbf{I}$, such that every vector can be written uniquely as

$$v = \sum_{\text{finite}} \alpha_i e_{\gamma_i}, \quad \alpha_i \in \mathbb{C}\,.$$

The cardinality of I is known as the *algebraic dimension* of the space.

Definition 2 *A* norm *on a vector space* V *is a map* $\|\| : V \to \mathbb{R}^+$ *such that:*

- $\|v\| = 0$ *iff* $v = 0$.

- $\|\alpha v\| = |\alpha|\|v\|$ *for all scalars* α.
- $\|u + v\| \leq \|u\| + \|v\|$.

The norm induces a metric on V *where the distance between* u *and* v *is* $\|u - v\|$. *If* V *is complete with respect to this metric, then* V *is called a* Banach space.

Definition 3 *A* scalar product *(or* inner product*) on a complex vector space* V *is a map* $\langle\,|\,\rangle : V \times V \to \mathbb{C}$ *such that:*

- $\langle u|(\alpha_1 v_1 + \alpha_2 v_2)\rangle = \alpha_1\langle u|v_1\rangle + \alpha_2\langle u|v_2\rangle$.
- $\langle u|v\rangle^* = \langle v|u\rangle$.
- $\langle v|v\rangle \geq 0$ *with* $\langle v|v\rangle = 0$ *iff* $v = 0$.

The scalar product induces a norm on V *where the* $\|v\|^2 = \langle v|v\rangle$. *If* V *is complete with respect to this norm, then* V *is called a* Hilbert space.

Remark.

1. It is possible to introduce a smaller basis than the Hamel basis for complete normed spaces V (e.g. Banach and Hilbert spaces). A set of vectors $\{e_\gamma\}$, where $\gamma \in \mathbf{I}$, is said to be *total* whenever the set of its finite linear combinations is dense in V. If $\mathbf{I}$ is countable, then V is *separable* (as a topological space).

2. By the axiom of choice, the e_γ can even be chosen to be orthonormal in a Hilbert space. If this has been done and $v = \sum_{\gamma\in\mathbf{I}} c_\gamma e_\gamma, c_\gamma = \langle e_\gamma|v\rangle$ then $\|v\|^2 = \sum_{\gamma\in\mathbf{I}} |c_\gamma|^2$, and the Hilbert space can be considered as $L^2(\mathbf{I}, \mu)$ where μ assigns every element of $\mathbf{I}$ the measure 1. If $\mathbf{I}$ is countable, then the Hilbert space is isomorphic to an l^2 space.

Definition 4 *A* linear map *between two vector spaces* U *and* V *is a mapping* $A : U \to V$ *such that:*

$$A(\alpha u_1 + \beta u_2) = \alpha A(u_1) + \beta A(u_2) \;\; \textit{for all} \;\; \alpha, \beta \in \mathbb{C} \;\; \textit{and} \;\; u_1, u_2 \in V_1\,.$$

The set of all linear maps $A : U \to V$, denoted by $\mathcal{L}(U, V)$, is itself a vector space, and $\mathcal{B}(V) = \mathcal{L}(V, V)$. The elements $A \in \mathcal{L}(U, V)$ are also called operators.

A linear functional *on a vector space V is a linear map between V and $\mathbb{C}$. The vector space of all linear functionals on V is called its* dual space*, and is denoted by V^*.*

In a vector space V with scalar product $\langle\,|\,\rangle$, a natural map between V and its dual space V^* can be defined as follows. To each vector $v \in V$ associate a map $A_v : V \to \mathbb{C}$ defined by:

$$A_v(u) = \langle v|u\rangle\,.$$

1.1.2 Measure and Probability

The following definitions from measure theory are required for the discussion of the semantics of quantum computation in Chapter 4.

A *measurable space* is a pair (X, M) where X is a set and M is a σ-algebra of subsets of X, i.e. M is a Boolean algebra of subsets of X closed under countable union. Elements of M are called *measurable sets* or *events* and are denoted by $B, C, \cdots$ and $\neg B$ denotes the complement of B in X.

Definition 5 *A function $f : (X, M) \to (Y, N)$ is* measurable *iff for all $B \in N$ we have $f^{-1}(B) \in M$.*

Let (X_n, M_n) be a sequence of measurable spaces and let $\prod_n X_n$ be the direct product of the X_n with projection $\pi_i : \prod_n X_n \to X_i$. The *cartesian product* $\prod_n (X_n, M_n)$ is the space $(\prod_n X_n, M)$, where M is the smallest σ-algebra containing all *cylinders* $\pi_i^{-1}(B), B \in M_i$.

Definition 6 *A* measure *or* distribution *μ on (X, M) is a function in $M \to \mathbb{R}$ that is countably additive, i.e., $\mu(\cup_n B_n) = \sum_n \mu(B_n)$ where $\{B_n\}$ is a countable set of pairwise disjoint elements of M.*

A measure is positive *iff $\forall B \in M : \mu(b) \geq 0$. It is a* probability measure *if it is positive and $\mu(X) = 1$ and a subprobability measure if it is positive and $\mu(X) \leq 1$.*

If X and Y are two measurable spaces and μ and ν are measures over them, then the *product* of μ and ν, denoted by $\mu \times \nu$ is a measure on the cartesian product $X \times Y$ defined with:

$$(\mu \times \nu)(B \times C) = \mu(B) \times \nu(C)\,.$$

Definition 7 *Assume μ is a measure and $B \in M$ is given. The* conditional probability *of μ relative to B is defined with $\mu_B/\mu(B)$ where $\mu_B(A) = \mu(A \cap B)$.*

Every measure can be decomposed into its positive and negative parts: to every measure μ there correspond unique positive measures μ^+ and μ^- such that for some $B \in M$ we have $\mu^+ = \mu_B$ and $\mu^- = -\mu_{\neg B}$. This is called the *Jordan decomposition* of μ.

Definition 8 *The* total variation *or* absolute value *of μ is the measure $|\mu| = \mu^+ + \mu^-$. The* total variation norm *is a map $\|\| : \mathbf{B} \to \mathbb{R}^+$ associating with each measure μ the non-negative real number $\|\mu\| = |\mu|(X)$.*

A *measure space* (X, M, μ) is a measurable space equipped with a measure. A *probability space* is a measure space (X, M, μ) where μ is a probability measure. A *random variable* is a partial measurable function whose domain is a probability space.

A random variable $x : (X, M, \mu) \to (Y, N)$ induces a subprobability measure $\mu \circ x^{-1}$ on (Y, N):

$$\mu \circ x^{-1}(B) = \mu(x^{-1}(B))\,.$$

If x is total then $\mu \circ x^{-1}$ is a probability measure. When the domain of x is clear we denote the value of $\mu \circ x^{-1}(A)$ by $\mathrm{Prob}(x \in A)$.

A random vector is a list of random variables

$$x_i : (X, M, \mu) \to (Y_i, N_i)$$

with the same domain. Equivalently, a *random vector* is a random variable from (X, M, μ) into the cartesian product $\prod_i (Y_i, N_i)$.

Definition 9 *The* joint distribution *of the random variables* $x = x_1, x_2, \cdots$ *is the subprobability measure* $\mu \circ x^{-1}$ *on* $\prod_i (Y_i, N_i)$ *induced by* x*.*

1.2 Quantum Mechanics

Plank, Einstein and Bohr obtained the early great success in the quantum theory in the period from 1900 to 1925. Nevertheless, up to this time there existed no complete mathematical system for quantum theory to capture everything known up to that time in a unified picture. The year 1925 brought the resolution. A procedure initiated by Heisenberg was developed by Born, Heisenberg, Jordan and a little later by Dirac, into a new system of quantum theory. A little later Schrödinger developed the wave mechanics from an entirely different starting point. These two procedures, known as Heisenberg's and Schrödinger's pictures , soon proved to be equivalent.

There are two main mathematical frameworks within which quantum theory can be developed. One takes as its central object a certain algebraic structure (a C* algebra) on the set of physical observables. States are then defined in relation to this algebra. On the other hand in the well-known Hilbert space approach the primary object is the vector space of states, with observables being defined in relation to this space. In this thesis we only work within the latter frameworks. A brief review of the Hilbert space framework for quantum mechanics has been described in what follows.

We will use the notation and terminology of the following books: Quantum Theory by Isham [63]; Quantum Computation and Quantum Information by Nielsen and Chuang [79]; and Mathematical Foundation of Quantum Mechanics by von Neumann [108].

1.2.1 Hilbert Space Framework

In 1925 Schrödinger proposed one of the first formulations of quantum mechanics. His structure, known as *wave mechanics*, can be generalised within the Hilbert Space framework where the mathematical tool to describe the physical postulates is linear algebra. The standard notation of quantum mechanics for linear algebraic

concepts was introduced by Dirac in 1920.

In Dirac's notation, a vector in the state space is represented with $|\psi\rangle$. The state space of a physical system is a Hilbert space. Postulates 1 below will formalise this fact. The dual of the vector $|\psi\rangle \in \mathcal{H}$ is the function

$$\begin{aligned} \langle\psi| : \mathcal{H} &\rightarrow \mathbb{C} \\ |\phi\rangle &\mapsto \langle\psi|\phi\rangle \,, \end{aligned}$$

where $\langle .|.\rangle$ is the inner product of the two vectors. A linear map (operator, transformation) is always represented by a matrix, A. The following tables gives a summary of the Dirac's notation.

Notation	Description
z^*	Complex conjugate of the complex number z.
$\|\psi\rangle$	Vector. Also known as a *ket*.
$\langle\psi\|$	Vector dual to $\|\psi\rangle$. Also known as a *bra*.
$\langle\phi\|\psi\rangle$	Inner vector product.
$\|\phi\rangle \otimes \|\psi\rangle$	Tensor vector product. For simplicity we omit $\otimes$ and just write $\|\phi\rangle\|\psi\rangle$ or $\|\phi, \psi\rangle$.
A^*	Complex conjugate of the matrix A.
A^T	Transpose of the matrix A.
$A^\dagger$	Hermitian conjugate of the matrix A, $A^\dagger = (A^T)^*$.
$A\|\psi\rangle$	Application of operator A on vector $\|\psi\rangle$.
$\langle\phi\|A\|\psi\rangle$	The inner product of $\|\phi\rangle$ and $A\|\psi\rangle$, $\langle\phi\|(A\|\psi\rangle)$.

The four postulates that follow deal with the general mathematical framework within which it has been found possible so far to describe all quantum mechanical systems.

The first postulate sets up the state space in which quantum mechanics takes place.

Postulate 1. The predictions of results of measurements of an isolated system are probabilistic in nature. In situations where the maximum amount of information is

available, this probabilistic information is represented mathematically by a vector in a complex Hilbert space $\mathcal{H}$ that forms the state space of the quantum theory. This vector is thought to be the mathematical representative of the physical notion of *state* of the system. In this framework, a physical observable is represented by a Hermitian matrix.

The following postulate is concerned with the evolution of the system.

Postulate 2. In a *closed* system, the evolution of the system is described by a *unitary transformation*. That is, the state $|\psi_1\rangle$ of the system at time t_1 is related to the state $|\psi_2\rangle$ at time t_2 by a unitary operator U which depends only on the time t_1 and t_2,

$$|\psi_2\rangle = U|\psi_1\rangle \, .$$

A refined version of this postulates describes the continuous time evolution of the system as follows.

Postulate 2′. The state vector $|\psi(t)\rangle$ of a closed system changes smoothly in time t according to the time-dependent Schrödinger equation

$$i\hbar \frac{d|\psi(y)\rangle}{dt} = \widehat{H}|\psi(y)\rangle \, .$$

In the above formula, $\hbar$ is the Planck's constant $\hbar \approx 6.63 \times 10^{-34}$ Joule-second divided by $t\pi$ and $\widehat{H}$ is the Hamiltonian operator which is described by a Hermitian matrix.

The next postulate describes the effect of observing (measurement) a quantum system.

Postulate 3. Quantum measurements are described by a collection M_m of *measurements operators*. These are operators acting on the state space of the system being measured. The index m refers to the measurements outcome that may occur in the experiment. If the state of the quantum system is $|\psi\rangle$ immediately before the

measurement then the probability that the result m occurs is given by

$$p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle\,,$$

and the state of the system after the measurement is

$$\frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}}\,.$$

The measurements operators satisfy the *completeness equation*,

$$\sum_m M_m^\dagger M_m = I\,.$$

The last postulate deals with composite quantum system.

Postulate 4. The state space of a composite physical system is the tensor product of the state spaces of the component physical systems. Moreover, if we have systems numbered 1 to n, and system i is prepared in the state $|\psi_i\rangle$, then the joint state of the total system is $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$.

In other word, the first postulate describes the encoding of the information, the second postulates explains the process of information, the third postulate deals with retrieving the information and finally the last postulates speaks about combining different systems.

Mixed states arise when we do not have complete information about the state of the physical system. This is always the case in experiments, since the system we are trying to prepare in a pure state interacts with an uncontrolled environment. A mixed state is a probabilistic mixture of pure states, denoted by $\{p_i, |\psi_i\rangle\}$ or alternatively with a *density matrix*

$$\rho \equiv \sum_i p_i|\psi_i\rangle\langle\psi_i|\,.$$

A density matrix $\rho \in B(\mathcal{H}$ is a hermitian (i.e. $\rho = \rho^\dagger$) semi positive definite operator with $\mathrm{Tr}(\rho) = 1$ (where $\mathrm{Tr}(.)$ indicates the trace of .). Note that a given

pure state $|\psi\rangle$ can also be represented with the density matrix $|\psi\rangle\langle\psi|$.

The most general operation on quantum states are the transformations of density matrices i.e. linear operators on operators (*super-operator*). The physically allowed super-operators are linear completely positive and trace-preserving operators, called *CP maps* for short. A super-operator T is positive if it sends positive semi-definite Hermitian matrices to positive semi-definite Hermitian matrices; it is completely positive if $T \otimes I_d$ is positive, where I_d is the identity operator on a d-dimensional Hilbert space.

In what follows we reformulate the postulates of quantum mechanics in terms of density matrices.

Postulate 1. The predictions of results of measurements of an isolated system are probabilistic in nature. This probabilistic information is represented mathematically by a density operator, which is a positive operator ρ with trace one, acting on a complex Hilbert space $\mathcal{H}$ that forms the state space of the quantum theory. If a quantum system is in the state ρ_i with probability p_i, the denisty operator for the system is $\sum_i p_i \rho_i$.

Postulate 2. In a *closed* system, the evolution of the system is described by a *unitary transformation*. That is, the state ρ_1 of the system at time t_1 is related to the state ρ_2 at time t_2 by a unitary operator U which depends only on the time t_1 and t_2,

$$\rho_2 = U \rho_1 U^\dagger \,.$$

Postulate 3. Quantum measurements are described by a collection M_m of *measurements operators*. These are operators acting on the state space of the system being measured. The index m refers to the measurements outcome that may occur in the experiment. If the state of the quantum system is ρ immediately before the measurement then the probability that the result m occurs is given by

$$p(m) = \mathrm{Tr}(M_m^\dagger M_m \rho) \,,$$

and the state of the system after the measurement is

$$\frac{M_m \rho M_m^\dagger}{\mathrm{Tr}(M_m^\dagger M_m \rho)} .$$

The measurements operators satisfy the *completeness equation*,

$$\sum_m M_m^\dagger M_m = I .$$

Postulate 4. The state space of a composite physical system is the tensor product of the state spaces of the component physical systems. Moreover, if we have individual systems numbered 1 to n, and system i is prepared in the state ρ_i (independently from other systems), then the joint state of the total system is $\rho_1 \otimes \rho_2 \otimes \cdots \otimes \rho_n$.

1.2.2 Technical Developments

In this subsection we discuss some technical developments of the quantum rules presented in the previous subsection. First we briefly review the notation of *entangled states* and *LOCC maps* which will be the topic of Chapter 5. Then Gleason's Theorem for determining all the measures on a Hilbert space will be presented. This is required for our discussion on semantics of quantum computing in Chapter 4.

Entanglement is a uniquely quantum resource that plays a key role in most of the applications of quantum computation and information theory [67, 105].

Definition 10 *A pure state of a composite system that cannot be written as a product of states of its component systems is called an* entangled *state.*

We are also interested in the manipulation of entanglement, by which we mean: Given an entangled state of a composite system, what other entangled states can be prepared using arbitrary operations only on the local systems, including measurement, and classical communications between components of the system? In other word the class of transformations which are allowed to be performed are:

Definition 11 LOCC *(local operations and classical communication) consists of arbitrary quantum operations acting separately on individual parts of a composite*

system, assisted by classical communications between the individual parts.

In terms of understanding entanglement both in a phenomenological sense and as a resource, it would be useful to be able to measure the amount of entanglement for a given state. To this end, a measure of entanglement is required to order states according to the amount of entanglement they contain. This issue will be discussed in more detail in Chapter 5.

In the remaining part of this subsection we present the following important theorem by Gleason [50], which provides a correspondence between density operators and probability measures on measurable sets of the corresponding Hilbert space.

Theorem 12 *[50] Let μ be a probability measure on the closed subspaces of a separable Hilbert space $\mathcal{H}$ of dimension at least three. There exists a positive semi-definite self-adjoint operator T of the trace class (density matrix) such that for all closed subspaces A of $\mathcal{H}$*

$$\mu(A) = \mathrm{Tr}(TP_A)\,,$$

where P_A is the orthogonal projection of $\mathcal{H}$ onto A.

We omit the proof as it needs special treatment and it can be found in [50]. The following lemmas can be also proven in the same way.

Lemma 13 *Let μ be a positive measure on the closed subspaces of a separable Hilbert space $\mathcal{H}$ of dimension at least three. There exists a positive semi-definite self-adjoint operator T such that for all closed subspaces A of $\mathcal{H}$*

$$\mu(A) = \mathrm{Tr}(TP_A)\,,$$

where P_A is the orthogonal projection of $\mathcal{H}$ onto A.

Lemma 14 *Let μ be a measure on the closed subspaces of a separable Hilbert space $\mathcal{H}$ of dimension at least three. There exists a self-adjoint operator T such that for all closed subspaces A of $\mathcal{H}$*

$$\mu(A) = \mathrm{Tr}(TP_A)\,,$$

where P_A is the orthogonal projection of $\mathcal{H}$ onto A.

1.3 Quantum Computation

The bounds on encoding and the speed of information processing using quantum systems are different to those based on the laws of classical physics. Since classical laws can be consider as a special case of the more general quantum laws it is clear that a quantum computer will be at least as efficient as the classical computer. In other word a quantum computer can efficiently simulate any classical processing with the same computational costs on a classical computer. The exciting discovery was that quantum computer is in fact provably more efficient than any classical computer [9]. One of the key effects leading to this efficiency is the quantum superposition phenomenon which allows a quantum computer to perform a given tasks simultaneously (in parallel) on multiple data.

There are few distinct algorithms which show that a quantum computer can be more efficient than its classical counterpart. These include factoring of numbers [97], database search [53], solution to the Pell's equation [54, 69], computing orders for solvable groups [110] to name a few [29]. There are also a number of quantum communication protocols that can be viewed as elementary quantum computations, such as the cryptographic key exchange [19], quantum teleportation [13] and dense coding [11]. The clearest advantage of using quantum systems is seen in factorisation which is an NP problem on the classical computer [97], whereas on the quantum computer it can be performed in polynomial time [46]. Factorisation is also potentially of great importance for the field of cryptography. It is known that this algorithm is a special case of a general problem, the hidden sub-group problem (HSP) [68]. HSP has been studied recently and for the Abelian case the general solution is known [77]. The other key example for the quantum speed-up is Grover's database search [53], which can achieve a quadratic speed-up over its classical counter-part. Grover's search idea has been generalised to the amplitude amplification method which can be applied to speed up a number of other algorithms [47]. Search itself lies at the root of many other important difficult computational tasks so that this algorithm has a wide applicability. All these indicate that there is

an enormous potential in using quantum systems to encode and process information which is much more powerful than the present classical computers.

In this section we present the two models of quantum computation, quantum Turing machine and quantum circuit model. Subsequently we review the basic definitions of quantum complexity analysis.

1.3.1 Quantum Turing Machine

Here we give the formal definition of quantum Turing Machine; more details can be found in [32, 14, 80]. The quantum Turing machine was introduced by Benioff [8]. Afterward Deutsch described a universal simulator for QTMs with exponential overhead [32]. And finally Bernstein and Vazirani constructed a universal QTM with polynomial overhead [14].

A quantum Turing machine (QTM), M, consists of a processor, a two-way infinite tape and a head. We denote the set of processor configurations, a finite set of symbols, with Q and the set of finite alphabet with Σ. Σ always contains the special symbol $\natural$, the blank symbol. The sets of initial (I) and final (F) states are proper subsets of Q. Then the system configuration is represented by a triple $(q, S, n) \in Q \times \Sigma^{\omega} \times \mathbb{Z}$ where q is the current state, S the infinite string of the tape and n the head position. The quantum state of M is represented by a unit vector in $\mathcal{H}$, the Hilbert space spanned by vectors in $Q \times \Sigma^{\omega} \times \mathbb{Z}$. The transition function of M is a complex-valued function,

$$\delta : Q \times \Sigma \times Q \times \Sigma \times \{-1, 1\} \to \mathbb{C}\,.$$

The quantum Turing machine M defines a linear operator (the unitary time evolution):

$$U_M : \mathcal{H} \to \mathcal{H}\,,$$

such that

$$U_M|q, S, n\rangle = \Sigma_{p\in Q, s\in\Sigma, d\in\{-1,1\}}\delta(q, S(n), p, s, d)|p, S_n^s, n + d\rangle\,,$$

where

$$S_n^s(i) = \begin{cases} s & \text{if } i = n \\ S(i) & \text{if } i \neq n \end{cases}$$

A final configuration of a QTM is any configuration in state q_f. If when QTM M is run with input x, at time T the superposition contains only final configurations and at any time less than T the superposition contains no final configuration, then M *halts* with running time T on input x. The superposition of M at time T is called the final *final superposition*.

Definition 15 *A QTM is called* well-behaved *if it halts on all input strings in a final superposition where each configuration has the tape head in the same cell. If this cell is always the start cell, we call the machine* stationary. *A well-behaved QTM is in* normal form *if q_f always leads back to q_0.*

Despite its simple appearance, the Turing Machine can efficiently simulate arbitrary algorithms. The concept of *Languages* in Turing model is defined in the following way.

Definition 16 *We define $L \subset (\Sigma \setminus \{\flat\})^*$ to be a* Language, *i.e. a language is a set of strings of symbols. Let M be a Turing machine such that, for any string $x \in (\Sigma \setminus \{\flat\})^*$, if $x \in L$, then $M(x) =$"yes" (i.e. M on input x halts at the "yes" state), and if $x \notin L$, then $M(x) =$"no". Then we say that M* decides *L.*

The following notation of recursive language is required for our discussion on computability (Chapter 4).

Definition 17 *If a language L is decided by some Turing machine M, then L is called* recursive. *We say that M* accepts *L whenever, for any string $x \in (\Sigma \setminus \{\flat\})^*$, if $x \in L$ then $M(x) =$"yes"; however for $x \notin L$, then M does not halt. If L is accepted by some Turing machine M, then L is called* recursively enumerable.

It is clear that, if a language L is recursive, then it is also recursively enumerable.

We shall not only deal with the decision and acceptance languages, but also occasionally with the computation of string of functions.

Definition 18 *Suppose that f is a function from $(\Sigma \setminus \{\flat\})^*$ to Σ^*, and let M be a Turing machine with alphabet Σ. We say that M* computes *f if, for any string $x \in (\Sigma \setminus \{\flat\})^*$, $M(x) = f(x)$. If such an M exists, f is called a* recursive *function.*

In quantum computation, we will consider the probabilistic analogue of the above definitions.

Definition 19 *Let M be a stationary, normal forma, multi track QTM. We say that M accepts x with probability p and rejects x with probability $1 - p$, if when we run M with string x on the first track and empty string elsewhere, after M halts* [2] *we observe 1 with probability p on the last track of the start cell.*

We define two different settings for accepting a language L with a quantum Turing machine.

Definition 20 *We say that QTM M accepts L* exactly *if M accepts every string $x \in L$ with probability 1 and rejects every string $x \in (\Sigma \setminus \{\flat\})^* \setminus L$ with probability 1. In the* bounded error *setting, M accepts with probability at least p every string $x \in L$ and rejects with probability at least p every string $x \in (\Sigma \setminus \{\flat\})^* \setminus L$.*

1.3.2 Quantum Circuit Model

Here we discuss the quantum circuit model for quantum computation which will be the main framework for all the discussions in this thesis [33, 114]. In analogy with a classical bit, a two-state quantum system is called a *qubit* or a *quantum bit*. Mathematically, a qubit takes a value in the vector space $\mathbb{C}^2$. We single out two orthogonal basis vectors, $|0\rangle$ and $|1\rangle$, to denote the computational basis. A quantum circuit is built out of logical quantum wires carrying qubits, and quantum gates acting on these qubits.

Definition 21 *A quantum gate, U, of order k is a unitary linear map on k qubits. Its action on a state $|\psi\rangle$ is denoted as $U|\psi\rangle$.*

[2] This can be accomplished by performing a partial measurement to check whether the machine is in the final state.

The matrix representations of the quantum operations used in this thesis are:

$$
\begin{aligned}
\text{Hadamard} \quad H &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \\
\text{Pauli-X} \quad X &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \\
\text{Pauli-Y} \quad Y &= \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \\
\text{Pauli-Z} \quad Z &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \\
\text{Phase} \quad P &= \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \\
\text{Rotation-}\pi/8 \quad T &= \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}, \\
\text{controlled-Not} \quad CNOT &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \\
\text{swap} \quad S &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\end{aligned}
$$

A set of quantum gates is said to be *universal for quantum computation* if any unitary operation can be approximated to arbitrary accuracy by a quantum circuit involving only those gates. in the literature, there exists many examples of universal set of gates [79]:

- The Hadamard, Phase, CNOT and $\pi/8$ Rotation gates,
- Single qubit and CNOT gates.

In quantum circuit model, measurements can always be moved to the end of the circuit and this process is performed in the computational basis of one or more of the qubits of the circuit.

All the different settings of *exact, zero-error* and *two-sided bounded error* can be also considered for the computation of a function with a quantum circuit model.

In the remaining part of this subsection we present the quantum circuits model in the most general setting, with mixed state, which was introduced by Aharonov et al. in [4]. They also showed that this model is polynomially equivalent in computational power to the standard unitary quantum circuit model, introduced by Deutsch [33].

We start by definition of the building blocks of a network i.e. gates.

Definition 22 *A quantum gate, g, of order (k, l) is a trace preserving, completely positive, linear map from density matrices on k qubits to density matrices on l qubits. Its action on a density matrix ρ is denoted as $g \circ \rho$.*

The definition of a quantum network in the general setting of working with mixed states and CP maps is:

Definition 23 *Let $\mathcal{G}$ be a family of quantum gates. A quantum circuit that uses gates from $\mathcal{G}$ is a directed acyclic graph. Each node v in graph is labeled by a gate $g_v \in \mathcal{G}$ of order (k_v, l_v). The in-degree and out-degree of v are equal k_v and l_v, respectively. An arbitrary subset of the inputs are labeled blank. An arbitrary subset of the outputs are labeled result.*

The final definition describes the function computed by a quantum network:

Definition 24 *Let Q be a quantum circuit, with n inputs and r result outputs. The probabilistic function computed by Q, $f_Q : \{0,1\}^n \rightarrow [0,1]^{\{0,1\}^r}$ is defined as follows: For input i, the probability for getting the output j is*

$$f_{i,j} = \langle j|(Q \circ |i\rangle\langle i|)|_A|j\rangle \,,$$

where A is the set of the result outputs.

1.3.3 Complexity Analysis

Complexity theory studies the required cost of solving computational problems [84]. The cost is measured in terms of different well-defined resources e.g. elementary operations, memory usage, amount of communication. A *computational problem* can be thought of as a function whose input is a *program instance* and whose corresponding output is the *solution* to it. A *decision problem* deals with a question that requires either a "yes" or "no" answer and can be represented with a function $f : \{0,1\}^* \to \{0,1\}$. Decision problems are simple tools for developing a rigourous mathematical theory for complexity analysis and they are general as many other problems can be recast in terms of decision problems that are essentially equivalent to the original problem.

In complexity theory, it is common to use the following asymptotic notation.

Definition 25 *Assume f and g are functions from $\mathbb{N}$ to $\mathbb{N}$. We say f is* bounded above *with g, denoted by $f(n) = \mathcal{O}(g(n))$, iff*

$$\exists \text{ positive integers } c, n_0 : \ (\forall n \geq n_0 : f(n) \leq cg(n)) \, .$$

Also f is bounded below *with g, denoted by $f(n) = \Omega(g(n))$, iff $g(n) = \mathcal{O}(f(n))$. Finally $f(n) = \Theta(g(n))$ means that $f(n) = \mathcal{O}(g(n))$ and $f(n) = \Omega(g(n))$.*

There are different known frame-works for quantum complexity analysis: computational complexity, query complexity and communication complexity [14, 28]. In the first scenario the complexity involves the number of elementary gates that need to be applied to execute the problem, as well as the number of qubits used in the computation. In the query complexity we assume that in addition to elementary gates we are given a black-box performing a special computational task which we can query as many times as needed to solve the problem. The complexity is now the number of times we have to query the black box. In the final scenario we consider the number of qubits needed for communication between the two parties who wants to perform a computational tasks.

Despite the differences between these models, there are also some intimate relationships between them [28]. The query model is a simple model to compare the

computational power of quantum and classical computer (see below). Quantum algorithms in the query complexity model can also be transformed into protocols in communication complexity model and most of the currently known quantum algorithms evolved from algorithms in the query model.

In this thesis we mainly use query model (next section) for complexity analysis.

1.3.4 Quantum Query Model

One important way of comparing the efficiencies of quantum and classical algorithms is by analysing *query complexity*, which measures the number of invocations of an *oracle* — which may be a standard circuit (or a Turing machine) implementing a useful sub-routine, a physical device, or a purely theoretical construct — needed to complete a task.

In this thesis we mainly consider an oracle to be a given quantum circuit which efficiently implements a boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$. Equivalently, an oracle (black-box) contains an N-tuple ($N = 2^n$) of Boolean variables $X = (x_0, x_1, \cdots, x_{N-1})$. The box is equipped to output x_i on input i. The goal is to determine some property of X accessing the x_i only through the black box. Such a black-box access is called a *query* and assumes to have a unit cost of evaluation. A property of X is any Boolean function that depends on X. Assume $N = 2^n$, a property can be represented with a function of the following type:

$$F : \{0,1\}^N \rightarrow \{0,1\} .$$

As mentioned before we can consider different settings for computing F on $\{0,1\}^N$ in the query model. The minimum number of queries required by a quantum circuit to compute F in the exact, zero-error, and bounded-error settings, is denoted by $Q_E(F), Q_0(F)$ and $Q_2(F)$, respectively.

A number of general results show the limitations and advantages of quantum computers using the query complexity models [34, 15, 9, 7, 103, 22, 28]. It is clear that upper bounds in the query model implies upper bounds for computational complexity, i.e. for the circuit description model in which the function X is suc-

cinctly described as a $(\log N)^{\mathcal{O}(1)}$-sized circuit computing x_i from i. On the other hand, lower bounds in the black-box model do not imply lower bounds in the circuit model, though they can provide useful guidance, indicating what certain algorithmic approaches are capable of accomplishing. In [7], some general lower bounds for query complexity of computing an arbitrary Boolean function F are given. In Chapter 3 we discuss the quantum oracles in more detail.

Complexity Classes

A *complexity class* is a set of languages representing a set of decision problems. All the languages in a complexity class can be decided within some bound on some aspect of their performance [84]. In what follows we give the definitions of standard complexity classes that we will refer to, in this thesis.

- **P**. The class of decision problems that can be solved in polynomial time by deterministic Turing machines.
- **NP**. The class of decision problems that can be solved in polynomial time by nondeterministic Turing machines.
- **PSPACE**. The class of decision problems that can be solved in polynomial space by deterministic Turing machines.
- **BPP**. The class of decision problems that can be solved in polynomial time by probabilistic Turing machines with error probability bounded $1/3$ (for all inputs).
- **BQP**. The class of decision problems that can be solved in polynomial time by quantum Turing machines with error probability bounded $1/3$ (for all inputs).

We presented in this chapter all the basic definitions and structures which are required for the rest of our discussion throughout this thesis. In the first part, we present complexity analysis of different scenarios in quantum computation framework. We work mainly within the quantum query model, which offers an elegant way of putting bounds on the efficiency of quantum algorithms. Furthermore we

consider the notions of states and operators complexity as a key way to find the relationship between physical complexity and computational complexity.

In the second part of this thesis we study semantics of quantum computation. Semantics studies the meaning of programs, mainly in order to be able to state correctness properties of the instructions within them. Domain theory has proven to be a proper mathematical structure to describe denotational semantics for programming languages. We extend this structure to the quantum setting and derive a denotational semantics for quantum computing.

2

Quantum One-way Function

2.1 Introduction

The existence of one-way functions is one of the most important open problems in classical computation. It is also well-known that one-way functions have applications in cryptography [84]. Loosely speaking, a one-way function is one that is easy to compute but hard to invert (the precise definition of one-way function will be given later). The existence of one-way functions is linked to the complexity class **UP**, the class of languages accepted by a special, called *unambiguous*, polynomial time bounded nondeterministic Turing machines and the following relationship is well-known, $\mathbf{P} \subseteq \mathbf{UP} \subseteq \mathbf{NP}$ [84]. Furthermore the existence of one-way functions is equivalent to the separation between the complexity classes **P** and **UP** [52], and hence **P** and **NP** which indicates the difficulty of the problem of the existence of one-way functions.

In this chapter we consider the quantum one-way permutations which is a restricted class of quantum one-way functions. We prove a necessary and sufficient condition for inverting efficiently a polynomial time computable permutation [72, 73]. In the classical case, Hemaspaandra and Rothe [59] presented a necessary and sufficient condition for the existence of one-way permutations. We show that in the quantum setting, the problem of inverting a permutation in polynomial time is

equivalent to the problem of constructing polynomial size quantum networks for the reflection about a class of quantum states that we will define later. In the proof of this equivalence, we present a quantum algorithm for inverting a permutation efficiently under the condition that reflections about their quantum states are efficiently implementable. Furthermore, we consider the relationship between the complexity of preparing a state and the reflection about that state.

Through out this chapter we will refer to the search and invert problems.

Problem 1 *For a given boolean function on n-bit strings, $f : \{0,1\}^n \to \{0,1\}$, let U_f denote the unitary operator mapping the basis state $|x\rangle|y\rangle$ to $|x\rangle|f(x) \oplus y\rangle$, where $|x\rangle$ consist of n qubits and $|y\rangle$ is a single qubit. Given U_f as an oracle, the goal is to find $x_0 = f^{-1}(1)$. We assume that there exists a unique x_0. This problem is called* SEARCH.

Grover's algorithm [53] for SEARCH consists of the following steps.

ALGORITHM **A**

Step 1 (Preparation).

Prepare the uniform superposition

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \,.$$

Step 2 (Iteration).

Iterate Step 2.1 and Step 2.2.

Step 2.1. Perform the tagging operator given by

$$I - 2|f^{-1}(1)\rangle\langle f^{-1}(1)| \,.$$

Step 2.2. Perform the reflection operator about the state $|\psi\rangle$ given by

$$I - 2|\psi\rangle\langle\psi| \,.$$

The state in Step 1 is prepared by performing n Hadamard gates on n qubits with initial state $|0\rangle$:

$$H^{\otimes n}|0\cdots 0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle\,.$$

Step 2.1 is implemented by querying the oracle U_f twice:

$$\{(I-2|f^{-1}(1)\rangle\langle f^{-1}(1)|)\otimes I\}|x\rangle|0\rangle = \{U_f(I\otimes(I-2|1\rangle\langle 1|))U_f\}|x\rangle|0\rangle.$$

And finally Step 2.2 is implemented using n Hadamard gates and CNOT gates:

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

Problem 2 *For a given one-to-one function on n-bit strings, $g:\{0,1\}^n\to\{0,1\}^n$, let U_g denote the unitary operator mapping the basis state $|x\rangle|y\rangle$ to $|x\rangle|g(x)\oplus y\rangle$, where $|x\rangle$ and $|y\rangle$ each consist of n qubits and $\oplus$ is addition modulo 2^n. Given U_g as an oracle, the goal is to find $x_0 = g^{-1}(y)$ for any given $y\in\{0,1\}^n$. This problem is called* INVERT.

An algorithm for INVERT (Algorithm **B** below) is as follows [18].

ALGORITHM **B**

Step 1 (Preparation).

Prepare the uniform superposition

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} |y\rangle.$$

Step 2 (Iteration).

Iterate Step 2.1 and Step 2.2.

Step 2.1 Perform the tagging operator given by

$$I-2|g^{-1}(y)\rangle\langle g^{-1}(y)|\,.$$

Step 2.2. Perform the reflection operator about the state $|\psi\rangle$ given by

$$I - 2|\psi\rangle\langle\psi| \,.$$

Step 1 and Step 2.2 are implemented similar to Algorithm **A** and Step 2.1 is implemented using two queries to the oracle U_g:

$$\{(I - 2|g^{-1}(y)\rangle\langle g^{-1}(y)|) \otimes I\}|y\rangle|0\rangle = U_g(I \otimes (I - 2|y\rangle\langle y|))U_g|y\rangle|0\rangle \,. \quad (2.1)$$

2.2 Worst Case Complexity

In this section we consider "one-wayness" in the worst case complexity, i.e. the highest computational cost among all the possible inputs. The following definitions give the precise description of quantum one-way permutation in the worst case scenario. We consider permutation functions in the following setting.

Definition 26 *A function $f : \{0,1\}^* \to \{0,1\}^*$ is called a* permutation *if it satisfies the following conditions*

(i) f is one-to-one and length preserving.

(ii) For some strictly increasing function $a : \mathbb{N} \to \mathbb{N}$ we have:

$$\mathrm{Dom}(f) = \bigcup_{n \in \mathbf{N}} \{0,1\}^{a(n)} \,.$$

These conditions imply that the restriction of f to $\{0,1\}^n \subseteq \mathrm{Dom}(f)$ is a permutation on $\{0,1\}^n$. The definition of one-way function in the worst case complexity is as follows.

Definition 27 *A function f is a* worst case quantum one-way function*, if the following conditions are satisfied:*

(i) f is one-to-one, and for all $x \in \{0,1\}^*, |x|^{\frac{1}{k}} \leq |f(x)| \leq |x|^k$ [1] for some $k > 0$. That is, $f(x)$ is at most polynomially longer or shorter than x.

[1] Here $|x|$ denotes the length of the string x.

(ii) f *can be computed by a uniform polynomial size classical network.*

(iii) f^{-1} *cannot be computed by any polynomial size quantum network.*

Note that condition (i) is naturally satisfied for one-way permutations.

As we saw in the introduction Algorithm **B** for INVERT uses the tagging operator O (defined below) which can be simulated by two applications of U_f and n controlled-not gates (Equation 2.1).

$$O|x\rangle|y\rangle = \begin{cases} -|x\rangle|y\rangle & \text{if } f(y) = x \\ |x\rangle|y\rangle & \text{if } f(y) \neq x \end{cases} \tag{2.2}$$

Moreover, if f is polynomial time computable, then it is also possible to efficiently construct the unitary operator $O[k]$ defined by

$$O[k]|x\rangle|y\rangle = \begin{cases} -|x\rangle|y\rangle & \text{if } f(y)_{(k,k+1)} = x_{(k,k+1)} \\ |x\rangle|y\rangle & \text{if } f(y)_{(k,k+1)} \neq x_{(k,k+1)} \end{cases}$$

where $s_{(i,j)}$ denotes the bit string from i-th bit to j-th bit of the bit string s. The operators $O[k]$'s will enable us to mark all the states $|y\rangle$ such that 2 qubits of $|f(y)\rangle$ are equal to the corresponding qubits of $|x\rangle$. Geometrically, $O[k]$ can be considered to be the reflection about the hyper-plane spanned by the vectors $\{|y\rangle : \ f(y)_{(k,k+1)} \neq x_{(k,k+1)}\}$. We will show that if we can efficiently implement $O[k]$'s and the set of unitary operators

$$Q_j = \sum_{x \in \{0,1\}^n} |x\rangle\langle x| \otimes (2|\psi_{j,x}\rangle\langle\psi_{j,x}| - I)\,,$$

where

$$|\psi_{j,x}\rangle = \frac{1}{\sqrt{2^{n-2j}}} \sum_{y : f(y)_{(1,2j)} = x_{(1,2j)}} |y\rangle\,,$$

then we can efficiently invert f by a polynomial size network. Conversely, we will also prove that if f is difficult to invert, then Q_j's are also difficult to construct. Now we state and prove this result formally. We say that a set F of unitary operators is

easy if every $U \in F$ is easy i.e. it can be implemented with a quantum polynomial size network. The precise definition of easy operator is given in Section 2.5.

Theorem 28 *Suppose f satisfies condition (i) and (ii) of definition 27. Then f is a worst case quantum one-way permutation if and only if the set $F_n = \{Q_j\}_{j=0,1,\ldots,\frac{n}{2}-1}$ of unitary operators is not easy.*

Proof Without loss of generality, we can assume that n is even.

($\Rightarrow$) Suppose that F_n is easy. Then we show that f^{-1} is computable by a polynomial size quantum network. A quantum algorithm computing f^{-1} is as follows (Algorithm **C** below). Assume that x is given as the input in the first register of the quantum network to be constructed.

ALGORITHM **C**

Step 1 (Preparation).

Prepare the second register in the uniform superposition

$$|\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} |y\rangle.$$

Step 2 (Iteration).

For $j = 0$ to $\frac{n}{2} - 1$, implement the following steps $2.j.1$–$2.j.2$.

Step $2.j.1$ Perform the operator $O[2j+1]$ on the first and the second registers.

Step $2.j.2$ Perform the operator Q_j on the first and the second registers.

Step $2.j.1$ can be implemented through the following three steps: (1) Perform the operator $U_f : |y\rangle|z\rangle \mapsto |y\rangle|f(y) \oplus z\rangle$ on the second and third registers. (2) Compare the $2j+1$-th and the $2j+2$-th qubits of the first register with the corresponding qubits of the third register, and apply a phase shift of -1 if they are same; otherwise do nothing. (3) Perform the operator U_f on the second and third registers (Figure 2.1).

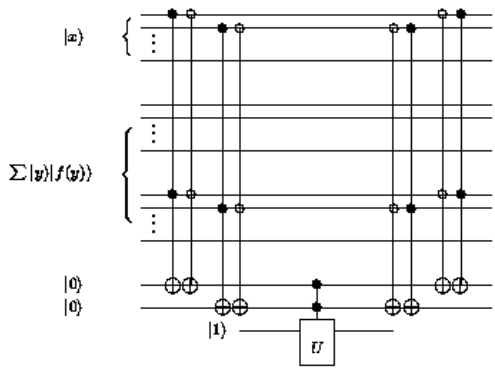

Figure 2.1: A quantum circuit for tagging operator.

Now we show that Algorithm **C** computes f^{-1}. After Step 1, the state of the system is

$$\frac{1}{\sqrt{2^n}}|x\rangle \sum_{y\in\{0,1\}^n} |y\rangle.$$

We show that after Step $2.j.2$ the state of the system is

$$\frac{2^{j+1}}{\sqrt{2^n}}|x\rangle \sum_{y:f(y)_{(1,2j+2)}=x_{(1,2j+2)}} |y\rangle\,,$$

which means that Algorithm **C** computes f^{-1} after $\frac{n}{2}$ iterations. In the case $j=0$, the state evolves as follows (note that for any x we have $|\psi_{0,x}\rangle = |\psi_0\rangle$):

$$\begin{aligned}
&\frac{1}{\sqrt{2^n}}|x\rangle \sum_{y\in\{0,1\}^n} |y\rangle \\
\xrightarrow{2.0.1}\quad &\frac{1}{\sqrt{2^n}}|x\rangle \left(\sum_{y:f(y)_{(1,2)}\neq x_{(1,2)}} |y\rangle - \sum_{y:f(y)_{(1,2)}=x_{(1,2)}} |y\rangle \right) \\
&= \frac{1}{\sqrt{2^n}}|x\rangle \left(\sqrt{2^n}|\psi_0\rangle - 2 \sum_{y:f(y)_{(1,2)}=x_{(1,2)}} |y\rangle \right)
\end{aligned}$$

$$
\begin{aligned}
\xrightarrow{2.0.2} \quad & \frac{1}{\sqrt{2^n}}|x\rangle(2|\psi_0\rangle\langle\psi_0| - I)\left(\sqrt{2^n}|\psi_0\rangle - 2\sum_{y:f(y)_{(1,2)}=x_{(1,2)}}|y\rangle\right) \\
& = \frac{1}{\sqrt{2^n}}|x\rangle\left(2\sqrt{2^n}|\psi_0\rangle - \sqrt{2^n}|\psi_0\rangle - 4|\psi_0\rangle\sum_{y:f(y)_{(1,2)}=x_{(1,2)}}\langle\psi_0|y\rangle\right) \\
& +2\sum_{y:f(y)_{(1,2)}=x_{(1,2)}}|y\rangle \\
& = \frac{2}{\sqrt{2^n}}|x\rangle\sum_{y:f(y)_{(1,2)}=x_{(1,2)}}|y\rangle.
\end{aligned}
$$

On the other hand, suppose that the case $j = k-1$ holds. Then, following Steps $2.k.1$–$2.k.2$, the state evolves as follows:

$$
\begin{aligned}
& \frac{2^k}{\sqrt{2^n}}|x\rangle\sum_{y:f(y)_{(1,2k)}=x_{(1,2k)}}|y\rangle \\
\xrightarrow{2.k.1} \quad & \frac{2^k}{\sqrt{2^n}}|x\rangle\left(\sum_{y:f(y)_{(1,2k)}=x_{(1,2k)}}|y\rangle - \sum_{y:f(y)_{(1,2k+2)}=x_{(1,2k+2)}}|y\rangle\right) \\
& = \frac{2^k}{\sqrt{2^n}}|x\rangle\left(\sqrt{2^{n-2k}}|\psi_{k,x}\rangle - 2\sum_{y:f(y)_{(1,2k+2)}=x_{(1,2k+2)}}|y\rangle\right) \\
\xrightarrow{2.k.2} \quad & \frac{2^k}{\sqrt{2^n}}|x\rangle(2|\psi_{k,x}\rangle\langle\psi_{k,x}| - I)\left(\sqrt{2^{n-2k}}|\psi_{k,x}\rangle - 2\sum_{y:f(y)_{(1,2k+2)}=x_{(1,2k+2)}}|y\rangle\right) \\
& = \frac{2^k}{\sqrt{2^n}}|x\rangle \\
& \left(2\sqrt{2^{n-2k}}|\psi_{k,x}\rangle - \sqrt{2^{n-2k}}|\psi_{k,x}\rangle - 4|\psi_{k,x}\rangle\sum_{y:f(y)_{(1,2k+2)}=x_{(1,2k+2)}}\langle\psi_{k,x}|y\rangle\right) \\
& +\frac{2^k}{\sqrt{2^n}}|x\rangle\left(2\sum_{y:f(y)_{(1,2k+2)}=x_{(1,2k+2)}}|y\rangle\right) \\
& = \frac{2^{k+1}}{\sqrt{2^n}}|x\rangle\sum_{y:f(y)_{(1,2k+2)}=x_{(1,2k+2)}}|y\rangle\,.
\end{aligned}
$$

Thus, the case $j = k$ holds. From the assumption that $\{Q_j\}$ is easy, it is simple to see that Algorithm **B** can be implemented by a polynomial size quantum network.

($\Leftarrow$) Suppose that f is not a worst-case one-way permutation. Then we show that $\{Q_j\}_{j=0,1,\ldots,\frac{n}{2}-1}$ can be implemented by a polynomial size quantum network.

According to the assumption, f and f^{-1} are quantum polynomial time computable. The following operator

$$M_f : |x\rangle \mapsto |f(x)\rangle$$

can be implemented by a polynomial size quantum network [12, 71] (Chapter 3). To see why note that, for any $x \in \{0,1\}^n$ we have

$$[M_f \otimes I]|x\rangle|0\rangle = [(U_{f^{-1}})^{-1} S U_f]|x\rangle|0\rangle\,,$$

where the swap gate S is defined as $S : |a\rangle \otimes |b\rangle \mapsto |b\rangle \otimes |a\rangle$.

In the following we show that the unitary operator $Q'_j = (I \otimes M_f) Q_j (I \otimes M_f)^\dagger$ can be implemented by a polynomial size quantum network, which means that Q_j can also be implemented by a polynomial size quantum network. The operator Q'_j can be rewritten as follows:

$$\begin{aligned}
Q'_j &= (I \otimes M_f) \left\{ \sum_{x\in\{0,1\}^n} |x\rangle\langle x| \otimes \left(2 \left(\frac{1}{2^{n-2j}} \sum_{y,y'}{}^{*} |y\rangle\langle y'| \right) - I \right) \right\} (I \otimes M_f)^\dagger \\
&= \sum_{x\in\{0,1\}^n} |x\rangle\langle x| \otimes \left(2 \frac{1}{2^{n-2j}} \sum_{y,y'}{}^{*} |f(y)\rangle\langle f(y')| - I \right) \\
&= \sum_{x\in\{0,1\}^n} |x\rangle\langle x| \otimes \\
&\quad \left(2|x_{(1,2j)}\rangle\langle x_{(1,2j)}| \frac{1}{2^{n-2j}} \sum_{y,y'}{}^{*} |f(y)_{(2j+1,n)}\rangle\langle f(y')_{(2j+1,n)}| - I \right) \\
&= \sum_{x\in\{0,1\}^n} |x\rangle\langle x| \otimes \left(2|x_{(1,2j)}\rangle\langle x_{(1,2j)}| \otimes |\psi_j\rangle\langle\psi_j| - I \right) \\
&= \sum_{x\in\{0,1\}^n} |x\rangle\langle x| \otimes \\
&\quad \left(|x_{(1,2j)}\rangle\langle x_{(1,2j)}| \otimes (2|\psi_j\rangle\langle\psi_j| - I) + \sum_{y: y \neq x_{(1,2j)}} |y\rangle\langle y| \otimes I \right).
\end{aligned}$$

Here, $\sum^*_{y,y'}$ denotes $\sum_{y,y': f(y)_{(1,2j)} = f(y')_{(1,2j)} = x_{(1,2j)}}$ and $|\psi_j\rangle$ denotes

$$|\psi_j\rangle = \frac{1}{\sqrt{2^{n-2j}}} \sum_{i\in\{0,1\}^{n-2j}} |i\rangle\,.$$

Thus, we can implement Q'_j by comparing the first $2j$ qubits of the first register with the corresponding qubits of the second register and applying $2|\psi_j\rangle\langle\psi_j| - I$ if they are the same and applying the identity otherwise (i.e. conditional-$(2|\psi_j\rangle\langle\psi_j| - I)$). The operator $2|\psi_j\rangle\langle\psi_j| - I$ is easy, since $2|\psi_j\rangle\langle\psi_j| - I = H^{\otimes n-2j}(2|0\rangle\langle 0| - I)H^{\otimes n-2j}$, where H is the Hadamard gate and the superscript $n-2j$ indicates that the Hadamard gate is applied to the last $n-2j$ qubits. Therefore, Q'_j is easy and this completes the proof. □

Note that all unitary operators U_k are easy if and only if the operation

$$\sum_k |k\rangle\langle k| \otimes U_k \,,$$

which implements U_k conditionally, is easy. The operator Q_j implements the reflection about the state $|\psi_{j,x}\rangle$ conditionally, therefore Theorem 28 gives a necessary and sufficient condition for quantum one-way permutations in terms of the reflection about a quantum state.

Using quantum amplitude amplification method [47] we can generalise the definition of operators $O[k]$ and Q_j in the Algorithm **C** as follows. In each step of Algorithm **C** we are concerned with only 2 qubits of input, i.e. the tagging operator $O[k]$ works only with the kth and $(k+1)$th qubits of its input register. However, one can consider the more general operators $O[k,l]$ as follows

$$O[k,l]|x\rangle|y\rangle = \begin{cases} -|x\rangle|y\rangle & \text{if } f(y)_{(k,k+l-1)} = x_{(k,k+l-1)} \\ |x\rangle|y\rangle & \text{if } f(y)_{(k,k+l-1)} \neq x_{(k,k+l-1)}, \end{cases}$$

where l is any integer satisfying $2 \leq l \leq O(\log(n))$. The corresponding reflection operators $Q_{j,l}$ are

$$Q_{j,l} = \sum_{x\in\{0,1\}^n} |x\rangle\langle x| \otimes (2|\psi_{j,l,x}\rangle\langle\psi_{j,l,x}| - I) \,,$$

where

$$|\psi_{j,l,x}\rangle = \frac{1}{\sqrt{2^{n-lj}}} \sum_{y:f(y)_{(1,lj)}=x_{(1,lj)}} |y\rangle \,.$$

Now the generalised Algorithm $\mathbf{C}'$ has the same structure as Algorithm **C**, but in Algorithm $\mathbf{C}'$ steps $2.j.1$ and $2.j.2$ will be iterated $T_l = O(\sqrt{2^l})$ times, where the integer T_l is known in advance. Note that T_l is a polynomial in n. Intuitively, Step 2 of Algorithm **C** is an analogue of Grover's algorithm for the search problem where the number of the required items is $\frac{1}{4}$ of the total number of items. On the other hand, Step 2 of Algorithm $\mathbf{C}'$ is also an analogue of Grover's algorithm for the search problem where the number of required items is $\frac{1}{2^l}$ of the total number of items. After applying steps $2.j.1$ and $2.j.2$ (for $j = k$) of Algorithm $\mathbf{C}'$, we obtain the state

$$|x\rangle \left(\sum_{y \in S_{k+1}} A_l |y\rangle + \sum_{y \in S_{k+1} \setminus S_k} B_l |y\rangle \right),$$

where $S_k = \{y : f(y)_{(1,lk)} = x_{(1,lk)}\}$ and positive numbers A_l and B_l are known in advance. Thus, using the quantum amplitude amplification process [47], we obtain the desired state:

$$\frac{1}{\sqrt{2^{n-l(k+1)}}} |x\rangle \sum_{y:f(y)_{(1,l(k+1))}=x_{(1,l(k+1))}} |y\rangle$$

and hence we can proceed to the next step.

2.3 Average Case Complexity

In order to apply our result to a realistic cryptographic scenario we need to consider also the average case complexity domain. This is because a realistic cryptographic protocol should be secure in "most" cases, which implies that it is hard to break on the average. We define two types of one-wayness in the average case setting. In what follows, for a property P defined on $\mathbf{N}$, we say that $P(n)$ holds for all

sufficiently large n if the set $\{n \in \mathbf{N} \mid P(n)$ does not hold$\}$ is finite.

Definition 29 *A permutation f is a weakly quantum one-way, if the following conditions are satisfied:*

(i) *f can be computed by a polynomial size network.*

(ii) *There exists a polynomial p such that for any polynomial size quantum network A and all sufficiently large $n \in \mathbf{N}$,*

$$\frac{1}{2^n} \sum_{x \in \{0,1\}^n} \mathrm{Prob}[A(f(x)) \neq x] > \frac{1}{p(n)} \,,$$

where Prob: $\{0,1\}^n \to [0,1]$ is a probability distribution induced by the measurement in the standard basis on the output register of the network A given the input x, and $A(x)$ is a random variable distributed with the function Prob.

In other words, a weakly quantum one-way permutation is easy to compute but the probability that any quantum algorithm fails to invert it is not negligible.

Definition 30 *A permutation f is a strongly quantum one-way, if the following conditions are satisfied*

(i) *f can be computed by a polynomial size network.*

(ii) *For any quantum polynomial size network A, any polynomial p, all sufficiently large n,*

$$\frac{1}{2^n} \sum_{x \in \{0,1\}^n} \mathrm{Prob}[A(f(x)) = x] < \frac{1}{p(n)},$$

where $A(x)$ is a random variable given as the output of the quantum algorithm A with the input x.

Again, in simple terms, a strongly quantum one-way permutation is easy to compute but the probability that any quantum algorithm succeeds in inverting it is negligible.

From the above definitions, it is easy to check the following relations.

Proposition 31 *In general we have*

(i) Every strongly quantum one-way permutation is also a weakly quantum one-way permutation.

(ii) Every weakly quantum one-way permutation is also a worst case quantum one-way permutation.

In the applications to cryptography, the existence of strongly quantum one-way permutations is the main concern. However, the following proposition shows that it is sufficient to characterise the existence of weakly quantum one-way permutations. We omit the proof as it is the same as the proof of Theorem 2.8 in [51].

Proposition 32 *Weakly quantum one-way permutations exist if and only if strongly quantum one-way permutations exist.*

For the rest of this section we discuss the relationships between weakly quantum one-way permutations and reflection operators, as we did in the worst case setting. We give a weaker analogue of Theorem 28 in the average case and finish the section with an open conjecture regarding the characterisation of weakly quantum one-way permutations. In order to carry out our discussion in the average case setting we need to introduce an approximation of the identity operator as follows:

Definition 33 *Let d: $\mathbf{N} \to \mathbf{N}$ be a function satisfying $d(n) \geq n$. A $d(n)$ qubit unitary operator J_n is called $(a(n), b(n))$-pseudo identity, if there exists a set X_n with $|X_n|/2^n \leq b(n)$ such that for $i \in \{0,1\}^n \setminus X_n$,*

$$|1 - (\langle i|_1 \langle 0|_2) J_n (|i\rangle_1 |0\rangle_2)| \leq a(n)\,,$$

where $|\cdot\rangle_1$ and $|\cdot\rangle_2$ denote the first n qubit state and the last $d(n) - n$ qubit state.

In what follows, I_j denotes the j-qubit identity operator, and $|\psi\rangle_{i_1 \cdots i_l}$ means that the system consists of the registers $i_1, \ldots, i_l$ and its state is $|\psi\rangle$. For a vector v, we denote the length of v by $|v|$. Now we can give the first result on the link between average case one-wayness and the reflections about quantum states.

Theorem 34 *Let f: $\{0,1\}^* \to \{0,1\}^*$ be a permutation that can be computed by a classical polynomial size network. If f is not weakly quantum one-way, then for any polynomial p and infinitely many n, there exist a polynomial r_p and $r_p(n)$-qubit $(1/2^{p(n)}, 1/p(n))$-pseudo identity operators $J_{p(n)}$ such that the family*

$$F_{p,n} = \{(I_n \otimes J_{p(n)})^\dagger (Q_j \otimes I_{r_p(n)-n})(I_n \otimes J_{p(n)})\}_{j=0,1,\ldots,\frac{n}{2}-1}$$

is easy, where Q_j is the same reflection operator defined in Section 3.

Proof Assume that f is not weakly quantum one-way. Then, for any polynomial p, there exist a polynomial size quantum network A and infinitely many n such that

$$\frac{1}{2^n} \sum_{y \in \{0,1\}^n} \mathrm{Prob}[A(y) = f^{-1}(y)] > 1 - \frac{1}{p(n)}\,. \tag{2.3}$$

Let $X'_n = \{y \in \{0,1\}^n | \ \mathrm{Prob}[A(y) = f^{-1}(y)] \leq \frac{1}{2}\}$ and $Y'_n = \{0,1\}^n \setminus X'_n$. From Equation (2.3) we have

$$\frac{1}{2^n}\left(|Y'_n| \cdot 1 + |X'_n| \cdot \frac{1}{2}\right) > 1 - \frac{1}{p(n)}\,,$$

and hence we obtain $|X'_n| < \frac{2}{p(n)} 2^n$. Define $q(n) = \frac{1}{4}p(n)$, then $|Y'_n| \geq (1 - \frac{1}{2q(n)})2^n$.

Now assume $y \in Y'_n$. The final state of the network A for input y is:

$$\alpha_y |y\rangle_1 |f^{-1}(y)\rangle_2 |\psi_y^a\rangle_3 + |y\rangle_1 |w(y)\rangle_2 |\phi_y^a\rangle_3\,, \tag{2.4}$$

where $\alpha_y \in \mathbf{R}$, $|1 - \alpha_y| \leq \frac{1}{2}$, $|f^{-1}(y)\rangle_2 \perp |w(y)\rangle_2$, and $||\psi_y^a\rangle_3| = ||w(y)\rangle_2| = 1$ (note that $|\phi_y^a\rangle_3$ is not a unit vector). By repeating the network A at most $O(q(n))$ times, we can easily construct a polynomial size quantum network B whose final state has the same form as Equation (2.4), where now $|1 - \alpha_y| \leq \frac{1}{2^{q(n)+1}}$. Denote by C the quantum network constructed from B by the approximate clean garbage method [9] as follows: (1) Apply B, (2) copy the contents of the second register (which is the output register of B) to an extra register, (3) apply the inverse of B and change the contents of the second and the extra registers. Then, we can see that

the final state of C on y is written in the following form:

$$\beta_y|y\rangle_1|f^{-1}(y)\rangle_2|0\rangle_3 + |\phi_y^b\rangle_{123},$$

where $\beta_y \in \mathbf{R}$, $|1-\beta_y| \leq \frac{1}{2^{q(n)}}$ and $|y\rangle_1|f^{-1}(y)\rangle_2|0\rangle_3 \perp |\phi_y^b\rangle_{123}$.

To establish the analogue result of Theorem 28 we define the following two approximation operators. First, the approximation of the operator M_f from Theorem 28 for the average case is defined as follows

$$\tilde{M}_f = (U_C)^{-1}(S \otimes I)(U_f \otimes I)\,, \tag{2.5}$$

where S denotes the swap operator on the first and the second registers and U_C is a unitary operator corresponding to the network C. The operator $\tilde{M}_f$ can be written in more detail as follows:

$$\begin{aligned}\tilde{M}_f &= \sum_{x\in Y_n}(\beta_{f(x)}|f(x)\rangle_1|0\rangle_{23} + |\phi_x^c\rangle_{123})\langle x|_1\langle 0|_{23} \\ &\quad + \sum_{x\in X_n}|\psi_x^c\rangle_{123}\langle x|_1\langle 0|_{23} + \sum_x\sum_{z:z\neq 0}|\psi_{x,z}^c\rangle_{123}\langle x|_1\langle z|_{23},\end{aligned}$$

where $|1-\beta_{f(x)}| \leq \frac{1}{2^{q(n)}}$ for any $x \in Y_n = \{x \in Y_n' | f(x) \in Y_n'\}$, $|f(x)\rangle_1|0\rangle_{23} \perp |\phi_x^c\rangle_{123}$, $X_n = \{0,1\}^n \setminus Y_n$, and $||\psi_x^c\rangle_{123}| = ||\psi_{x,z}^c\rangle_{123}| = 1$. The above form can be obtained by replacing the following forms of the operators $(U_C)^{-1}$ and $(S\otimes I)(U_f\otimes I)$ in the Equation (2.5):

$$\begin{aligned}(U_C)^{-1} &= \sum_{y\in Y_n'}|y,0,0\rangle_{123}(\beta_y\langle y, f^{-1}(y), 0|_{123} + \langle\phi_y^b|_{123}) \\ &\quad + \sum_{y\in X_n'}|y,0,0\rangle_{123}\langle y,0,0|_{123}U_C^{-1} \\ &\quad + \sum_y\sum_{(z,z')\neq(0,0)}|y,z,z'\rangle_{123}\langle y,z,z'|_{123}U_C^{-1}\end{aligned}$$

and

$$(S\otimes I)(U_f\otimes I) = \sum_{x\in Y_n'}|f(x),x,0\rangle_{123}\langle x,0,0|_{123}$$

$$+ \sum_{x \in X'_n} |f(x), x, 0\rangle_{123}\langle x, 0, 0|_{123}$$
$$+ \sum_{x} \sum_{(z,z') \neq (0,0)} |f(x) \oplus z, x, z'\rangle_{123}\langle x, z, z'|_{123}\,.$$

Next, the approximation of the reflection operators Q_j's from Theorem 28 is defined as follows

$$\begin{aligned} \tilde{Q}_j &= (I \otimes \tilde{M}_f)^\dagger (Q'_j \otimes I)(I \otimes \tilde{M}_f) \\ &= (I \otimes M_f^{-1}\tilde{M}_f)^\dagger (Q_j \otimes I)(I \otimes M_f^{-1}\tilde{M}_f), \end{aligned}$$

where Q'_j is the same unitary operator defined in the proof of Theorem 28. The family $\{\tilde{Q}_j\}_j$ satisfies the required conditions of Theorem 34. First, $\tilde{Q}_j$ is easy, since Q'_j, M_f and $\tilde{M}_f$ can be implemented by polynomial size quantum networks. Next, we check that $M_f^{-1}\tilde{M}_f$ is $(1/2^{q(n)}, 1/q(n))$-pseudo identity. Indeed, from $|Y'_n| \geq (1 - 1/2q(n))2^n$ and $|X'_n| \leq (1/2q(n))2^n$, we have that

$$\begin{aligned} |Y_n| &= |Y'_n| - |\{x \in Y_n | f(x) \in X_n\}| \\ &\geq (1 - \frac{1}{2q(n)})2^n - |X'_n| \\ &\geq (1 - \frac{1}{q(n)})2^n \end{aligned}$$

and hence $|X_n| \leq (\frac{1}{q(n)})2^n$. Thus, it is sufficient to check that for $x \in Y_n$ we have

$$|1 - (\langle x|_1\langle 0|_{23})M_f^{-1}\tilde{M}_f(|x\rangle_1|0\rangle_{23})| \leq \frac{1}{2^{q(n)}}. \tag{2.6}$$

This relation can be checked as follows. For $x \in Y_n$ we have

$$\tilde{M}_f|x\rangle_1|0\rangle_{23} = \sum_{x \in Y_n} (\beta_{f(x)}|f(x)\rangle_1|0\rangle_{23} + |\phi_x^c\rangle_{123})$$

and

$$\langle x|_1\langle 0|_{23}M_f^{-1} = \langle f(x)|_1\langle 0|_{23}.$$

Thus, for $x \in Y_n$ we have

$$(\langle x|_1\langle 0|_{23})M_f^{-1}\tilde{M}_f(|x\rangle_1|0\rangle_{23}) = \beta_{f(x)}$$

and hence from $|1-\beta_{f(x)}| \leq \frac{1}{2^{q(n)}}$ we obtain Equation (2.6), which completes the proof.□

It is an open problem whether the converse of the above theorem holds. However, by restricting the second parameters of pseudo identity operators, we can prove the following restricted version of the converse of Theorem 34.

Theorem 35 *Let f:$\{0,1\}^* \to \{0,1\}^*$ be a permutation that can be computed by a classical polynomial size network. If for any polynomial p and infinitely many n there exist a polynomial r_p and an $r_p(n)$-qubit $(1/2^{p(n)}, p(n)/2^n)$-pseudo identity operator $J_{p(n)}$ such that the family*

$$F_{n,p} = \{\tilde{Q}_j\}_j = \{(I_n \otimes J_{p(n)})^\dagger (Q_j \otimes I_{r_p(n)-n})(I_n \otimes J_{p(n)})\}_{j=0,1,\ldots,\frac{n}{2}-1}$$

is easy, then f is not weakly quantum one-way.

Proof Assume that for a fixed polynomial p, infinitely many n, and some $(1/2^{p(n)}, p(n)/2^n)$-pseudo identity operator $J_{p(n)}$ the family $F_{n,p}$ is easy. To show that f is not a weakly quantum one-way permutation we give a polynomial size algorithm for inverting f. Algorithm $\widetilde{\mathbf{C}}$ has the same steps as Algorithm **C** except the following two changes:

(i) The number of iterations of Step 2 is now $\frac{n}{2} - \lceil 2\log p(n)\rceil$.

(ii) The operator Q_j is now replaced by $\tilde{Q}_j$.

A quantum network implementation for Algorithm $\widetilde{\mathbf{C}}$ consists of three registers. The first and the second registers consist of n qubits similar to the network for Algorithm **C**. The third register consists of $r_p(n) - n$ qubits. From the definition of pseudo identity operators, there exists a set X_n with $|X_n| \leq p(n)$ such that if $y \in Y_n = \{0,1\}^n \setminus X_n$,

$$J_{p(n)}|y\rangle_2|0\rangle_3 = \alpha_y|y\rangle_2|0\rangle_3 + |\psi_y\rangle_{23}, \tag{2.7}$$

where $|\psi_y\rangle_{23} \perp |y\rangle_2|0\rangle_3$ and $|1-\alpha_y| \leq \frac{1}{2^{p(n)}}$.

In Algorithm $\widetilde{\mathbf{C}}$, we apply $J_{p(n)}$ before and after Step $2.j.2$ for each j. The application of $J_{p(n)}$ creates an error in computation of f^{-1}. We call the vector $J_{p(n)}|\psi\rangle - |\psi\rangle$, the error associated to $|\psi\rangle$. To measure the effect of this error, we use the following lemmas (the proof is given later).

Lemma 36 *Assume that $T \subseteq S \subseteq \{0,1\}^n$. Then length $l(S,T)$ of the error associated to the state*

$$|\psi(S,T)\rangle = \frac{1}{\sqrt{|S|}} \left(\sum_{y \in S \setminus T} |y\rangle|0\rangle - \sum_{y \in T} |y\rangle|0\rangle \right),$$

satisfies the following relation

$$l(S,T) \leq \frac{\frac{2}{2^{\frac{p(n)}{2}}} \cdot |S \cap Y_n| + 2|S \cap X_n|}{\sqrt{|S|}}.$$

From Lemma 36 one can easily check the following lemma.

Lemma 37 *Let $J_{p(n)}|\psi(S,T)\rangle = \alpha|\psi(S,T)\rangle + |\psi(S,T)^{\perp}\rangle$, where $|\psi(S,T)\rangle \perp |\psi(S,T)^{\perp}\rangle$. Then $||\psi(S,T)^{\perp}\rangle| \leq l(S,T)$.*

First, suppose that for some $j = k$ all steps before step $2.k.2$ of Algorithm $\widetilde{\mathbf{C}}$ have been implemented as Algorithm $\mathbf{C}$. By a similar argument to the proof of Theorem 28 we get the state

$$|x\rangle_1|\psi(S,T)\rangle_{23} = |x\rangle_1 \frac{2^k}{\sqrt{2^n}} \left(\sum_{y \in S \setminus T} |y\rangle_2 - \sum_{y \in T} |y\rangle_2 \right) |0\rangle_3,$$

where $S = \{y : f(y)_{(1,2k)} = x_{(1,2k)}\}$ and $T = \{y : f(y)_{(1,2k+2)} = x_{(1,2k+2)}\}$. In Algorithm $\widetilde{\mathbf{C}}$, $J_{p(n)}$ is applied for the state $|\psi(S,T)\rangle_{23}$. For $k \leq n/2 - \lceil 2\log p(n) \rceil$, from Lemma 36 we have

$$l(S,T) \leq \frac{\frac{2}{2^{\frac{p(n)}{2}}} \cdot |S \cap Y_n| + 2|S \cap X_n|}{\sqrt{|S|}}$$

$$\begin{aligned}
&\leq \frac{\frac{2}{2^{\frac{p(n)}{2}}} \cdot |S| + 2|X_n|}{\sqrt{|S|}} \\
&\leq \frac{\frac{2}{2^{\frac{p(n)}{2}}} \times 2^{n-2k} + 2p(n)}{\sqrt{2^{n-2k}}} \leq \frac{2^{n+1-\frac{p(n)}{2}} + 2p(n)}{\sqrt{2^{n-2k}}} \\
&\leq \frac{4p(n)}{2^{\frac{n}{2}-k}} \leq \frac{4p(n)}{2^{\lceil 2\log p(n)\rceil}} \\
&\leq \frac{4}{p(n)}.
\end{aligned}$$

Therefore, for $k \leq n/2 - \lceil 2\log p(n)\rceil$, from Lemma 37 we get a vector $v = v_1 + v_2$ where $\frac{v_1}{|v_1|}$ is the unit vector corresponding to the state before Step $2.k.2$ (up to a total phase) and v_2 is a vector of length at most $\frac{4}{p(n)}$ orthogonal to v_1. The vector v_2 corresponds to an error which happens when $J_{p(n)}$ is applied before Step $2.k.2$.

Next, assume that for some $j = k$ all steps before Step $2.k.2$ and Step $2.k.2$ itself have been implemented in the same way as for Algorithm **C**. We obtain the state

$$|x\rangle_1 |\psi(S,T)\rangle_{23} = |x\rangle_1 \frac{2^{k+1}}{\sqrt{2^n}} \sum_{y\in S} |y\rangle_2 |0\rangle_3,$$

where $S = \{y : f(y)_{(1,2k+2)} = x_{(1,2k+2)}\}$ and $T = \emptyset$. By a similar argument to the above, we get a vector $v = v_1 + v_2$, where $\frac{v_1}{|v_1|}$ is the unit vector corresponding to the state after Step $2.k.2$ and v_2 is a vector of length at most $\frac{4}{p(n)}$ orthogonal to v_1. The vector v_2 corresponds to an error which occurs when $J_{p(n)}$ is applied after Step $2.k.2$.

Now, from the above analysis, we can see that after the completion of Algorithm $\widetilde{\mathrm{C}}$ on input x the final state is $v = v_1 + v_2$, where v_1 is parallel to

$$|x\rangle_1 \frac{1}{\sqrt{2^{2\lceil 2\log p(n)\rceil}}} \sum_{y: f(y)_{(1,n-2\lceil 2\log p(n)\rceil)} = x_{(1,n-2\lceil 2\log p(n)\rceil)}} |y\rangle_2 |0\rangle_3$$

and v_2 is a vector of length at most $2(n/2 - \lceil 2\log p(n)\rceil)(4/p(n))$ orthogonal to v_1. Thus, $|v_2| \leq 1/q(n)$ for some polynomial q. We know in advance that for any x the probability of obtaining $f^{-1}(x)$ upon measuring the second register in the state v_1 is $1/2^{2\lceil 2\log p(n)\rceil}$. Now, using the algorithm in [47] (the quantum amplitude

amplification when the success probability is known), we can change the state v into $w = w_1 + w_2$, where w_1 is parallel to $|x\rangle_1|f^{-1}(x)\rangle_2|0\rangle_3$, $w_1 \perp w_2$, and $|w_2|^2 = |v_2|^2 \leq \frac{1}{q^2(n)}$. Therefore, there exist a polynomial size quantum network B and infinitely many n such that

$$\frac{1}{2^n} \sum_{x \in \{0,1\}^n} \mathrm{Prob}[B(x) = f^{-1}(x)] > 1 - \frac{1}{q^2(n)} .$$

We can give any large polynomial $q^2(n)$ by taking any large polynomial p. Thus, f is not weakly quantum one-way. □

Finally, we give the proof of Lemma 36.

Proof Lemma 36 First, we show that the length of the error associated to the state $|y\rangle|0\rangle$ is at most $\frac{2}{2^{\frac{p(n)}{2}}}$ if $y \in Y_n$, and is at most 2 if $y \in X_n$. For $y \in Y_n$, from Equation (2.7) we have $1 - |\alpha_y| \leq |1 - \alpha_y| \leq \frac{1}{2^{p(n)}}$, and hence

$$||\psi_y\rangle_{23}|^2 = 1 - |\alpha_y|^2 = (1 + |\alpha_y|)(1 - |\alpha_y|) \leq \frac{2}{2^{p(n)}}.$$

Thus, for the length of the error associated to $|y\rangle|0\rangle$ we obtain the following relation

$$\begin{aligned} |J_{p(n)}|y\rangle_2|0\rangle_3 - |y\rangle_2|0\rangle_3| &= |(\alpha_y - 1)|y\rangle_2|0\rangle_3 + |\psi_y\rangle_{23}| \\ &= \sqrt{|\alpha_y - 1|^2 + ||\psi_y\rangle_{23}|^2} \\ &\leq \sqrt{(\frac{1}{2^{p(n)}})^2 + \frac{2}{2^{p(n)}}} \\ &\leq \sqrt{\frac{4}{2^{p(n)}}} = \frac{2}{2^{\frac{p(n)}{2}}} . \end{aligned}$$

On the other hand, if $y \in X_n$, we have

$$|J_{p(n)}|y\rangle|0\rangle - |y\rangle|0\rangle| \leq |J_{p(n)}|y\rangle|0\rangle| + ||y\rangle|0\rangle| \leq 2.$$

Finally, for the length $l(S,T)$ of the error associated to the state $|\psi(S,T)\rangle$ we have

$$l(S,T) = |J_{p(n)}|\psi(S,T)\rangle - |\psi(S,T)\rangle|$$

$$
\begin{aligned}
&\leq \frac{1}{\sqrt{|S|}}\left(\sum_{y\in S\setminus T}|(J_{p(n)}-I)|y\rangle|0\rangle|+\sum_{y\in T}|(J_{p(n)}-I)|y\rangle|0\rangle|\right)\\
&= \frac{1}{\sqrt{|S|}}\sum_{y\in S}|(J_{p(n)}-I)|y\rangle|0\rangle|\\
&= \frac{1}{\sqrt{|S|}}\left(\sum_{y\in S\cap Y_n}|(J_{p(n)}-I)|y\rangle|0\rangle|+\sum_{y\in S\cap X_n}|(J_{p(n)}-I)|y\rangle|0\rangle|\right)\\
&\leq \frac{1}{\sqrt{|S|}}\left(\frac{2}{2^{\frac{p(n)}{2}}}|S\cap Y_n|+2|S\cap X_n|\right).\Box
\end{aligned}
$$

From Proposition 32, Theorem 34 and Theorem 35, we obtain the following relationship between the existence of quantum one-way permutations and the reflection operators about a particular class of quantum states.

Theorem 38 *The following relations hold.*

(i) There exists a polynomial time computable function f such that: there exists a polynomial p such that for all sufficiently large n and all $(1/2^{p(n)}, 1/p(n))$-pseudo identity operators $J_{p(n)}$,

$$F_{n,p}(f)=\{(I_n\otimes J_{p(n)})^\dagger(Q_j(f)\otimes I_{r_p(n)-n})(I_n\otimes J_{p(n)})\}_{j=0,1,\ldots,\frac{n}{2}-1}.$$

is not easy.

$\Rightarrow$ *(ii) There exists a weakly quantum one-way permutation.*

$\Leftrightarrow$ *(iii) There exists a strongly quantum one-way permutation.*

$\Rightarrow$ *(iv) There exists a polynomial time computable function f such that: there exists a polynomial p such that for all sufficiently large n and all $(1/2^{p(n)}, p(n)/2^n)$-pseudo identity operators $J_{p(n)}$,*

$$F_{n,p}(f)=\{(I_n\otimes J_{p(n)})^\dagger(Q_j(f)\otimes I_{r_p(n)-n})(I_n\otimes J_{p(n)})\}_{j=0,1,\ldots,\frac{n}{2}-1}.$$

is not easy.

On the other hand, for the bounded-error setting in the worst case complexity, we can prove the following necessary and sufficient condition by a similar argument to the proofs of Theorems 34 and 35 (the proof is therefore omitted).

Theorem 39 *The following statements are equivalent.*

(i) Worst case quantum one-way permutations exist in the bounded error setting.

(ii) There exists a polynomially computable function f satisfying the condition: there exists a polynomial p such that for infinitely many n *and all $(1/2^{p(n)}, p(n)/2^n)$-pseudo identity operators $J_{p(n)}$,*

$$\begin{aligned} F_{n,p}(f) &= \{\tilde{Q}_j\}_j \\ &= \{(I_n \otimes J_{p(n)})^\dagger (Q_j(f) \otimes I_{r_{p(n)}-n})(I_n \otimes J_{p(n)})\}_{j=0,1,\ldots,\frac{n}{2}-1}. \end{aligned}$$

is not easy.

Comparing Theorem 39 with Theorem 38, we can see that condition (iv) of Theorem 38 is given essentially to characterise the existence of worst case quantum one-way permutation in the bounded-error setting (the only different part is the condition "all sufficient large" and "infinitely many"). We conjecture that condition (i) of Theorem 38 gives a necessary and sufficient condition for the existence of weakly (and strongly) quantum one-way permutations.

2.4 Complexity Classes

In this section we give the relationship between the existence of one-way functions and well-known complexity classes **UP** and **EQP**. To this end we recall some definitions given in [52]. Assume that C is a complexity class; then we define the complexity class C_g as follows:

$$C_g = \{f \in C \mid \mathrm{Graph}(f) \in \mathbf{P}\},$$

where

$$\mathrm{Graph}(f) = \{(x, y) \,|\, x \in \mathrm{Dom}(f) \,\&\, y = f(x)\}\,.$$

Denote by **QPSV**, the class of all single valued functions which can be computed exactly by polynomial time quantum Turing machines; **NPSV**, the class of all single valued non-deterministic polynomial time computable function; and **UPSV**, the class of all functions f in **NPSV** such that for every x in domain of f there exists a unique accepting computational path. The following lemma introduces two relationships between the quantum and classical complexity classes.

Lemma 40 *The following relations hold:*

(i) **UP** $\subseteq$ **EQP**

$\Rightarrow$ *(ii)* **UPSV** $\subseteq$ **QPSV**

$\Rightarrow$ *(iii)* **UPSV**$_g$ $\subseteq$ **QPSV**.

Proof The proof of (ii) $\Rightarrow$ (iii) is trivial. We give a sketch of the proof of (i) $\Rightarrow$ (ii) [52]. Assume that f is in **UPSV** and define R_f to be the following language:

$$R_f = \{(x, y) | x \in \mathrm{Dom}(f) \,\&\, y \leq f(x)\}\,.$$

Since $f \in$ **UPSV**, given input (x, y) one can compute $f(x)$ unambiguously and then check from the output whether $y \leq f(x)$. This shows that R_f belongs to **UP** and by assumption also belongs to **EQP**. Therefore using binary search one can show that $f \in$ **QPSV**. □

Now using a similar method to [52] we can prove the following theorem.

Theorem 41 *There exists a worst case quantum one-way function if and only if*

$$\textbf{UP} \not\subseteq \textbf{EQP}\,.$$

Proof ($\Rightarrow$) Assume that f is a worst case quantum one-way function. Then by definition we have $f^{-1} \in \mathbf{UPSV}_g$. However $f^{-1} \notin \mathbf{QPSV}$ therefore from Lemma 40 we derive $\mathbf{UP} \not\subseteq \mathbf{EQP}$.

($\Leftarrow$) Assume L to be a language in $\mathbf{UP} \setminus \mathbf{EQP}$ and M to be an unambiguous Turing machine accepting L. Then, the total function f defined below is a worst case one-way function:

$$f(x) = \begin{cases} y0 & \text{if } x = \mathrm{Comp}_M(y) \\ x1 & \text{otherwise,} \end{cases}$$

where $\mathrm{Comp}_M(y)$ denote the unique accepting computation of M on input y. □

2.5 State and Operator Complexity

The study of states and operators complexity is an important way to find the relationship between physical complexity and computational complexity. As we show in the following, the special case of the relationship between the complexity of preparing a state and the complexity of performing the operator of reflection about that state, has a close connection with the question of the existence of quantum one-way functions. We introduce a notion of complexity of preparing quantum states and constructing unitary transformations. We consider families of the states and unitary operators, and introduce the complexity classes similar to classical computation. Define $\mathcal{S}$ to be the set of all families $S_p^A = \{|\psi_x\rangle\}_{x \in A}$ where p is an increasing function, A is a language, and $|\psi_x\rangle$ is an arbitrary $p(|x|)$-qubits state. We also define $\mathcal{O}$ to be the set of all families $U_p^A = \{U_x\}_{x \in A}$, where p is an increasing function, A is a language, and U_x is an arbitrary unitary transformation acting on $p(|x|)$-qubits and p is a polynomial. In what follows, we omit the symbols p and A for the simplicity. (We consider $A = \{0,1\}^*$ and $p(x) = |x|$ in most of the cases.)

Definition 42 *A family $\{|\psi_x\rangle\}_x \in \mathcal{S}$ of states is defined to be* computable*, if there exists a uniform quantum network family $N = \{N_x\}$ such that on input x, N_x produces exactly the output state $|\psi_x\rangle$. We denote by $C\mathcal{S}$ the set of all* computable families of states.

Definition 43 *A family $\{U_x\}_x \in \mathcal{O}$ of unitary operators is defined to be* computable*, if there exists a uniform quantum network family $N = \{N_x\}$ such that on input x and $|\psi\rangle$, N_x produces exactly the output $U_x(|\psi\rangle)$. We denote by C$\mathcal{O}$ the set of all* computable families of unitary operators*.*

The analogue of the complexity classes of families of states and unitary operators corresponding to **P** and **PSPACE** can also be defined in a similar fashions.

Definition 44 *A family $\{|\psi_x\rangle\}_x \in \mathcal{S}$ of states is* polynomial-time (or space) computable*, if there exists a polynomial-time (or space) uniform quantum network family $N = \{N_x\}$ such that on input x, N_x produces exactly the output state $|\psi_x\rangle$. We denote by* **P**$\mathcal{S}$ *(or* **PSPACE**$\mathcal{S}$*) the set of all polynomial-time (or space) computable families of states.*

Definition 45 *A family $\{U_x\}_x \in \mathcal{O}$ of unitary operators is polynomial-time (or space) computable, if there exists a polynomial-time (or space) uniform quantum network family $N = \{N_x\}$ such that on input x and $|\psi\rangle$, N_x produces exactly the output $U_x(|\psi\rangle)$. We denote by* **P**$\mathcal{O}$ *(or* **PSPACE**$\mathcal{O}$*) the set of all polynomial-time (or space) computable families of unitary operators.*

In what follows, we consider the relationship between states and reflection operators about those states. The reflection operator about a given state $|\psi\rangle$ is defined to be

$$2|\psi\rangle\langle\psi| - I\,.$$

The reflection operators have many interesting properties. Here we mainly study them from the complexity theoretic point of view. It is well-known that if a state is preparable in polynomial time, then the reflection about that state can implement in polynomial time (Problem $6.2(1)$ in [79]). To see this, without loss of generality assume that $\{|\psi_x\rangle\}_x$ is a polynomial-time computable family of states and $N = \{N_x\}_x$ is a uniform polynomial size network family implementing a family $\{U_x\}_x$ of unitary operators, where $U_x|0\rangle^{\otimes n} = |\psi_x\rangle$. Therefore, we have:

$$U_x(2|0\rangle\langle 0| - I)U_x^\dagger = 2|\psi_x\rangle\langle\psi_x| - I\,,$$

which can be implemented by a uniform polynomial size network family. This arguments can be easily applicable to the case of a computable, polynomial-time or polynomial-space computable family:

Proposition 46 *If $\{|\phi_x\rangle\}_x$ is in $\mathcal{CS}$ (resp.* **PSPACE**$\mathcal{S}$*,* **P**$\mathcal{S}$*), then the sequence of reflection operators:*

$$\{2|\phi_x\rangle\langle\phi_x| - I\}_x \,,$$

is in $\mathcal{CO}$ (resp. **PSPACE**$\mathcal{O}$*,* **P**$\mathcal{O}$*).*

Does the inverse hold? In particular, we consider the inverse of the above proposition for the polynomial-time case

Reflection Assumption: *Assume that $\{|\phi_x\rangle\}_x$ is in $\mathcal{CS}$ is given such that the family of reflection operators $\{2|\phi_x\rangle\langle\phi_x| - I\}_x$ is polynomial-time computable. Then, $\{|\phi_x\rangle\}_x$ is also polynomial-time computable.*

We shall relate the Reflection Assumption to the existence of quantum one-way permutation by revisiting INVERT problem. Let f be a permutation on n-bit strings, and U_f the unitary operator mapping the basis state $|x\rangle|y\rangle$ to $|x\rangle|f(x) \oplus y\rangle$, where $|x\rangle$ and $|y\rangle$ each consist of n qubits. Given U_f as an oracle, Algorithm **B** for INVERT computes $f^{-1}(x)$ with high probability in $O(\sqrt{2^n})$ queries and this algorithm is shown to be optimal [6]. Note that the operator

$$2|f^{-1}(x)\rangle\langle f^{-1}(x)| - I \,,$$

is performing the reflection about the state $|f^{-1}(x)\rangle$. Thus, Algorithm **B** shows that even if the reflection about the state $|f^{-1}(x)\rangle$ is *assumed to be* polynomial-time computable (Equation 2.1), the state itself is not necessarily computable by a polynomial-time quantum Turing machine with oracle U_f.

Now consider a family of unitary operators $\{U_{f_n}\}_n$, where U_{f_n} is the unitary operator implementing f_n exactly. By condition (ii) of Definition 27, $\{U_{f_n}\}_n$ is

polynomial-time computable. Therefore using Equation 2.1 we obtain the following:

Lemma 47 *Assume that f is a quantum one-way permutation, the following family of unitary operators is in* **P**$\mathcal{O}$*:*

$$\{2|f^{-1}(x)\rangle\langle f^{-1}(x)| - I\}_x .$$

On the other hand, by condition (iii), the family of states $\{|f^{-1}(x)\rangle\}_x$ is not easy. This implies the following interesting fact:

Proposition 48 *If there exists a quantum one-way permutation, then we can construct a counter-example to the Reflection Assumption.*

We can make sure that by a minor modification the above proposition holds under the existence of a quantum one-way function, which is equivalent to the open problem that **UP** is not included in **EQP**. Can we make this assumption weaker, for example, based on the separation between **EQP** and **PSPACE**? This is still open. Instead, we present the following simple facts.

Theorem 49 *If* **EQP** $\neq$ **PSPACE**, *then* **PSPACE**$\mathcal{S} \setminus$ **P**$\mathcal{S} \neq \emptyset$.

Proof Consider a **PSPACE**-complete language L, therefore

$$L \in \textbf{PSPACE} \setminus \textbf{EQP} .$$

We identify L with its characteristic function. Clearly, $\{|x, L(x)\rangle\}_x \in$ **PSPACE**$\mathcal{S}$. Now assume that $\{|x, L(x)\rangle\}_x$ is in **P**$\mathcal{S}$. Then, $\{I - 2|x, L(x)\rangle\langle x, L(x)|\}_x$ is in **P**$\mathcal{O}$. We have:

$$\begin{aligned}(I - 2|x, L(x)\rangle\langle L(x), x|)(|x, 0\rangle + |x, 1\rangle + |x, 2\rangle + |x, 3\rangle) \\ = \begin{cases} |x, 0\rangle - |x, 1\rangle + |x, 2\rangle + |x, 3\rangle & \text{if } L(x) = 1 \\ -|x, 0\rangle + |x, 1\rangle + |x, 2\rangle + |x, 3\rangle & \text{if } L(x) = 0 \end{cases}\end{aligned}$$

and hence one can design an exact quantum polynomial algorithm for accepting L on input x. This contradicts the assumption and we derive that

$$\{|x, L(x)\rangle\}_x \in \mathbf{PSPACE}\mathcal{S} \setminus \mathbf{P}\mathcal{S}\,.$$

□

Corollary 50 *$\mathcal{S} \setminus C\mathcal{S} \neq \emptyset$.*

Proof Define the family $\{|x, L(x)\rangle\}_x$ of states similar to the proof of Theorem 49, where L is the halting function. □

Corollary 51 *If* **EQP** $\neq$ **PSPACE**, *then there exists a family of states*

$$\{|\psi_x\rangle\}_x \in \mathbf{PSAPCE}\mathcal{S} \setminus \mathbf{P}\mathcal{S} \neq \emptyset\,,$$

such that the family of reflection operators $\{I - 2|\psi_x\rangle\langle\psi_x|\}_n$ is in **PSPACE**$\mathcal{O} \setminus \mathbf{P}\mathcal{O}$.

Proof The family in the proof of Theorem 49 will work. □

Note that in Theorem 49 and Corollary 51, the complexity class **PSPACE** can be replaced with any other complexity class **A** having a complete language L, as far as the following condition is satisfied: $\{|x, L(x)\rangle\}_x$ (or $\{I - 2|x, L(x)\rangle\langle x, L(x)|\}_x$) is in the class of states (or unitary operators) corresponding to the class **A** of languages.

The notion of Turing reducibility can also be generalized to the setting of the state and operator complexity as follows.

Definition 52 *Assume that the two families $\{|\phi_x\rangle\}_x$ and $\{|\psi_x\rangle\}_x$ of states are given, we define $\{|\phi_x\rangle\}_x$ to be* polynomial-time Turing reducible *to $\{|\psi_x\rangle\}_x$ (denoted by $\{|\phi_x\rangle\}_x \leqslant_T^p \{|\psi_x\rangle\}_x$), if $\{|\phi_x\rangle\}_x$ can be prepared by a polynomial-time quantum Turing machine given oracle $|x\rangle|0\rangle \mapsto |x\rangle|\psi_x\rangle$.*

Definition 53 *Assume the two families $\{U_x\}_x$ and $\{V_x\}_x$ of unitary operators are given, we define $\{U_x\}_x$ to be* polynomial-time Turing reducible *to $\{V_x\}_x$ (denoted by $\{U_x\}_x \leqslant_T^p \{V_x\}_x$), if $\{U_x\}_x$ can be implemented by a polynomial-time quantum Turing machine given oracle $|x\rangle|y\rangle \mapsto |x\rangle V_x|y\rangle$.*

We end this section with the following two conjectures which seem to hold intuitively.

Conjecture 1 *For given states $|\phi\rangle$ and $|\psi\rangle$ we have the following relationship between states and reflection operators:*

$$\{|\phi_x\rangle\}_x \leqslant_T^p \{|\psi_x\rangle\}_x \Leftrightarrow \{I - 2|\phi_x\rangle\langle\phi_x|\}_x \leqslant_T^p \{I - 2|\psi_x\rangle\langle\psi_x|\}_x \,.$$

Conjecture 2 *If* **EQP** $\neq$ **PSPACE***, then we can construct a counter-example to the Reflection Assumption.*

2.6 Discussion

We have reduced the problem of the existence of a quantum one-way permutation to the problem of constructing a polynomial size network for performing the specific task of the reflection about a given state. Ambainis [6] proved that inverting a permutation on the n-bit strings in the standard query model requires $\Omega(\sqrt{2^n})$ queries. In the standard query model [7], a quantum computation with T queries is a sequence of unitary operators

$$U_0 \to O \to U_1 \to O \cdots \to U_{T-1} \to O \to U_T,$$

where U_j's are arbitrary unitary operators independent of a database to be searched or a permutation to be computed, and O is the standard query operator. However, our algorithm is consistent with Ambainis' result, since we consider the case that U_j's depend on a permutation to be computed and this does not fit his model.

Another related issue is the work of Chen and Diao [25] where they attempted to present an efficient quantum algorithm for the problem SEARCH, which is similar to our algorithm for the problem INVERT. They mentioned that the tagging operation and the reflection about a given state which varies dynamically can be constructed by polynomial size networks, but they did not show the construction for their operations. (This construction is, of course, impossible given Grover's black box, since it would violate the optimality proof of Grover's algorithm [18, 115, 6].)

For the problem INVERT we have given a polynomial size network for the tagging operation and we have shown that the difficulty of the construction of the reflection operation is equivalent to the existence of the quantum one-way permutation. Furthermore it is an interesting open problem whether there exists a reduction from other types of one-way functions to constructing a polynomial size network for performing the reflection about a given state.

On the other hand, we have seen that Grover's algorithm gives us an example of states that are difficult to prepare but the reflections about these states are easy, i.e., it provides a counter-example to Reflection Assumption assuming the existence of one-way permutations. This investigation of Reflection Assumption seems to be useful for cryptographic applications since recently, quantum bit commitment protocols based on quantum one-way permutations have been proposed [36, 3]. Moreover, it is interesting to find such a concrete counter-example without the existence of quantum one-way permutations. Presenting such examples of states may provide us with more ideas for constructing novel quantum algorithms.

3

Quantum Oracle

3.1 Introduction

Query complexity is a simple framework to study the power of oracles to separates quantum complexity classes from classical one. The query complexity of a function is the minimum number of queries to some oracle that are needed to compute one value of this function (Chapter 1). Most quantum algorithms are defined in this simple setting. Examples of quantum oracle algorithms that are provably better than any classical algorithms can be found in [34, 53, 98, 14, 104, 30, 7, 21]. In this chapter we introduce an alternative definitions for quantum oracle and compare its computational power with the standard oracle.

3.2 Minimal Oracle

In this section we compare the query complexity analysis of quantum algorithms given two different ways of representing a permutation in terms of a black box quantum oracle. Consider the following oracles, defined for a permutation function

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

- the *standard* oracle, $S_f : |x\rangle|b\rangle \rightarrow |x\rangle|b \oplus f(x)\rangle$.
- the *Fourier phase* oracle, $P_f : |x\rangle|b\rangle \rightarrow e^{2\pi i f(x)b/2^n}|x\rangle|b\rangle$.

Here x and b are strings of n bits, represented as numbers modulo $N = 2^n$, $|x\rangle$ and $|b\rangle$ are the corresponding computational basis states, and $\oplus$ is addition modulo 2^n.

Note that the oracles P_f and S_f are equivalent, in the sense that each can be constructed by an f-independent quantum circuit containing just one copy of the other. To see this, define the quantum Fourier transform operation F by

$$F : |j\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{2^n-1} e^{2\pi i j k/N} |k\rangle \, .$$

Then one query to the standard oracle can be simulated with one query to the Fourier phase oracle as following:

$$\begin{aligned}
|x\rangle|b\rangle &\xrightarrow{F} \frac{1}{\sqrt{N}} \sum_{k=0}^{2^n-1} e^{2\pi i b k/N} |x\rangle|k\rangle \\
&\xrightarrow{P_f} \frac{1}{\sqrt{N}} \sum_{k=0}^{2^n-1} e^{2\pi i b k/N} e^{2\pi i k f(x)/N} |x\rangle|k\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{k=0}^{2^n-1} e^{2\pi i k (b+f(x))/N} |x\rangle|k\rangle \\
&\xrightarrow{F^{-1}} |x\rangle|b \oplus f(x)\rangle
\end{aligned}$$

In a similar way one query to the Fourier phase oracle can be simulated with one query to the standard oracle. In summary the following relations holds:

$$(I \otimes F^{-1}) \circ P_f \circ (I \otimes F) = S_f \, , (I \otimes F) \circ S_f \circ (I \otimes F^{-1}) = P_f \, ,$$

where $\circ$ represents the composition of operations (or the concatenation of networks).

Furthermore if f is a one-to-one function (e.g. a permutation on the set $\{0,1\}^n$), then there is a simpler invertible quantum map associated to f:

- the *minimal* oracle: $M_f : |x\rangle \to |f(x)\rangle$.

In the following, we examine the minimal and standard oracle in simulating each other. Figure 3.1 shows a simulation of standard oracle with minimal oracle. Starting with the initial state $\sum_{x\in S} |x\rangle \otimes |0\rangle$, the first n CNOT gates create the entangled superposition $\sum_{x\in S} |x\rangle|x\rangle$. Then applying the minimal oracle M_f on the second register gives $\sum_{x\in S} |x\rangle|f(x)\rangle$ and this completes the simulation. In the case that the initial state is $\sum_{x\in S} |x\rangle \otimes |b\rangle$, we can construct S_f from M_f and $(M_f)^{-1}$ as follows:

$$S_f = (M_{f^{-1}} \otimes I) \circ A \circ (M_f \otimes I)\,,$$

where the modulo N adder A is defined by $A : |a\rangle \otimes |b\rangle \to |a\rangle \otimes |a \oplus b\rangle$ and

$$(M_f)^{-1} = M_{f^{-1}}\,.$$

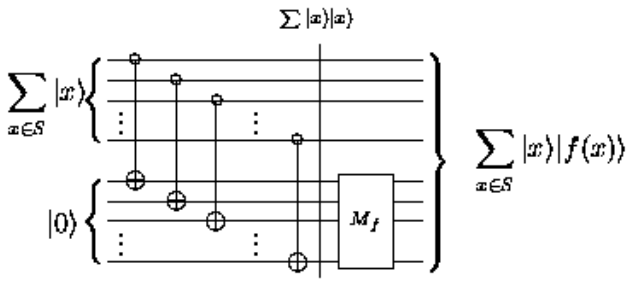

Figure 3.1: A quantum circuit for simulating standard oracle with minimal oracle.

Note that if M_f is given in the form of a specified complicated quantum circuit, we may be completely unable to simplify the circuit or deduce a simpler form of f from it. However, by reversing the circuit gate by gate, we can construct a circuit for $(M_f)^{-1}$. Hence, by the above construction, we can produce a circuit for S_f,

using one copy and one reversed copy of the circuit for M_f. This way of looking at oracles can be formalised as following :

Definition 54 *The query complexity of an algorithm involving an oracle O_f associated to a function f is the number of copies of O_f and/or O_f^{-1} required to implement the algorithm in a circuit that, apart from the oracles, is independent of f.*

In the circuit model, a standard oracle can easily be simulated given a minimal oracle. Ignoring constant factors, we say that the minimal oracle is at least as strong as the standard oracle. On the other hand we show that simulating M_f requires exponentially many uses of S_f.

First, consider the standard oracle $S_{f^{-1}}$ which maps a basis state $|y\rangle|b\rangle$ to $|y\rangle|b \oplus f^{-1}(y)\rangle$. Since

$$S_{f^{-1}} : |y\rangle|0\rangle \rightarrow |y\rangle|f^{-1}(y)\rangle\,,$$

simulating it allows us to solve the search problem of identifying $|f^{-1}(y)\rangle$ from a database of N elements. It is known that, using Grover's search algorithm, one can simulate $S_{f^{-1}}$ with $O(\sqrt{N})$ invocations of S_f [18, 47]. In the following we explain one possible way of doing that.

Prepare the state $|y\rangle|0\rangle|0\rangle|0\rangle$, where the first three registers consist of n qubits and the last register is a single qubit. Apply Hadamard transformations on the second register to get

$$|\phi_1\rangle = \frac{1}{\sqrt{N}}|y\rangle \sum_{x\in\{0,1\}^n} |x\rangle|0\rangle|0\rangle\,.$$

Invoking S_f on the second and third registers now gives

$$\frac{1}{\sqrt{N}}|y\rangle(\sum_{x\in\{0,1\}^n} |x\rangle|f(x)\rangle)|0\rangle\,.$$

Using CNOT gates, compare the first and third registers and put the result in the fourth, obtaining

$$\frac{1}{\sqrt{N}}\Big(\Big(|y\rangle \sum_{x\in\{0,1\}^n, x\neq f^{-1}(y)} |x\rangle|f(x)\rangle|0\rangle\Big) + \Big(|y\rangle|f^{-1}(y)\rangle|y\rangle|1\rangle\Big)\Big) .$$

Now apply $(S_f)^{-1}$ on the second and third registers, obtaining

$$\frac{1}{\sqrt{N}}\Big(\Big(|y\rangle \sum_{x\in\{0,1\}^n, x\neq f^{-1}(y)} |x\rangle|0\rangle|0\rangle\Big) + \Big(|y\rangle|f^{-1}(y)\rangle|0\rangle|1\rangle\Big)\Big) .$$

Note that the simulation of $(S_f)^{-1}$ given S_f is easy based on the following relation:

$$(I \otimes R) \circ S_f \circ (I \otimes R) = (S_f)^{-1} .$$

where $R = F^2$ is the oarity reflection operator defined by :

$$R : |j\rangle \rightarrow |-j\rangle .$$

Taken together, these operations leave the first and third registers unchanged, while their action on the second and fourth defines an oracle for the search problem. Applying Grover's algorithm to this oracle, we obtain the state $|y\rangle|f^{-1}(y)\rangle$ after $O(\sqrt{N})$ invocations.

Theorem 55 *To simulate the inverse oracle $S_{f^{-1}}$ with a quantum network using oracles S_f and $(S_f)^{-1}$, a total number of $\Theta(\sqrt{N})$ invocations of S_f are necessary.*

Proof. The upper bound of $O(\sqrt{N})$ is implied by the Grover-based algorithm just discussed. Ambainis [6] has shown that $\Omega(\sqrt{N})$ invocations of the standard oracle S_f are required to invert a general permutation f. □

Given S_f and $S_{f^{-1}}$, Bennett has shown how to simulate M_f within classical reversible computation [12]. Using a quantum version of this construction, we can establish the following result:

Theorem 56 *To simulate the minimal oracle M_f with a quantum network using oracles S_f and $(S_f)^{-1}$, a total number of $\Theta(\sqrt{N})$ invocations of S_f are necessary.*

Proof. Given S_f and $S_{f^{-1}}$, we can simulate M_f as follows:

$$M_f \otimes I = (S_{f^{-1}})^{-1} \circ X \circ S_f \,,$$

where the swap gate X is defined by $X : |a\rangle \otimes |b\rangle \rightarrow |b\rangle \otimes |a\rangle$. From Theorem 55, $S_{f^{-1}}$ needs $\Theta(\sqrt{N})$ invocations of S_f and $(S_f)^{-1}$. Therefore we get the upper bound of $O(\sqrt{N})$ for simulation of M_f.

However this is the optimal simulation. For suppose there is a network which simulates M_f with less than $\Omega(\sqrt{N})$ queries. The reversed network simulates $M_{f^{-1}}$. From these two, by our earlier results, we can construct a network that simulates $S_{f^{-1}}$ with fewer than $\Omega(\sqrt{N})$ queries, which contradicts Theorem 55. □

It is worth remarking that we could equally well have carried through our discussion using variants of S_f and P_f, such as the bitwise acting versions:

- the *bit string standard* oracle, $S_f^{\text{bit}} : |\mathbf{x}\rangle|\mathbf{b}\rangle \rightarrow |\mathbf{x}\rangle|\mathbf{b} \oplus \mathbf{f}(\mathbf{x})\rangle$.
- the *bit string phase* oracle, $P_f^{\text{bit}} : |\mathbf{x}\rangle|\mathbf{b}\rangle \rightarrow e^{2\pi i \mathbf{f}(\mathbf{x})\cdot\mathbf{b}/2}|\mathbf{x}\rangle|\mathbf{b}\rangle$.

Here $\mathbf{b} \oplus \mathbf{x}$ denotes the bitwise sum mod 2 of the strings $\mathbf{b}$ and $\mathbf{x}$, and $\mathbf{b} \cdot \mathbf{x}$ their inner product mod 2. Again, S_f^{bit} and P_f^{bit} are equivalent: writing

$$\mathcal{F} = H \otimes H \otimes \cdots \otimes H \,,$$

for the tensor product of n Hadamard operators acting on register qubits, we have

$$\begin{aligned} (\quad I \otimes \mathcal{F}) \circ S_f^{\text{bit}} \circ (I \otimes \mathcal{F}^{-1}) &= P_f^{\text{bit}} \,, \\ (\quad I \otimes \mathcal{F}^{-1}) \circ P_f^{\text{bit}} \circ (I \otimes \mathcal{F}) &= S_f^{\text{bit}} \,. \end{aligned}$$

Note also that $S_f^{\text{bit}} = (S_f^{\text{bit}})^{-1}$, $P_f^{\text{bit}} = (P_f^{\text{bit}})^{-1}$. Our results still apply: S_f^{bit} has essentially the same relation to M_f that S_f does.

3.3 Promise Problems

Intuitively minimal oracles seem at least as strong as standard ones, though it is not clear how to simulate the latter with the former without also having access to

the inverse oracle $S_{f^{-1}}$. The question that we consider in this subsection is whether minimal oracles are more useful than standard ones for some problems. To illustrate the different behaviour of standard and minimal oracles, we introduce a promise problem.

Problem 3 *Suppose we are given two permutations, α and β, of Z_N, and a subset S of Z_N. It is promised that the images $\alpha(S)$ and $\beta(S)$ are either identical or disjoint. The problem is to determine which.*

This problem has been also considered in a different context by Buhrman et al [20]. For simplicity we take $N = 2^n$, where n is an integer. We represent elements $x \in Z_N$ by computational basis states of n qubits in the standard way, and write $|S\rangle = \sum_{x \in S} |x\rangle$.

Figure 3.2 gives a quantum network with minimal oracles that identifies disjoint images with probability at least $1/2$.

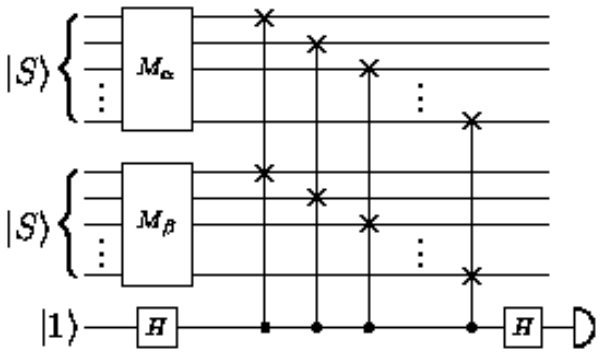

Figure 3.2: A quantum circuit for the permutation promise problem. M_α and M_β are minimal oracles for computing the permutations α and β respectively, $|S\rangle$ is the superposition of all the basis states, H is the Hadamard transformation, and all the other gates are controlled swap gates, where circles signify control bits.

Let $A = \{\alpha(x) | x \in S\}$ and $B = \{\beta(x) | x \in S\}$. One query to the oracles M_α and M_β creates the states $\sum_{i \in A} |i\rangle$ and $\sum_{j \in B} |j\rangle$ respectively. The state before

applying the controlled gates is:

$$\sum_{i\in A, j\in B} |i\rangle|j\rangle \otimes (|0\rangle - |1\rangle)$$

After controlled swap gates, the state becomes:

$$(\sum_{i\in A, j\in B} |i\rangle|j\rangle)|0\rangle - (\sum_{i\in A, j\in B} |j\rangle|i\rangle)|1\rangle\,.$$

The final Hadamard gate on the ancilla qubit gives:

$$\begin{aligned}&(\sum_{i\in A, j\in B} |i\rangle|j\rangle - \sum_{i\in A, j\in B} |j\rangle|i\rangle)|0\rangle\\ +&(\sum_{i\in A, j\in B} |i\rangle|j\rangle + \sum_{i\in A, j\in B} |j\rangle|i\rangle)|1\rangle\,.\end{aligned}$$

A $|0\rangle$ outcome shows unambiguously that the images are disjoint. A $|1\rangle$ outcome is generated with probability 1 if the images are identical, and with probability $1/2$ if the images are disjoint. Repeating the computation K times allows one to exponentially improve the confidence of the result. If after K trials we get $|0\rangle$ at least once, we know for certain that $\alpha(S) \neq \beta(S)$. When all the K outcomes were $|1\rangle$, the conclusion that $\alpha(S) = \beta(S)$ has the conditional probability $p_K = \frac{1}{2^K}$ of having been erroneously generated by disjoint input images. Note that p_K is independent of the problem size and decreases exponentially with the number of repetitions.

Clearly, a naive adaptation of the algorithm to standard oracles does not work. Replacing M_α and M_β by S_α and S_β, and replacing the inputs by $|S\rangle \otimes |0\rangle$, results in output states which are orthogonal if the images are disjoint, but also in general very nearly orthogonal if the images are identical. Applying a symmetric projection as above thus almost always fails to distinguish the cases.

By reformulating the above problem, Aaronson showed an exponential gap between standard and minimal oracle [1].

Problem 4 *[1] Suppose we are given two sequences, $X = x_1 \ldots x_n$ and $Y = y_1 \ldots y_n$, such that for each i, $x_i, y_i \in \{1, \ldots, 2n\}$. A query has the form (b, i),*

where $b \in \{0,1\}$ and $i \in \{1,\ldots,n\}$, and produces $(0,x_i)$ if $b=0$ and $(1,y_i)$ if $b=1$. Sequences X and Y are both one-to-one; that is, $x_i \neq x_j$ and $y_i \neq y_j$ for all $i \neq j$ and it is promised that either

(i) X and Y are equal as sets (that is, $\{x_1,\ldots,x_n\} = \{y_1,\ldots,y_n\}$) or

(ii) X and Y are far as sets (that is, $|\{x_1,\ldots,x_n\} \cup \{y_1,\ldots,y_n\}| \geq 1.1n$).

The problem is to determine which cases holds.

This problem can be solved with high probability in a constant number of queries using an minimal oracle, by using a trick similar to that of Watrous [109] for verifying group non-membership. First, using the oracle, we prepare the uniform superposition

$$\frac{1}{\sqrt{2n}} \sum_{i \in \{1,\ldots,n\}} \left(|0\rangle |x_i\rangle + |1\rangle |y_i\rangle\right) .$$

We then apply a Hadamard gate to the first register, and finally we measure the first register. If X and Y are equal as sets, then interference occurs between every $(|0\rangle |z\rangle , |1\rangle |z\rangle)$ pair and we observe $|0\rangle$ with certainty. But if X and Y are far as sets, then basis states $|b\rangle |z\rangle$ with no matching $|1-b\rangle |z\rangle$ have probability weight at least $1/10$, and hence we observe $|1\rangle$ with probability at least $1/20$ [1].

In [1] Aaronson showed that no efficient quantum algorithm using a standard oracle exists for this problem and proved a lower bound of $\Omega(n^{1/7})$ for this problem with standard oracle.

The above promise problems can be generalised to yet another important problem.

Problem 5 *Suppose we are given two graphs, $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, represented as sets of vertices and edges in some standard notation. The graph isomorphism (GI) problem is to determine whether G_1 and G_2 are isomorphic: that is, whether there is a bijection $f : V_1 \to V_2$ such that $(f(u), f(v)) \in E_2$ if and only if $(u,v) \in E_1$. (We assume $|V_1| = |V_2|$, else the problem is trivial.)*

GI is a problem which is **NP** but not known to be **NP**-complete for classical computers, and for which no polynomial time quantum algorithm is currently known. We are interested in a restricted version (NAGI) of GI, in which it is given that G_1 and G_2 are non-automorphic: i.e., they have no non-trivial automorphisms. So far as we are aware, no polynomial time classical or quantum algorithms are known for NAGI either. The following observations suggest a possible line of attack in the quantum case.

First, for any non-automorphic graph $G = (V, E)$, we can define a unitary map M_G that takes permutations ρ of V as inputs and outputs the permuted graph $\rho(G) = (\rho(V), \rho(E))$, with some standard ordering (e.g. alphabetical) of the vertices and edges, in some standard computational basis representations. That is, writing $|V| = N$, for any $\rho \in S_N$ [1], M_G maps $|\rho\rangle$ to $|\rho(G)\rangle$. Consider a pair (G_1, G_2) of non-automorphic graphs. Given circuits implementing M_{G_1}, M_{G_2}, we could input copies of the state $\frac{1}{\sqrt{N!}} \sum_{\rho \in S_N} |\rho\rangle$ to each circuit, and compare the outputs $|\psi_i\rangle = \sum_{\rho \in S_N} |\rho(G_i)\rangle$. Now, if the graphs are isomorphic, these outputs are equal; if not, they are orthogonal. These two cases can be distinguished with arbitrarily high confidence in polynomial time (as described above), so this would solve the problem.

Our algorithm for NAGI requires constructing circuits for the M_{G_i}, which could be at least as hard as solving the original problem. On the other hand, it is easy to devise a circuit, S_G, which takes two inputs, $|\rho\rangle$ and a blank set of states $|0\rangle$, and outputs $|\rho\rangle$ and $|\rho(G)\rangle$. However, simulating a minimal oracle requires exponentially many invocations of a standard oracle. Therefor to solve the NAGI one should directly construct a polynomial size network defining an M_f oracle for any given one-to-one function f, which would lead to a polynomial time solution of NAGI.

[1] S_N is the set of all permutations on $\{1, 2, \cdots, N\}$.

3.4 Discussion

We have defined the Minimal oracle as an alternative definition for query model. And we have shown that Minimal oracle is exponentially more powerful than standard oracle and can solve different promised problems with constant number of queries. Constructing a minimal oracle requires exponentially many invocations of a standard oracle. We have not, however, been able to exclude the possibility of directly constructing a polynomial size network defining an M_f oracle for any given one-to-one function f, which would lead to a polynomial time solution of NAGI.

To finish our discussion on oracle complexity we consider other oracles settings called states and operators oracle and briefly discuss their relationship with standard and Minimal oracles.

Definition 57 *Suppose a family $S = \{\psi_x\}_x$ of states is given, the* state oracle O_S *is defined as*

$$O(S)|x\rangle|0\rangle = |x\rangle|\psi_x\rangle.$$

The unitary oracle O_U *for a given family $U = \{U_x\}_x$ of unitary operators, is defined as*

$$O(U)|x\rangle|y\rangle = |x\rangle U_x|y\rangle.$$

Minimal oracle can be considered to be a sort of unitary oracle. Proposition 43 in Chapter 2 shows that a quantum Turing machines with a family of reflection operators can be efficiently simulated by a quantum Turing machine with a family of the corresponding states. On the other hand, in [113] Yamakami implicitly suggests the following fact.

Proposition 58 *For any family $\{|\psi_x\rangle\}_x$ of states, there exist a language A and a polynomial-time quantum Turing machine M with oracle A such that M on input x and 1^l produces the output state $|\psi'_{x,l}\rangle$ satisfying $|||\psi'_{x,l}\rangle - |\psi_x\rangle|| \leq 1/l$.*

This fact implies that in bounded error setting a quantum Turing machine with a state oracle can be efficiently simulated by a quantum Turing machine with a standard oracle. On the other hand, we do not know if quantum Turing machines with a family of unitary operators can be efficiently simulated by quantum Turing machine with a language, i.e., quantum Turing machine with standard oracle. If it is impossible, we will be able to see some complexity theoretical gap between the general unitary operators and the reflection operators. We showed that M_f cannot be efficiently simulated by using S_f. However, it is open whether a quantum Turing machine with a standard oracle S_f can be efficiently simulated by a quantum Turing machine with a minimal oracle M_g, where the function f and the permutation g may be different.

4

Quantum Domain Theory

4.1 Introduction

The first two fundamental models of computation encountered by every computer scientist are: $1)$ Turing machines introduced by Alan Turing and $2)$ Lambda calculus introduced by Alonzo Church. The Turing model is the foundation of von-Neumann computers, computational complexity analysis, and imperative programming languages. Lambda calculus on the other hand is the proper framework to study the formal methods and functional programming languages. Both models have been extensively studied in classical computer science and many other equivalent models of computation have also been introduced to address different aspects of information processing.

Quantum computation is traditionally studied via quantum circuit models or in terms of quantum Turing machines, which fit into the first model of computation [32, 33]. In this approach, one specifies how to build more complicated quantum processes out of a few basic building blocks. This is a proper foundation to study the computational complexity and design of new quantum algorithms. It is the case however that in order to analyse other aspects of quantum computation it is necessary that alternative models be developed. For example, the one-way quantum computer (a new model in which measurement plays the central role) presents

new aspects of quantum information processing that can not be analysed properly in other models, such as temporal complexity [89, 88]. As another example, recent developments in quantum programming languages suggest the requirement of models with higher levels of abstraction [81, 82, 91, 23, 95].

Domain theory provides us with an alternative and more abstract model for computation. Domain theory is traditionally a suitable model for information processing given incompletely specified elements [2, 38]. Furthermore domain theory has proven to be a proper mathematical framework to describe denotational semantics for programming languages whilst also being applicable to the study of computability of partial functions [2, 38]. In this chapter we outline this model and extend it to the quantum setting. First we review classical domain theory, and its application in the context of programming languages and computability analysis. Subsequently, we integrate these ideas and present quantum domain theory. This includes a rigourous definition of quantum computability for quantum states and operators, a denotational semantics of quantum computation and a brief review of a recent result on the application of quantum domain theory to quantum information processing.

4.2 Classical Domain Theory

Domain theory was introduced independently by Scott [94] for the study of denotational semantics and by Ershow [43] as a tool for the study of partial computable functions. A complete survey of domain theory and its applications can be found in [2, 38]. Domain Theory has been developed towards the following key applications:

- A mathematical theory of computation for the semantics of programming languages;
- A mathematical theory of computation over partial information;
- An algebraic approach to computability;

In the general picture, a domain may be viewed as a partially ordered set, with added structures to model information processing. In this picture of computation, a specific input (output) is represented by a sequence of elements approximating it.

An algorithm is a function from the input domain to the output domain. In order to describe this model precisely, first we introduce the standard basic language of the domain theory.

Definition 59 *A* partial order set *(poset) is a pair $(P, \sqsubseteq)$, where $\sqsubseteq$ is a binary relation on P such that the following conditions are satisfied:*

- *Reflexibility* . $\forall x \in P : \quad x \sqsubseteq x$.
- *Transitivity* . $\forall x, y, z \in P : \quad x \sqsubseteq y \,\&\, y \sqsubseteq z \Rightarrow x \sqsubseteq z$.
- *Anti-symmetry* . $\forall x, y \in P : \quad x \sqsubseteq y \,\&\, y \sqsubseteq x \Rightarrow x = y$.

An element $\perp \in P$ is called a least element *iff* $\forall x \in P : \quad \perp \sqsubseteq x$.

It is easy to see that if a poset has a least element, then it is unique.

The poset structure appears in many different fields of computer science and physics and in each context the ordering, $\sqsubseteq$, is interpreted differently. In this chapter, $\sqsubseteq$ refers to a notion of information which will be described more precisely later. The notion of a sequence of data is captured via the following structures.

Definition 60 *A subset A in a poset P is called a* chain *iff*

$$\forall x, y \in A : \quad x \sqsubseteq y \quad \vee \quad y \sqsubseteq x \,.$$

Assume A is a chain in the poset P. An upper bound *of A is an element $u \in P$ such that*

$$\forall x \in A : \quad x \sqsubseteq l \,;$$

The least upper bound *of A is denoted by $\sqcup A$.*

Not every chain in a poset has a least upper bound. Adding this property to a poset (chain completeness) will result in a structure rich enough to model denotational semantics, as we describe later.

Definition 61 *The partial order set P is a* chain-complete *(CCPO) iff all chains A in P have a least upper bound $\sqcup A$ in P.*

We shall be interested in continuous functions:

Definition 62 *Assume $(P_1, \sqsubseteq_1)$ and $(P_2, \sqsubseteq_2)$ are given posets. A function $f : P_1 \to P_2$ is called* continuous *iff it is :*

- *Monotone:* $\forall x, y \in P_1 : \quad x \sqsubseteq_1 y \Rightarrow f(x) \sqsubseteq_2 f(y)$.
- *It preserves the least upper bounds of the chains, i.e. for all chains A in P_1:*

$$\sqcup_2\{f(x) \,|\, x \in A\} = f(\sqcup_1 A)\,.$$

For a given function f, define f^0 to be the identity function and $f^{(n+1)} = f \circ f^n$. Now, we can state the fixed-point theorem which is a canonical tool to construct the mathematical object corresponding to a recursive definition.

Theorem 63 Knaster-Tarski Fixed-Point Theorem *Assume $f : P \to P$ is a continuous function on the chain complete poset P with a least element $\bot$. Then*

$$\mathrm{Fix}f = \sqcup\{f^n(\bot) \,|\, n \geq 0\}\,,$$

defines an element of P which is the least fixed-point of f.

Proof First we show that $\mathrm{Fix}f$ is well-defined, by showing that the set

$$\{f^n(\bot) \,|\, n \geq 0\}\,,$$

is a chain in P. Using induction, we can show that for all n and $x \in P$ we have $f^n(\bot) \sqsubseteq f^n(x)$.

- Base step: $f^0(\bot) = \bot$ and therefore $\forall x \in P : \quad \bot \sqsubseteq x$.
- Induction step: Assume for all x in P we have $f^n(\bot) \sqsubseteq f^n(x)$, therefore monotonicity of f implies $f^{(n+1)}(\bot) \sqsubseteq f^{(n+1)}(x)$.

Assume $n \leq m$ and define $x = f^{m-n}(\bot)$, from the above arguments we have $f^n(\bot) \sqsubseteq f^n(x)$ so $f^n(\bot) \sqsubseteq f^m(\bot)$. Hence $\{f^n(\bot) \,|\, n \geq 0\}$ is a chain in P.

Next we show that $\mathrm{Fix}f$ is indeed a fixed-point of f:

$$\begin{aligned} f(\mathrm{Fix}f) &= f(\sqcup\{f^n(\bot) \,|\, n \geq 0\}) \\ &= \sqcup\{f(f^n(\bot)) \,|\, n \geq 0\} \\ &= \mathrm{Fix}f\,. \end{aligned}$$

The last part is to show that $\mathrm{Fix}f$ is the least fixed-point. Assume that x is another fixed-point of f. By definition, $\bot \sqsubseteq x$, and from continuity of f, we have $f^n(\bot) \sqsubseteq f^n(x)$ for all n. On the other hand, since x is a fixed-point for all n, we have $x = f^m(x)$, which then implies $f^n\bot \sqsubseteq x$ for all n. This shows that x is an upper bound of $\{f^n(\bot) \,|\, n \geq 0\}$ and from the definition of least upper bound we obtain $\mathrm{Fix}f \sqsubseteq x$. □

A similar structure to a chain in a poset is a directed set:

Definition 64 *A non-empty subset $A \subset P$ of a poset $(P, \sqsubseteq)$ is* directed *iff :*

$$\forall x, y \in A \;\; \exists z \in A : x, y \sqsubseteq z\,.$$

A directed set corresponds to a consistent set of data. We denote by $\sqcup A$ the *least upper bound* of a directed set, if it exists.

Definition 65 *A partial order set in which every directed subset has a least upper bound, is called a* domain.

The notion of approximation in domain theory is described via the following relation:

Definition 66 *Assume that x and y belong to a domain D. We say that x is* way-below y *or equivalently x* approximates y*, denoted by $x \ll y$, iff for every directed subset $A \subset D$:*

$$y \sqsubseteq \sqcup A \;\; \Rightarrow \;\; \exists a \in A : x \sqsubseteq a\,.$$

A constructive structure for a domain can be introduced via basis elements:

Definition 67 *A subset B of the domain D is called a* basis *iff for each $d \in D$:*

$$A = \{b \in B \mid b \ll d\} \text{ is directed and } d = \sqcup A\,.$$

A domain with a basis is called a *continuous* domain and if the basis is also countable the domain is called an *ω-continuous* domain.

The following definitions provide a topological structure for a domain.

Definition 68 *An* open set *$O \subset D$ of the Scott topology of D is a set which satisfies the following conditions:*

(i) $x \in O \;\&\; x \sqsubseteq y \Rightarrow y \in O$.

(ii) For any directed subset A of D we have $\sqcup A \in O \Rightarrow \exists x \in A : x \in O$.

Dually a closed set *$C \subset D$ is defined with the following conditions:*

(i) $x \in C \;\&\; y \sqsubseteq x \Rightarrow y \in C$.

(ii) For any directed subset $A \subset C$ we have $\sqcup A \in C$.

In any continuous domain, subsets $\Uparrow b = \{x \mid b \ll x\}$ where b belongs to a given basis of the domain, forms a basis for the Scott topology.

We denote by $[D \to D']$ the set of all continuous functions (with respect to the Scott topology) between two domains D and D', which also forms a domain with pointwise ordering:

$$f \sqsubseteq g \quad \text{iff} \quad \forall x \in D : f(x) \sqsubseteq g(x)\,.$$

In summary, in the domain picture of information processing, data are elements of an ω-continuous domain D, and represented as least upper bound of the basis elements. A program is an element of domain of continuous functions, $[D \to D]$ and can be represented as least upper bound of basis elements in $[D \to D]$. In what follows we review the main applications of domain theory in computability analysis and denotational semantics. As we show in each scenario a suitable domain will be constructed.

4.2.1 Computability Analysis

There exist two main approaches to computability analysis in the literature. One is the machine-oriented framework and the other one is the analysis-oriented approach [111]. In the former scenario, the computation is performed on a certain kind of abstract machine. Whereas in the latter, concepts from classical analysis are extended to develop a computability theory for real numbers or indeed any other mathematical spaces.

Recently, a new approach to computability has been developed which is based on domain theory and fits into the aforementioned second framework for computability [112, 48, 17, 40]. In his famous article [94], Scott points out the relationship between continuity versus computability. For most purposes, to detect whether some construction is computationally feasible - it is sufficient to check that it is continuous (which is much easier to determine than computability). We describe briefly how to define computability via domain theory. In the next section we extend this concept to the quantum setting. We define the notion of an *effectively given ω-continuous domain* by putting a proper recursive structure on the elements of a basis of the domain [99, 38].

Definition 69 *Assume domain D is ω-continuous with a countable basis $B = \{b_0, b_1, b_2, \cdots\}$. We say D is* effectively given with respect to B*, if the relation $b_n \ll b_m$ is r.e. (recursively enumerable) in n and m.*

The definition of computable elements is:

Definition 70 *Assume that D is effectively given. An element $x \in D$ is called* computable*, if the set $\{n \in \mathbb{N} \mid b_n \ll x\}$ is r.e.*

We state the following important theorem (without proof) which provides us with a constructive definition of computability.

Theorem 71 *[40] Assume domain D is effectively given, $x \in D$ is computable iff it is the least upper bound of an effective given chain in the basis B i.e. iff there exists*

a total recursive function $f : \mathbb{N} \to \mathbb{N}$ *such that*

$$b_{f(0)} \sqsubseteq b_{f(1)} \sqsubseteq b_{f(2)} \sqsubseteq \cdots \quad \textit{and} \quad x = \bigsqcup_{n \in \mathbb{N}} b_{f(n)} \, .$$

Moreover, the chain can be chosen to be a $\ll$*-chain, i.e. such that* $b_{f(0)} \ll b_{f(1)} \ll b_{f(2)} \ll \cdots$.

Finally the computability of a function is defined as follows.

Definition 72 *Assume that domains* D *and* D' *are effectively given with respect to the basis sets* B *and* B'*. A continuous function* $f : D \to D'$ *is called* computable*, if the relation* $b'_m \ll f(b_n)$ *is r.e. in* n *and* m*.*

4.2.2 Denotational Semantics

The main problem which gave rise to domain theory was that of describing the meaning of recursive definitions of objects or data-types [94]. An important result in this direction is the fixed-point theorem (Section 4.2). Traditionally, semantics studies the meaning of programs, mainly in order to be able to state some correctness properties. The meaning of each phrase in a program is the computation that it describes. There are two main directions in the area of semantics of programming languages that differ in the eras they are based on:

- Operational Semantics, basically uses infinite automata, and programs are studied in terms of the steps or operations by which each program is executed.
- Denotational Semantics, where programs are interpreted as mathematical functions.

Denotational semantics was developed in the early 1970s by Strachey and Scott [93]. They aimed to place the semantics of programming languages on a purely mathematical basis. Denotational semantics assigns a mathematical function not only to a complete program but also to every phrase in the language. This approach has important benefits such as the ability of predicting the behaviour of each program without actually executing it on a computer or reasoning mathematically about programs, for example to prove that one program is equivalent to another.

In this subsection we review a denotational semantics introduced by Kozen for probabilistic computation [74]. This framework will be the basis of our approach to quantum semantics. We will show that quantum computation over density matrices with completely positive maps, has a similar semantical structure as probabilistic computation over random variables. To this end first we present some standard basic definitions for vector spaces [16, 74].

Definition 73 *A subset* $\mathbf{P}$ *in a vector space* $\mathbf{V}$ *is called* positive cone *iff it satisfies the following conditions:*

$$\forall x, y \in \mathbf{P} \textit{ and positive scalars } a, b : ax + by \in \mathbf{P}$$
$$\forall x \in \mathbf{P} : x, -x \in \mathbf{P} \Rightarrow x = 0\,.$$

$\mathbf{P}$ induces a partial order on $\mathbf{V}$ with the following relation:

$$x \sqsubseteq_{\mathbf{P}} y \;\text{ iff }\; y - x \in \mathbf{P}\,.$$

A similar structure to a domain where every directed set has a least upper bound is a lattice where every pair of elements has a least upper bound. Vector lattices (see below) are the main mathematical structure of the Kozen's denotational semantics for probabilistic computation.

Definition 74 *Let* $\mathbf{V}$ *be a normed vector space and* $\mathbf{P} \subset \mathbf{V}$ *a positive cone,* $(\mathbf{V}, \mathbf{P})$ *is called a* vector lattice *iff every pair* $x, y \in \mathbf{V}$ *has a* $\sqsubseteq_{\mathbf{P}}$*-least upper bound in* $\mathbf{V}$*. A vector lattice is called* conditionally complete *if every set of elements of* $\mathbf{V}$ *with an* $\sqsubseteq_{\mathbf{P}}$*-upper bound has a least upper bound.*

To partially order a measurable space we will consider Banach lattices.

Definition 75 *Assume that* $\mathbf{B}$ *is a normed vector space with norm* $\|.\|$*, if* $(\mathbf{B}, \mathbf{P}, \|.\|)$ *is both a Banach space and vector lattice such that:*

$$\||x|\| = \|x\| \quad \textit{and} \quad \forall x, y \in \mathbf{P} : x \sqsubseteq_{\mathbf{P}} y \Rightarrow \|x\| \leq \|y\|\,,$$

then $\mathbf{B}$ *is called a* Banach lattice.

In the semantics introduced by Kozen for probabilistic computation, programs are interpreted as continuous linear operators on Banach space of distributions [74]. In this framework one could work only with the joint distribution of the program variables instead of dealing directly with variables. Any simple program P maps the input distributions μ to the output distribution $P(\mu)$. In his paper Kozen has considered a probabilistic WHILE programming language over the variables $x_1, \cdots, x_n$. Syntactically, there are five types of statements in the language described in the following (from [74]).

Core syntax of probabilistic WHILE:

- simple assignment: $x_i := f(x_1, \cdots, x_n)$, where $f : X^n \to X$ is a measurable function.
- random assignment: $x_i :=$ random.
- composition: $S; T$.
- conditional: if B then S else T.
- while loop: while B do S.

Let (X, M) be a measurable space and let $\mathbf{B} = \mathbf{B}(X^n, M^n)$ be the set of all measures on the cartesian product (X^n, M^n). Then $\mathbf{B}$ consists of all possible joint distributions of the program variables $x_1, x_2, \cdots, x_n$, plus all their linear combinations. Let $\mathbf{P}$ denote the set of all positive measures and $\|.\|$ to be the total variation norm then $(\mathbf{B}, \mathbf{P}, \|.\|)$ is a conditionally complete Banach lattice [74].

Every program P will map a probability distribution into a subprobability measure. This can be extended uniquely to a linear transformation in $\mathbf{B} \to \mathbf{B}$. Moreover, this extension will be $\|.\|$-bounded and therefore continuous. Thus, each program will define a continuous linear operator in $\mathbf{B} \to \mathbf{B}$ [74].

The space $\mathbf{B}'$ of operators in $\mathbf{B} \to \mathbf{B}$ forms a Banach space which is conditionally complete. The partial order on $\mathbf{B}'$ is defined as follows:

$$S \sqsubseteq T \;\text{ iff }\; S(\mu) \sqsubseteq T(\mu) \;\text{ for all }\; \mu \in \mathbf{P}\,.$$

Programs will be interpreted as elements of this space. In what follows, we present the semantics of the probabilistic WHILE programming language introduced above.

- *simple assignment:* If P is the program "$x_i := f(x_1, \cdots, x_n)$" where $f : X^n \to X$ is a measurable function, then the meaning of P, $[\![P]\!]$, is the linear operator $P : \mathbf{B} \to \mathbf{B}$ such that:

$$P(\mu) = \mu \circ F^{-1} ,$$

  where $F : X^n \to X^n$ is the measurable function

$$F(a_1, \cdots, a_n) = (a_1, \cdots, a_{i-1}, f(a_1, \cdots, a_n), a_{i+1}, \cdots a_n) .$$

  Since f is measurable, so is F, thus $\mu \circ F^{-1}$ is indeed a measure.

- *random assignment:* If P is the program "$x_i :=$ random" then the meaning of P, $[\![P]\!]$, is the linear operator $P : \mathbf{B} \to \mathbf{B}$ such that:

$$P(\mu)(B_1 \times \cdots \times B_n) = \mu(B_1 \times \cdots, B_i, X, B_{i+1}, \cdots B_n)\rho(B_i) ,$$

  where ρ is an arbitrary fixed distribution.

- *composition:* The meaning of the program "$S; T$" is the functional composition of operators $[\![T]\!] \circ [\![S]\!]$.

- *conditional:* Let μ_B denote the measure $\mu_B(A) = \mu(A \cap B)$. The conditional test checks the membership of $x_1, \cdots, x_n$ in B, which will occur with probability $\mu(B)$ and hence S will be executed on the conditional probability distribution $\mu_B/\mu(B)$. Similarly, with probability $\mu(\neg B)$ the program T will be executed on $\mu_{\neg B}/\mu(\neg B)$. Formally, the semantics of the program "if B then S else T" is the linear operator $P : \mathbf{B} \to \mathbf{B}$ such that:

$$\begin{aligned} A &\mapsto \mu(B)S(\mu_B/\mu(B))(A) + \mu(\neg B)T(\mu_{\neg B}/\mu(\neg B))(A) \\ &= (S(\mu_B) + T(\mu_{\neg B}))(A) , \end{aligned}$$

which can be written as $S \circ e_B + T \circ e_{\sim B}$ where e_B is the operator $e_B(\mu) = \mu_B$ and $+$ is addition in B'.

- *while loop:* The meaning of the program "while B do S" is equivalent to the program

$$\text{if } \neg B \text{ then } I \text{ else } S; \text{while } B \text{ do } S\,,$$

therefore the meaning of a "while statement" must be a solution of

$$W = e_{\sim B} + W \circ P \circ e_B\,.$$

Using well established techniques (which we will see later in this chapter) one can solve the above equation to derive the following solution. The meaning of a "while statement" is the fixed-point of the affine transformation $\tau : \mathbf{B}' \to \mathbf{B}'$ defined by

$$\tau(W) = e_{\sim B} + W \circ S \circ e_B\,,$$

which is equal to

$$\tau^n(0) = \sum_{0 \leq k \leq n-1} e_{\sim B} \circ (S \circ e_B)^k\,.$$

4.3 Quantum Setting

In this section we present some applications of domain theory in the framework of quantum computation. In the first subsection we study the domain computability for quantum computation. Subsequently a denotational semantics for quantum computation is given. Finally we review recent work on information aspects of quantum domain theory by Coecke and Martin [31]. By introducing a domain framework for quantum computation we aim to address different aspects of information processing which has not yet been studied in other existing models of quantum computation.

4.3.1 Computability Analysis

The Church-Turing thesis is about classical computability, (i.e. the computability which is defined based on a computing machine which obeys classical mechanics). Hence, it might be thought that quantum mechanical computing can violate the Church-Turing thesis. However, Deutsch [32] and the Jozsa [66] discussed this problem and showed that the class of functions computable by a deterministic quantum Turing machine is equal to the class of recursive functions (computable by a classical Turing machine). Ozawa extended this argument to the probabilistic quantum Turing machine [83]. He also distinguished the notation of measurability from computability to answer the problem that has been alleged by Nielsen in [78].

Apart from these few discussions, there have been no further attempts in this direction. We believe that, by introducing a rigourous framework for quantum computability, we can address more interesting questions. Furthermore, quantum domain theory provides us with a topological structure for quantum computation that can be useful for the study in other fields of quantum computation.

To develop a computational model to analyse quantum computability, it would be enough to consider a model for a Hilbert space. Different effective structures for metric spaces can be found in the literature. We use the domain of the closed balls [112, 39] to introduce a model for quantum pure states and the power domain of the former domain [64, 38, 76] will capture the quantum mixed states. It is important to emphasise main definitions and results of this subsection have already appeared in [39, 38] under the theory of computability for Metric spaces. We rephrase these results in order to suit our purposes of defining a mathematical foundation for quantum computability.

Pure quantum states

A standard way to construct a partially ordered set for a given metric space (X, d) is based on ordering of the set of closed balls [58]. Define a closed ball $C(x, r)$ of given metric space (X, d) with $x \in X$ and $r \in \mathbb{R}$ to be the following set:

$$C(x, r) = \{y \in X \mid d(x, y) \leq r\}.$$

The Hilbert space $\mathcal{H}$ of the quantum pure state is a metric space by virtue of the metric induced by the standard scalar product. Denote the poset of all closed balls of $\mathcal{H}$ by $C\mathcal{H}$ with the following partial order:

$$C(|\phi\rangle, r) \sqsubseteq C(|\psi\rangle, s) \quad \text{iff} \quad C(|\phi\rangle, r) \supseteq C(|\psi\rangle, s).$$

This relation reflects a natural notion of information: $C(|\phi\rangle, r) \sqsubseteq C(|\psi\rangle, s)$ can be read as the statement that $(|\phi\rangle, r)$ has less information than $(|\psi\rangle, s)$. The quantum pure state $|\phi\rangle \in \mathcal{H}$ can be identified with the maximal closed ball $C(|\phi\rangle, 0) \in C\mathcal{H}$, i.e. the maximal element of the poset $C\mathcal{H}$ is in one-to-one correspondence with $\mathcal{H}$. The following results from [39] prove that the poset $C\mathcal{H}$ has the required structure for the foundation of a computational model.

Theorem 76 *[39] Let B be a dense subset of a separable Hilbert space $\mathcal{H}$. Then $B \times \mathbb{Q}^+$ is a basis of $C\mathcal{H}$ where $\mathbb{Q}^+$ is the set of all non-negative rational numbers.*

There are many different choices for a dense subset of $\mathcal{H}$. Any universal set of quantum gates (Chapter 1, Subsection 1.3.2) provides us with a different dense subset of quantum states of a Hilbert space $\mathcal{H}$. To see this fact consider a discrete set of universal quantum gates, $\mathcal{S}$ (e.g. Hadamard + Phase + CNOT + $\pi/8$ Rotation), therefore any unitary operator on $\mathcal{H}$ can be approximated by a combination of elements in $\mathcal{S}$. In other word a universal set of gates is a dense subset of the set of all unitary operators on $\mathcal{H}$. Denote by $< \mathcal{S} >$ the set of all finite combinations of elements of $\mathcal{S}$. The following lemma gives a dense subset of $\mathcal{H}$.

Lemma 77 *The image of $<\mathcal{S}>$ on state $|0\rangle \in \mathcal{H}$ is a dense subset of $\mathcal{H}$.*

Proof Assume that $|\psi\rangle$ is an arbitrary quantum state in $\mathcal{H}$. Consider a unitary operator U such that $U|0\rangle = |\psi\rangle$. From the universality of $\mathcal{S}$ we derive that U can be approximated by a sequence of elements, $V_1, V_2, \cdots, V_n$ in $\mathcal{S}$. The following sequence of the states

$$V_1|0\rangle, V_2|0\rangle, \cdots, V_n|0\rangle\,,$$

belongs to the image of $< \mathcal{S} >$ on $|0\rangle$ and approximates $|\psi\rangle$. This finishes the proof.□

Theorem 78 *[39] The poset of the closed balls of a separable Hilbert space, ordered by reversed inclusion, is an ω-continuous domain.*

It is easy to see that the way-below relation is nothing but

$$C(|\phi\rangle, r) \ll (|\psi\rangle, s) \quad \text{iff} \quad C(|\phi\rangle, r) \supset C(|\psi\rangle, s).$$

The embedding of $\mathcal{H}$ into $C\mathcal{H}$ is defined with the following function:

$$\begin{aligned} e_P : \mathcal{H} &\rightarrow C\mathcal{H} \\ |\phi\rangle &\mapsto (|\phi\rangle, 0)\,. \end{aligned}$$

Clearly, the elements of $C\mathcal{H}^+ = \{(|\phi\rangle, 0) \,|\, |\phi\rangle \in \mathcal{H}\}$ are the maximal elements of $C\mathcal{H}$. Following the definitions of Subsection 4.2, we can introduce a topological structure for $C\mathcal{H}$. It is easy to check that for any given element $(|\phi\rangle, r) \in C\mathcal{H}$ we have:

$$e_P^{-1}(\Uparrow(|\phi\rangle, r)) = O(|\phi\rangle, r)\,,$$

where $O(|\phi\rangle, r)$ is the open ball with centre $|\psi\rangle$ and radius r. The subsets $\Uparrow(|\phi\rangle, r)$ form a basis for the Scott topology on $C\mathcal{H}$, while the open balls $O(|\phi\rangle, r)$ are a basis for metric topology on $\mathcal{H}$. Hence, e_P is a topological embedding, which makes $\mathcal{H}$ homomorphic to the subspace of maximal elements of $C\mathcal{H}$.

The ω-continuity of $C\mathcal{H}$ introduces an effective structure along the lines of Subsection 4.2.1. The homomorphism between $\mathcal{H}$ and maximal elements of $C\mathcal{H}$ derives an effective structure for $\mathcal{H}$ and hence it provides a computational framework for $\mathcal{H}$. In a similar way to the Subsection 4.2.1 we can define a computable pure state as follows.

Definition 79 *A quantum pure state $|\psi\rangle$ is called* computable*, if its domain image $e_P(|\psi\rangle) = (|\psi\rangle, 0)$ is computable in $C\mathcal{H}$, i.e. iff the set $\{n \in \mathbb{N} \,|\, b_n \ll (|\psi\rangle, 0)\}$ is r.e. (where $\{b_n\}$ are elements of the basis $\mathcal{B}_{C\mathcal{H}}$).*

Mixed quantum states

As we explained in Chapter 1, there exists a correspondence between density matrices and probability measures on $\mathcal{H}$ (Theorem 12). Therefore, to present a computational framework for mixed states, it is enough to construct such a framework for probability measures on $\mathcal{H}$. We use the following notations and results from [56, 38, 5, 76].

The domain of probability measures will be defined in terms of continuous valuation functions. A continuous valuation function is a finite measure which is defined on open subsets of a topological space [16, 57, 38].

Definition 80 *Assume that X is a topological space. A function ν from open sets of X to non-negative real number, $\mathbb{R}^+$, is called a* continuous valuation *function iff the following conditions are satisfied:*

- *Strictness.* $\nu(\emptyset) = 0$*;*
- *Monotonicity.* $A \subseteq B \Rightarrow \nu(A) \leq \nu(B)$*;*
- *Modularity.* $\nu(A \cup B) + \nu(A \cap B) = \nu(A) + \nu(B)$*;*
- *Continuity. whenever $\mathcal{I}$ is a directed subset of open sets (with respect to $\subseteq$),* $\nu(\bigcup \mathcal{I}) = \sup_{A \in \mathcal{I}} \nu(A)$.

A continuous valuation on an ω-continuous domain is a continuous valuation on its Scott topology.

Definition 81 *[64] Assume that X is a topological space. The* probabilistic power domain *PX of X consists of all continuous valuations ν on X with $\nu(X) \leq 1$, ordered pointwise, i.e.*

$$\mu \sqsubseteq \nu \quad \textit{iff} \quad \mu(O) \leq \nu(O) \textit{ for all open sets in } X\,.$$

The simple valuation functions provide a basis for the probabilistic power domain.

Definition 82 *[64] For any point $x \in X$ the* point valuation, *δ_x, is defined as follows:*

$$\delta_x(O) = \begin{cases} 1 & \text{if } x \in O \\ 0 & \text{if } x \notin O \end{cases}$$

A finite linear combination of point valuations i.e. $\sum_{i=1}^{n} r_i \delta_{x_i}$ with $x_i \in X$ and positive rational numbers r_i satisfying $\sum_{i=1}^{n} r_i \leq 1$, is called a simple valuation.

Theorem 83 *[64] The probabilistic power domain of an ω-continuous domain is also ω-continuous with a basis of simple valuation.*

Now, we can introduce the domain of quantum mixed states. The set of all closed subspaces of $\mathcal{H}$ is the σ-algebra, $\mathcal{M}$, of the measurable sets. Let $\mathbf{M}(\mathcal{H})$ denote the set of all probability measures on $\mathcal{H}$. Based on Gleason's Theorem (Chapter 1, Theorem 12), a mixed state can be considered to be an element of $\mathbf{M}(\mathcal{H})$. We embed $\mathbf{M}(\mathcal{H})$ into the probabilistic power domain $PC\mathcal{H}$ of the closed ball domain $C\mathcal{H}$, which forms an ω-continuous domain.

The maximal element of $PC\mathcal{H}$ is the set of all valuations ν such that:

$$\nu(O) = 1 \quad \text{for all open subsets} \quad O \in C\mathcal{H}^{+} .$$

The embedding of $\mathbf{M}(\mathcal{H})$ into $PC\mathcal{H}$ is defined with the following function:

$$\begin{aligned} e_M : \mathbf{M}(\mathcal{H}) &\rightarrow PC\mathcal{H} \\ \mu &\mapsto \mu \circ e_P^{-1} . \end{aligned}$$

The following result from [38] provides the correspondence between $\mathbf{M}(\mathcal{H})$ and $PC\mathcal{H}^{+}$:

Theorem 84 *[38] The space $\mathbf{M}(\mathcal{H})$ is homomorphic with the space of maximal elements of the ω-continuous domain $PC\mathcal{H}$. These maximal elements are characterised by $\nu(C\mathcal{H}^{+}) = 1$. Every mixed state on $\mathcal{H}$ can be obtained via this homomorphism as the least upper bound of an increasing chain of simple valuations on $C\mathcal{H}$.*

Similarly to the case of pure states, we define the computability of a mixed state via the computational framework of $PC\mathcal{H}$.

Definition 85 *A quantum mixed state ρ is called* computable, *if its corresponding measure $\mu_\rho \in \mathbf{M}(\mathcal{H})$ is computable i.e. if the domain element $e_M(\mu)$ is computable in $PC\mathcal{H}$, i.e. iff the set $\{n \in \mathbb{N} \mid b_n \ll e_M(\mu)\}$ is r.e. (where $\{b_n\}$ are elements of the basis $\mathcal{B}_{PC\mathcal{H}}$).*

The process of quantum computation over a pure state is described with a unitary operator, and over a mixed state is described with a CP map. As we explained before, in a domain-picture of computation, programs are functions from the domain of the input to the domain of the output. The set of all continuous functions forms the domain of operators. This is exactly the same in the case of quantum computation.

Unitary Operators

Following the notation of Subsection 4.2 we denote by $[C\mathcal{H} \to C\mathcal{H}]$, the domain of the all continuous functions on $C\mathcal{H}$ with pointwise ordering. Every unitary operator $U : \mathcal{H} \to \mathcal{H}$ has a Scott-continuous extension to the domain of the closed ball $C\mathcal{H}$, i.e. there exists a Scott-continuous function $\tilde{U}$ in $[C\mathcal{H} \to C\mathcal{H}]$ such that

$$\tilde{U}(C(|\phi\rangle, 0)) = C(U|\phi\rangle, 0) \quad \text{for all} \quad |\phi\rangle \in \mathcal{H}\,,$$

and it is explicitly given by

$$\tilde{U}(C(|\phi\rangle, r)) = C(U|\phi\rangle, r)\,.$$

The following lemma shows that the $\tilde{U}$ is well-defined.

Lemma 86 *Let U to be a unitary operator on $\mathcal{H}$. The extension function $\tilde{U}$ (defined above), maps a closed ball in $C\mathcal{H}$ to another closed ball.*

Proof Assume the closed ball $C(|\phi\rangle, r)$ is given, then by applying $\tilde{U}$ we obtain the

following set:

$$\tilde{U}(C(|\phi\rangle, r)) = \{U|\psi\rangle \ : \ |\psi\rangle \in C(|\phi\rangle, r)\}\,.$$

U is a unitary operator therefore the angle and distance between vectors in $C(|\phi\rangle, r)$ and also their length under above transformation do not change and hence the resulting set is another closed ball in $C\mathcal{H}$. □

We define the computability of a unitary function via domain theory with:

Definition 87 *A unitary function $U : \mathcal{H} \to \mathcal{H}$ is* computable *iff its extension, $\tilde{U}$, is computable in $U : \mathcal{H} \to \mathcal{H}$ (in the terms of Subsection 4.2.1).*

CP Maps

For simplicity, we denote by T_μ the corresponding operator for a given measure μ which is derived from Theorem 12:

$$\forall \mu \ \ \exists T : \mu(A) = \mathrm{Tr}(TP_A) \ \text{ for all closed subspaces } \ A\,.$$

A CP map A is an operator over $B(\mathcal{H})$ and can also be considered as a function in $\mathbf{M}(\mathcal{H}) \to \mathbf{M}(\mathcal{H})$ (from Gleason's Theorem). Denote by $[PC\mathcal{H} \to PC\mathcal{H}]$ the domain of all continuous functions on $PC\mathcal{H}$ (with pointwise ordering).

Every CP map $\mathrm{A} : \mathbf{M}(\mathcal{H}) \to \mathbf{M}(\mathcal{H})$ has a Scott-continuous extension to the domain of $[PC\mathcal{H} \to PC\mathcal{H}]$, i.e. there exists a Scott-continuous function $\tilde{\mathrm{A}} \in [PC\mathcal{H} \to PC\mathcal{H}]$ such that for every probability measure $\mu \in \mathbf{M}(\mathcal{H})$ we have:

$$\tilde{\mathrm{A}}(\mu) = \mathrm{A}(\mu)\,.$$

The extension function $\tilde{\mathrm{A}}$ for a continuous valuation function $\nu \in PC\mathcal{H}$ is explicitly given as follows. For a Scott open subset O in $C\mathcal{H}$ define:

$$\tilde{\mathrm{A}}(\nu)(O) = \mathrm{Tr}(T_{\mathrm{A}(\nu)}P_{<O>})\,,$$

where $< O >$ is the closed subspace of $\mathcal{H}$ spanned by vectors in O. The following lemma shows that the above definition is well-defined:

Lemma 88 *Any CP map* A *on* $\mathbf{M}(\mathcal{H})$*, maps a continuous valuation function to another continuous valuation function.*

Proof Assume ν is a continuous valuation, we show $\mathrm{A}(\nu)$ is also a continuous valuation.

- Strictness. $\mathrm{A}(\nu)(\emptyset) = \mathrm{Tr}(T_{\mathrm{A}(\nu)} P_{<\emptyset>}) = 0\,.$
- Monotonicity. Let $O \subseteq O'$ then

$$\begin{aligned} \mathrm{A}(\nu)(O) &= \mathrm{Tr}(T_{\mathrm{A}(\nu)} P_{<O>}) \\ &\leq \mathrm{Tr}(T_{\mathrm{A}(\nu)} P_{<O'>}) \\ &= \mathrm{A}(\nu)(O')\,. \end{aligned}$$

- Modularity.

$$\begin{aligned} \mathrm{A}(\nu)(O \cup O') &+ \mathrm{A}(\nu)(O \cap O') \\ &= \mathrm{Tr}(T_{\mathrm{A}(\nu)} P_{<O \cup O'>}) + \mathrm{Tr}(T_{\mathrm{A}(\nu)} P_{<O \cap O'>}) \\ &= \mathrm{Tr}(T_{\mathrm{A}(\nu)} P_{<O>}) + \mathrm{Tr}(T_{\mathrm{A}(\nu)} P_{<O'>}) \\ &= \mathrm{A}(\nu)(O) + \mathrm{A}(\nu)(O')\,. \end{aligned}$$

- Continuity. Let $\mathcal{I}$ be a directed subset of open sets

$$\begin{aligned} \mathrm{A}(\nu)(\bigcup \mathcal{I}) &= \mathrm{Tr}(T_{\mathrm{A}(\nu)} P_{<\mathcal{I}>}) \\ &= \sup_{O \in \mathcal{I}} \mathrm{Tr}(T_{\mathrm{A}(\nu)} P_{<O>}) \\ &= \sup_{O \in \mathcal{I}} \mathrm{A}(\nu)(O)\,. \end{aligned}$$

□

We define the computability of a CP map function via domain theory with:

Definition 89 *A CP map* $\mathrm{A} : \mathbf{M}(\mathcal{H}) \to \mathbf{M}(\mathcal{H})$ *is computable iff its extension,* $\tilde{\mathrm{A}}$*, is computable in* $[PC\mathcal{H} \to PC\mathcal{H}]$.

Quantum Measurements

At the end of a computation a measurement operator will be applied. A measurement can be viewed as a CP map which takes a density matrix (the final state) to another density matrix (the probabilistic mixture of the outcomes).

Assume M_m is a collection of measurement operators. The corresponding measurement of this collection can be considered as a CP map over $B(\mathcal{H})$:

$$\begin{array}{rcl} \mathrm{M} : B(\mathcal{H}) & \to & B(\mathcal{H}) \\ \rho & \mapsto & \dfrac{M_m \rho M_m^\dagger}{\mathrm{Tr}(M_m^\dagger M_m \rho)} \, . \end{array}$$

Hence, the extension function and computability can be also defined exactly in the same way that we defined before for a given CP map.

4.3.2 Denotational Semantics

In this subsection we present a denotational semantics for quantum computation using domain theory, which could be considered as a foundation for designing a functional programming language for quantum computation. The recent literature contains several proposals for quantum programming languages. The first contribution in this direction is Knill's paper on the QRAM model [41]. The other attempts to define a true quantum programming language are two imperative languages. The first approach by Ömer [81, 82] has a C-like syntax, while a second proposal by Sanders and Zuliani [91] is based on Dijkstra's guarded-command language. A similar approach to the work of this subsection has been developed independently by Selinger [95]. He has presented the first functional programming language and discussed the denotational semantics of his proposed language. Our work is based on the Kozen's semantics for probabilistic computation [74].

We aim to develop a denotational semantics for a basic programming language, called Quantum WHILE. In this approach, we show how to define the mathemat-

ical object corresponding to the language constructors. We will consider a simple quantum computational machine with quantum memory registers. To develop the proper foundation for quantum semantics, in the most general setting, we consider density matrices and CP maps. Aharanov, Kitaev and Nisan in [4] introduced the first computational model based on mixed state where possible operators are represented by CP maps. We show in this subsection that the same structure of the classical probabilistic semantics which has been introduced by Kozen [74] can also capture the semantics of quantum computation.

To follow the procedure introduced by Kozen [74], we define a measurable space $(\mathcal{X}^n, \mathcal{M}^n)$ with the set of all measures $\mathbf{B} = \mathbf{B}(\mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_n, \mathcal{M}^n)$ such that the set of all probability measures in this space is in correspondence with the set of all density matrices over $\mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_n$. In this way, input to a quantum program P is represented by a probability measure $\rho \in \mathbf{B}$ which is the same as the corresponding density matrix of all input pure states $|\phi_1\rangle \otimes \cdots \otimes |\phi_n\rangle$ in $\mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_n$.

Let $\mathcal{H} = \mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_n$ denote the Hilbert space spanned by all the quantum variables which are involved in the computation. Define $\mathcal{X}_i$ to be the set of all unit vectors in $\mathcal{H}_i$ and $\mathcal{X}$ to be the set of all unit vectors in $\mathcal{H}$. The set of all closed subspaces of $\mathcal{H}$ is the σ-algebra, $\mathcal{M}$, of the measurable sets. Gleason's Theorem determines all measures on $\mathcal{M}$ (Chapter 1, Theorem 12) and shows a correspondence between operators in $\mathrm{B}(\mathcal{H})$ and measures on $\mathcal{M}^n$, we use interchangeably any of the two notions of measure and operator. In the same way as the classical case, the set of positive measures (positive self-adjoint operators) $\mathbf{P} \subset \mathbf{B}$ is the positive cone of the measure space $\mathbf{B}$. The definition of ordering of measures is defined as follows

$$\mu \sqsubseteq \nu \;\text{ iff }\; \nu - \mu \in \mathbf{P}\,.$$

The space $\mathbf{B}'$ of all CP maps in $\mathrm{B}(\mathcal{H}) \to \mathrm{B}(\mathcal{H})$ forms a Banach space (under the same definition of Subsection 4.2.2). The partial ordering of the set of all CP maps is defined as follows:

$$A \leq B \;\text{ iff }\; A(\rho) \leq B(\rho) \quad \text{for all} \quad \rho \in \mathbf{P}\,.$$

In semantics of the general setting for quantum computation, each program will be represented by a CP map. For simplicity, in what follows the symbol P refers to both a program and the corresponding CP map. A quantum program P maps distributions μ on $(\mathcal{X}, \mathcal{M})$ to distribution $P(\mu)$ on $(\mathcal{X}, \mathcal{M})$, or equivalently, maps a density matrix μ on $\mathcal{H}$ to the density matrix $P(\mu)$.

For the completeness of the discussion we will give the full semantics of the quantum WHILE language in the general setting. The syntax of this Language is the same as the syntax of the classical probabilistic WHILE language (Subsection 4.2.2). The only difference is that instead of "random assignment" we have "quantum measurement".

- *simple assignment*: If P is the program "$x_i := f(x_1, \cdots, x_n)$" where $f : \mathcal{X} \to \mathcal{X}_i$ is a measurable function, then the meaning of P is the following CP map:

$$\begin{aligned} \mu &\mapsto P(\mu) \\ P(\mu) &= \mu \circ F^{-1}, \end{aligned}$$

  where $F : \mathcal{X} \to \mathcal{X}$ is the measurable function

$$F(a_1, \cdots, a_n) = (a_1, \cdots, a_i, f(a_1, \cdots, a_n), a_{i+1}, \cdots a_n).$$

- *measurement assignment*: If P is the program "$x_i :=$ measure" then the meaning of P is the following CP map:

$$\begin{aligned} \mu &\mapsto P(\mu) \\ P(\mu)(A) &= \mathrm{Tr}(\rho' P_A). \end{aligned}$$

  where ρ' is a fixed distribution (density matrix) corresponding to the measurement process. To be more precise, assume that the collection $\{M_m\}$ describes the quantum measurement that has been applied on the ith variable, then

$$\rho' = \frac{(I \otimes \cdots \otimes M_m \otimes \cdots \otimes I)\, T_\mu\, (I \otimes \cdots \otimes M_m^\dagger \otimes \cdots \otimes I)}{\mathrm{Tr}((I \otimes \cdots \otimes M_m^\dagger \otimes \cdots \otimes I)\, (I \otimes \cdots \otimes M_m \otimes \cdots \otimes I)\, T\mu)}.$$

- *composition*: The meaning of the program "$S; T$" is the functional composition of the CP maps T and S, $T \circ S$.

- *conditional*: The semantics of the program "if B then S else T" is the CP map

$$S \circ e_B + T \circ e_{\sim B} ,$$

  where e_B is the CP map $e_B(\mu) = \mu_B$.

- *while loop*: The meaning of the program "while B do S" is the fixed-point of the affine transformation $\tau : \mathbf{B}' \to \mathbf{B}'$ defined by

$$\tau(W) = e_{\sim B} + W \circ S \circ e_B ,$$

  which is equal to

$$\tau^n(0) = \sum_{0 \leq k \leq n-1} e_{\sim B} \circ (S \circ e_B)^k .$$

In order to present a complete picture of the applications of quantum domain theory, in the next section we briefly review a domain framework for information theory.

4.4 Information Theory

Recently a new application of domain theory has been introduced by Coecke and Martin [31]. One of their main results was to show a domain formulation of existing results from information theory. They have shown the Shannon entropy and Von Neumann entropy can be captured as Scott continuous functions over the corresponding domain. Here we briefly review their work in order to give a complete picture of quantum domain theory. All the definitions and results in this subsection are taken from [31].

Coecke and Martin have constructed a domain structure over mixed states such that pure sates are the maximal elements. They first order classical states recursively

in terms of Bayesian order.

Definition 90 *Let $n \geq 2$. The* classical states *are*

$$\Delta^n = \left\{ x \in [0,1]^n \mid \sum_{i=1}^{n} x_i = 1 \right\}$$

A classical state $x \in \Delta^n$ is pure, *when $x_i = 1$ for some $i \in 1, \cdots, n$. Denote by $\{e_i \mid i = 1, \cdots, n\}$ the set of all pure states.*

A classical state in $x \in \Delta^n$ can be interpreted as the information that an observer has about the results of an event in which n different outcomes are possible i.e. x_i indicates the probability of obtaining the outcome i. If we know x and after measuring we determine that outcome i is not possible, our knowledge improves to

$$p_i(x) = \frac{1}{1 - x_i}(x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n) \in \Delta^{n-1}\,,$$

where $p_i(x)$ is obtained first by removing x_i from x and then reorganising. The partial functions p_i:

$$p_i : \Delta^n \rightharpoonup \Delta^{n-1}\,,$$

with $\mathrm{Dom}(p_i) = \Delta^n \setminus e_i$, are called the *Bayesian projections*. The classical states are partially ordered with the following recursive relation.

Definition 91 *Assume that x and y are in Δ^n; we write $x \sqsubseteq_B y$ iff:*

$$\forall i : \quad x, y \in \mathrm{Dom}(p_i) \Rightarrow p_i(x) \sqsubseteq_B p_i(y)\,.$$

For $x, y \in \Delta^2$ we have:

$$x \sqsubseteq_B y \;\; \textit{iff} \;\; (y_1 \leq x_1 \leq 1/2) \; \textit{or} \; (1/2 \leq x_1 \leq y_1)\,.$$

The above relation is called the Bayesian order.

The Bayesian order leads to a domain of classical states where the pure states are the maximal elements.

Theorem 92 *[31]* $(\Delta^n, \sqsubseteq_B)$ *is a domain with the following set of maximal elements:*

$$\{e_i \,|\, 1 \leq i \leq n\}\,,$$

and least element $\perp = (1/n, \cdots, 1/n)$.

Coecke and Martin have generalised the idea of the Bayesian order to the quantum setting using the spectral order. Informally speaking, to compare the amount of information of two given mixed states it is enough to consider an observable and measure both mixed states. The result of measurements are two classical states and can be ordered via the Bayesian order. Following the notation of [31], we denote by Ω^n the set of all density matrices on $\mathcal{H}^n$. For simplicity we also consider the following definition.

Definition 93 *Assume that* O *is a non-degenerate observable on* $\mathcal{H}^n$ *i.e. it has* n *different eigenvalues with orthogonal eigenvector spaces* $\{P_i\}_{i=1}^n$. *For a density matrix* ρ *on* $\mathcal{H}^n$ *we define:*

$$\mathrm{Spec}(\rho|O) = (\mathrm{Tr}(P_1 \cdot \rho), \cdots, \mathrm{Tr}(P_n \cdot \rho)) \in \Delta^n\,.$$

Definition 94 *Let* $n \geq 2$*, for quantum states* $\rho, \sigma \in \Omega^n$*, we have* $\rho \sqsubseteq_S \sigma$ *iff there exists a non-degenerate observable* $O : \mathcal{H}^n \to \mathcal{H}^n$ *such that* $[\rho, O] = [\sigma, O] = 0$ *and*

$$\mathrm{Spec}(\rho|O) \sqsubseteq_B \mathrm{Spec}(\sigma|O)\,.$$

This is called the spectral order.

Finally the domain of the quantum states can be defined with:

Theorem 95 *[31]* $(\Omega^n, \sqsubseteq_S)$ *is a domain with the following set of pure states as the maximal elements and least element* $\perp = I/n$*, where* I *is the identity matrix.*

The final part of the Coecke and Martin's work that we review here is concerned with measuring of information content. To this end we need the following definition from [31].

Definition 96 *A Scott continuous map $\mu : D \to [0,\infty)^{*1}$ on a domain is said to* measure the content of $x \in D$ *if*

$$x \in U \quad \Rightarrow \quad (\exists \epsilon > 0)\; x \in \mu_\epsilon(x) \subset U\,,$$

whenever U is Scott open in D and

$$\mu_\epsilon(x) = \{y \in D | y \sqsubseteq x \;\&\; |\mu(x) - \mu(y)| < \epsilon\}\,.$$

The map μ measures *X if it measures the content of each $x \in X$.*

A map μ is a measure of content if it distinguishes the maximal (in content) elements.

Definition 97 *A* measurement *is a Scott continuous map $\mu : D \to [0,\infty)^*$ on a domain if it measures the set $\{x \in D \,|\, \mu(x) = 0\}$.*

The following results from [31] present the domain picture of the well-known functions, the Shannon entropy and the von Neumann entropy. As we will discuss in the next chapter this can provide us with a uniform framework for measuring the entanglement.

Theorem 98 *[31] Shannon entropy*

$$\mu(x) = -\sum_{i=1}^{n} x_i \log(x_i)\,,$$

is a measurement of type $\Delta^n \to [0,\infty)^*$.

Von Neumann entropy

$$\sigma(\rho) = -\mathrm{Tr}(\rho \lg(\rho)\,,$$

[1] The set $[0,\infty)^*$ is the domain of nonnegative real numbers in their opposite order.

is a measurement of type $\Omega^n \to [0, \infty)^*$.

4.5 Discussions

In this chapter, we have discussed a new framework for quantum computation via quantum domain theory. Using domain theory a rigourous framework for quantum computability has been introduced. Although it is known that the class of quantum computable functions is same as the class of classical computable functions (Church-Turing Principle [32, 66, 83]), we believe that by considering a proper framework for quantum computability we may be able to address new and interesting questions. We also presented a topological structure for quantum computation using domain theory, which may prove to be useful in other aspects of theoretical quantum computation. Furthermore we introduced a denotational semantics for quantum computation and we showed that quantum computation over density matrices with completely positive maps, has a similar semantical structure as probabilistic computation over random variables. This could be considered as a foundation for designing a functional programming language for quantum computation. Finally we reviewed a domain structure for quantum information theory where the proposed partial order has interesting connections with theory of entanglement [31]. We believe a domain theoretical approach to the theory of entanglement manipulation may provide us with a uniform framework for measuring entanglement.

5

Axiomatic Information Theory

5.1 Introduction

In this chapter we will present a general mathematical formalism which, we believe, can describe key information processing aspects based on different physical theories. This formulation was originally introduced by Giles for the purpose of giving a rigourous mathematical framework for classical thermodynamics [49]. In particular, he wanted to formalise the statement of the Second Law of thermodynamics and derive a unique quantity, called entropy, which orders thermodynamical states according to their mutual accessability.

There are many different ways of stating the Second Law. Caratheodory [24] restated the Second Law by saying that in the neighbourhood of any state there exist states which are adiabatically inaccessible from it. This allowed him to derive an entropy function which is able to introduce ordering into the set of physical states. The Second Law thus tells us that adiabatic processes cannot decrease entropy of the system itself. The question, then, is whether entropy is the only such function. To answer this question, however, thermodynamics needed first to be put onto a more secure mathematical foundation. In the words of Caratheodory himself:

"*What Thermodynamics needs is the establishment of logical order, essentially an*

intellectual cleanup. This is a problem for a mathematician. The fundamental ideas and concepts have been introduced by physicists long ago and a mathematician need not worry about it".

The first such formalisation came from Giles in 1964 [49] and was recently extended by Lieb and Yngvson [75]. Giles' mathematical formalism can be presented in a way which is completely divorced from any underlying physical basis, and this is what potentially allows us to apply it to scenarios other than thermodynamics. We first review Giles' original application to thermodynamics, and then show that the same formalism can be interpreted to capture entanglement manipulations in quantum information processing [107, 106]. This allows us to prove in a novel way the uniqueness of the measure of entanglement for pure bipartite states. Since classical information processing can be seen as a special case of quantum information processing, this formalism will also allow us to derive the classical (Shannon) entropy [96] within the same framework, but from the dynamical perspective. This highlights the close relationship between information processing and statistical mechanics in general as we discuss at the end of the chapter, along with other open problems in this direction.

5.2 Formal Theory

An ideal physical theory should consist of two independent parts: a mathematical theory and a set of rules of interpretation of various mathematical objects involved in the theory. By formalising a physical theory in such a way as to divorce it from the physical interpretation, it is possible to derive a mathematical structure that may be useful in a completely different physical setting to the original one. We give a brief summary of Giles' mathematical theory [49] and then describe how one can capture thermodynamics and entanglement manipulations with the axioms introduced in this section. The physical motivation behind the axioms will then become clearer.

We study a non-empty set $\mathcal{S}$, whose elements are called states, in which two operations, $+$ and $\rightarrow$, are defined. The goal is to derive a unique ordering over the states with some particular conditions. In what follows, states are denoted by

$a, b, c, \ldots$:

Axioms 1-5

(1) The operation $+$ is associative and commutative.

(2) For all state a we have: $a \rightarrow a$.

(3) For all states a, b and c we have: $a \rightarrow b \ \& \ b \rightarrow c \Longrightarrow a \rightarrow c$.

(4) For all states a, b and c we have: $a \rightarrow b \Longleftrightarrow a + c \rightarrow b + c$.

(5) For all states a, b and c we have: $a \rightarrow b \ \& \ a \rightarrow c \Longrightarrow b \rightarrow c$ *or* $c \rightarrow b$.

Definition 99 *A* process *is an ordered pair of states* (a, b). *The set of all processes is denoted by* $\mathcal{P}$.

We extend definitions of $+$ and $\rightarrow$ to $\mathcal{P}$ as follows :

$$\begin{aligned} (a,b) \quad &+ \quad (c,d) = (a+c, b+d) \\ (a,b) \quad &\rightarrow \quad (c,d) \Longleftrightarrow a+d \rightarrow b+c\,. \end{aligned}$$

For simplicity, we also define a relation $\subset$ over $\mathcal{S}$.

Definition 100 *Given states* a *and* b *we write* $a \subset b$ *(and say that* a *is* contained *in* b*), if there exists a positive integer* n *and a state* c *such that*

$$na + c \rightarrow nb \ \textit{ or } \ nb \rightarrow na + c \ .$$

Informally, the above definition means that a state a is smaller than b if a requires the help of another state c to be converted to or derived from b. Now we can introduce an important class of states, internal states, which server as yardstick for ordering the states.

Definition 101 *A state* e *is an* internal *state if, given any state* x*, there exists a positive integer* n *such that* $x \subset ne$.

This definition introduces a reference state, which is the one that can contain any other physical state given sufficiently many copies of it. The concept of the internal state is necessary to give a basic metric unit to quantify the physical content of a state in a unique way (i.e. independent of states). The other important classes of states are the sets of equilibrium and anti-equilibrium states.

Definition 102 *A state a is an* equilibrium *state if there exists no state b such that $a \rightarrow b$ and $b \not\rightarrow a$. A state x is an anti-equilibrium state if there exists no state a such that $a \rightarrow x$ and $x \not\rightarrow a$.*

Now we can present the remaining axioms of the formal theory:

Axioms 6-8

(6) There exists an internal state.

(7) Given a process α, if there exists a state c such that for any positive real number ϵ there exists positive integers m, n and states x, y such that $m/n < \epsilon$, $x \subset mc$,$y \subset mc$, and $(x, y) + n\alpha \rightarrow 0$ then $\alpha \rightarrow 0$.

(8) Given a state a, there exists an anti-equilibrium state x such that $x \rightarrow a$. If x and y anti-equilibrium states then so $x + y$.

To order the states we define the following function.

Definition 103 *A real-valued function, E, is an* entropy *function, if it satisfies the following properties for all states a and b:*

(i) $E(a + b) = E(a) + E(b)$.

(ii) $a \rightarrow b \;\&\; b \rightarrow a \iff E(a) = E(b$.

(iii) $a \rightarrow b \;\&\; b \not\rightarrow a \iff E(a) < E(b)$.

(iv) For every anti-equilibrium state x we have $E(x) = 0$.

To present the uniqueness theorem for entropy functions we need to define the class of following functions. This notion is important as the entropy function will be unique up to the addition of this function.

Definition 104 *A real-valued function Q is a* component of content *function if for all states a and b we have:*

- $Q(a+b) = Q(a) + Q(b)$.
- $a \to b \Longrightarrow Q(a) = Q(b)$.

Definition 105 *A real-valued function Q is a* non-equilibrium component of content *if it satisfies the following properties:*

(i) Q is a component of content function.

(ii) For every anti-equilibrium state x we have $Q(x) = 0$.

Theorem 106 *[49] Let E_1 be an entropy function. If Q is a non-equilibrium component of content and λ a positive real number then $\lambda E_1 + Q$ is an entropy function. Moreover, any entropy function E may be written in this form.*

This theorem, proved by Giles, states that the measure of order is unique up to an affine transformation. With this we complete the formal part of the theory and turn our attention to the applications. Note that the main point of the formalism is to give conditions under which we can uniquely order states of a certain set.

5.3 Thermodynamics

In this section, we briefly discuss how the formal model introduced in Section 5.2 describes the structure of thermodynamics. Consider $\mathcal{S}$ to be the set of all thermodynamical states (e.g. a state of a simple gas is defined once its temperature T and volume V are known, therefore we can say that $a = (T_a, V_a)$). The operator $+$ represents the physical operation of considering two systems together. Therefore it must naturally be associative and commutative. On the other hand, the operator $\to$ represents an adiabatic process which is meant to convert different physical states into each other. Therefore, like any other physical process, it should naturally be reflective and transitive as in axioms 2 and 3.

Axiom 4 is the first non-intuitive property linking the operators $+$ and $\rightarrow$. In the forward direction it is clear that if state a can be converted into b then the presence of another state c should not alter this fact, i.e. we can convert a and c into b and c by converting a into b while doing nothing to c. In the backward direction, however, this axiom is not completely obvious. It says that if a process is possible with the aid of another state, then we, in fact, do not need this other state for the process. Thermodynamics deals with macroscopic systems with a large number of degrees of freedom (subsystems). It is in the "asymptotic" limit that this axiom becomes more natural.

Finally, axiom 5 is the key property which allows us to compare different states and processes. It says that any two states that are accessible from a third state must be accessible to each other at least in one direction. Not being able to do so would lead to states which would be incomparable as there would be no physical way of connecting them. Thus, a unique way of ordering states would be impossible.

Axiom 6 is necessary if we are to compare contents of different states in a unique way. Axiom 7 is the most complex axiom in the theory, although it is strongly motivated by the logic of thermodynamical reasoning. Loosely speaking, it states that if we can transform a into b with an arbitrarily small environmental influence, then this influence can be ignored. This, in some sense, introduces continuity into thermodynamical properties. Axiom 8 is self evident in thermodynamics. A more detailed on physical interpretation of the axioms can be found in Giles' book [49].

5.4 Entanglement Manipulation

Understanding of entanglement and its characterisation form the cornerstone of the new and rapidly growing filed of quantum information and computation [67, 105]. We need to know how much entanglement is at our disposal since entanglement is a form of resource that can enhance information processing [86].

Although a great deal of work has recently been performed in this direction [61], it is widely acknowledged that we do not have a complete understanding of even the bipartite entanglement for mixed states. There is a number of measures to quantify entanglement which apply in different settings and have different prop-

erties [61]. The consensus, however, is that local operations aided with classical communication (LOCC) are the key to explaining entanglement [10, 87, 102].

The LOCC maps separate disentangled states from entangled states and thus introduce a *directionality* to entanglement manipulation processes: an entangled state can always be converted to a disentangled one by LOCC, but not vice versa.

A comparison with thermodynamics will be very helpful at this point. The Second Law of thermodynamics tells us which (energy conserving) processes are allowed in nature, without any reference to the underlying physical structure. The central role is played by adiabatic processes and entropy is used to separate the possible from the impossible processes according to a very simple principle: if a state A has more entropy than B, then there is an adiabatic process to go from B to A, but not vice versa.

In order to describe entanglement manipulations [61, 105] within Giles' formal theory, consider $\mathcal{S}$ to be the set of all quantum bipartite pure states and the operation $+$ to be the tensor product $\otimes$. The arrow will be defined in terms of transformations which convert bipartite states by only using local operations on the subsystems separately aided with classical communication between the subsystems (LOCC) [61, 105]. First, we give the definition of $\subset$ in the quantum setting and then give the precise definition of arrow in the spirit of axiom 7.

Definition 107 *We say that a pure state a is* contained *in a pure state b, denoted by $a \subset b$, iff there exists an integer n and a state c such that either of the following two cases is valid*

$$\begin{aligned} &\textit{i)} \quad (\forall\epsilon)(\exists\Phi \in \mathbf{LOCC}) : ||\Phi(a^{\otimes n} \otimes c) - b^{\otimes n}|| < \epsilon \\ &\textit{ii)} \quad (\forall\epsilon)(\exists\Phi \in \mathbf{LOCC}) : ||\Phi(b^{\otimes n}) - a^{\otimes n} \otimes c|| < \epsilon\,. \end{aligned}$$

In other words, a quantum state a is contained within a state b if, with the help of some other state c, a can be transformed by LOCC into b. Now we define what we mean by a transformation of one quantum state into another.

Definition 108 *We say that a pure state a can be* converted *into a pure state b,*

designated as $a \to b$*, iff*

$$(\forall \epsilon)(\exists c)(\forall \delta)(\exists n, m \in \mathbb{N},\ \ \Phi \in \mathbf{LOCC}\ \textit{and}\ \ x, y \in \mathcal{S})$$
$$\textit{such that}\ \ m/n < \delta\ \ ,\ \ x \subset mc\ \ ,\ \ y \subset mc\ \ \textit{and}$$
$$||\Phi(a^{\otimes n} \otimes x) - b^{\otimes n} \otimes y|| < \epsilon\,.$$

Product states act as equilibrium states in the sense that LOCCs cannot create entanglement out of them and the space of product (disentangled) states is invariant under LOCCs. Likewise, maximally entangled states act as anti-equilibrium states, in the sense that LOCCs naturally tend to destroy entanglement and therefore move away from this set. Now we define an entanglement measure in a similar way that to an entropy function.

Definition 109 *A real-valued function* E *defined over* $\mathcal{S}$ *is a called an* entanglement measure*, if*

- *For all state* a *and* b *we have :* $E(a \otimes b) = E(a) + E(b)$ *.*
- *If* $a \to b\ \ \&\ \ b \to a$*, then* $E(a) = E(b)$*.*
- *If* $a \to b\ \ \&\ \ b \not\to a$*, then* $E(a) > E(b)$*.*
- $E(a) = 1$ *if* a *is a maximally entangled sate.*

Giles' proof of uniqueness of an ordering function is constructive and can be found in [49]. In case of entanglement manipulations this leads to a definition for a measure of entanglement as following:

$$E(a) = \inf_{m,n \in \mathbb{N}} \{m/n \mid \exists x, y \in \mathcal{S}\ :\ y \subset e^{\otimes m}\ \&\ y \to a^{\otimes n} \otimes x\}\,,$$

where e is any maximally entangled state.

Our abstract approach to entanglement is different to the existing method where one looks for a minimal number of conditions for a measure of entanglement that would single out a unique one [102, 62, 90, 35]. The existing method has a strong flavour of Shannon's pioneering approach to information theory [96]. Shannon

considered functions on the set of probability distribution which would describe their information content. By introducing three natural conditions that this function should satisfy he arrived at a unique measure, called the Shannon entropy. These conditions are remarkably similar to the conditions leading to a unique measure of entanglement for pure bipartite states [90]. This, of course, is not surprising. It is well known that the Shannon entropy of the probabilities derived from Schmidt coefficients in the Schmidt decomposition [92] of a pure bipartite state is a good measure of entanglement [10].

5.5 Elementary Classical Information

As a very simple realisation of Giles' theory we comment on how to capture the most elementary notion of classical information, namely that the information carried by an event happening with probability p is $\log p$. For this, the states are real numbers $0 < r \leq 1$, $+$ represents multiplication of real numbers, and the arrow is the relation "less then or equal". It is very simple to check that all the axioms are satisfied. Following Giles' construction, we can derive a unique measure for this case which is then proportional to $\log p$. An open question now is to find dynamical processes that naturally lead to this realisation and derive the more general Shannon entropy $-\sum_i p_i \log p_i$.

5.6 Discussion

We show in this chapter that the Giles' formal theory not only describes the mathematical foundation of thermodynamics but can also describe entanglement manipulations. This approach to uniqueness of measure of entanglement for pure bipartite states is different to the existing method where one looks for a minimal number of conditions for a measure to satisfy such that one could single out a unique one [90, 35].

A natural question to ask is whether the same model can be applied to classical information theory and derive the classical entropy from the dynamical perspective, rather than the usual axioms of Shannon [96]. This could be done by defining

the states to be probability distributions (much like in statistical mechanics, where these would correspond to occupational probabilities of different energy levels). In addition, the arrow would be defined by a stochastic map taking one probability distribution into another one. This is in the same spirit as Penrose's formulation of statistical mechanics [85]. However, in order to derive a unique measure, the notion of arrow needs to be generalised to satisfy all the axioms. We believe this can be done in an asymptotic way by converting multiple copies of the same probability distribution with the aid of another (arbitrary) catalyser probability distribution. This conjecture remains to be proven. This would not only show that there are deep mathematical connections between thermodynamics, statistical mechanics and information theory, both classical and quantum, but also that they all in fact arise from the same mathematical framework presented here.

[1] S. Aaronson. Quantum lower bound for the collision problem. In *Proceedings of STOC'96 – Symposium on the Theory of Computing*. ACM, 2001.

[2] S. Abramsky and A. Jung. Domain theory. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 3. Clarendon Press, 1994.

[3] M. Adcock and R. Cleve. A quantum goldreich-levin theorem with cryptographic applications. In *Proceedings of 19th Annual Symposium on Theoretical Aspect of Computer Sciences*, page 323, 2002.

[4] D. Aharonov, A. Kitaev, and N. Nisan. Quantum circuits with mixed states. In *Proceedings STOCK'98 – Symposium on Theory of Computing*, Dallas, TX USA, 1998. ACM.

[5] M. Alvarez-Manilla. *Measure Theoretic Results for Continuous Valuations on Partially Ordered Spaces*. PhD thesis, University of London, Imperial College, 2000.

[6] A. Ambainis. Quantum lower bounds by quantum arguments. In *Proceedings of STOC'00 – Symposium on the Theory of Computing*, page 636, 2000.

[7] R. Beals, H. Buhram, R. Cleve, M. Mosca, and R. Wolf. Quantum lower bounds by polynomials. In *Proceedings of FOCS'98 – Symposium on Foundations of Computer Science*, 1998.

[8] P.A. Benioff. The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. *Journal of Statistical Physics*, 22(5):563, 1980.

[9] C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26:1510, 1997.

[10] C. H. Bennett, H. J. Bernstein, S. Popescu, and B. Schumacher. Concentrating partial entanglement by local operations. *Phys. Rev. A*, 53, 1996.

[11] C. H. Bennett and S. J. Wiesner. Communication via one- and two-particle operators on einstein-podolsky-rosen states. *Phys. Rev. Lett.*, 69, 1992.

[12] C.H. Bennett. Logical reversibility of computations. *IBM Journal of Research and Development*, 17(6):525, 1973.

[13] C.H. Bennett, G. Brassard, C. Crepeau, R. Jozsa, A. Peres, and W.K. Wootters. Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Phys. Rev. Lett.*, 70, 1993.

[14] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM Journal of Computing*, 5(26):1411, 1997.

[15] A. Berthiaume and G. Brassard. The quantum challenge to structural complexity theory. In *Proceedings of the 7th Annual IEEE conference on Structure in Complexity Theory*, Boston, 1992.

[16] G. Birkhoff. *Lattice Theory*, volume 25 of *R. I.* Amer. Math. Soc. Colloquium, 1967.

[17] J. Blanck. Domain representability of metric spaces. *Annuals of Pure and Applied Logic*, 43, 1997.

[18] M. Boyer, G. Brassard, P. Høyer, and A. Tapp. Tight bounds on quantum searching. *Fortsch. Phys.*, page 493, 1998.

[19] G. Brassard and C.H. Bennett. Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers Systems and Signal Processing*, Bangalore India, 1984.

[20] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf. Quantum fingerprinting. 87, 2001.

[21] H. Buhrman, C. Dürr, M. Heiligman, P. Høyer, F. Magniez, M. Santha, and R. Wolf. Quantum algorithms for element distinctness. In *Proceedings of Sixteenth IEEE Conference on Computational Complexity*, page 131. IEEE Press, 2001.

[22] H. Buhrman and W. van Dam. Quantum bounded query complexity. In *Proceedings of COCC'99 – Annual IEEE Conference on Computational Complexity*, page 149, Atlanta,USA, 1999. IEEE Press.

[23] S. Bettelliand T. Calarco and L. Serafini. Toward an architecture for quantum programming. *arXive:cs.PL/0103009*, 2001.

[24] C. Carathodory. *Math. Ann.*, 67, 1909.

[25] Goong Chen and Zijian Diao. An exponentially fast quantum search algorithm. *submitted to Phys. Rev. Lett*, 2000.

[26] I.L. Chuang, N. Gershenfeld, and M. Kubinec. Experimental implementation of fast quantum searching. *Phys. Rev. Lett.*, 80:3408, 1998.

[27] K.L. Chung. *A Course in Probability Theory*. Academic Press, 1974.

[28] R. Cleve. An introduction to quantum computational complexity. In C. Macchiavello, G.M. Palma, and A. Zeilinger, editors, *Collected Papers on Quantum Computation and Quantum Information Theory*. World Scientific, 1999.

[29] R. Cleve, A. Ekert, L. Henderson, C. Macchiavello, and M. Mosca. On quantum algorithms. *Complexity*, 4, 1998.

[30] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. Quantum algorithms revisited. In *Proceedings of the Royal Society of London*, volume 454, page 339, 1998.

[31] B. Coecke and K. Martin. A partial order on classical and quantum states. *Programming Research Group Research Report RR-02-07*, 2002.

[32] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. In *Proceedings of the Royal Society of London*, volume A400, page 97, 1985.

[33] D. Deutsch. Quantum computational networks. In *Proc. Roy. Soc. Lond A*, volume 425, page 467, 1989.

[34] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. In *Proc. Roy. Soc. Lond A*, volume 439, page 553, 1992.

[35] M. J. Donald, M. Horodecki, and O. Rudolph. The uniqueness theorem for entanglement measures. *J. Math. Phys*, 43, 2002.

[36] P. Dumais, D. Mayers, and L. Salvail. Perfectly concealing quantum bit commitment from any one-way permutation. In B. Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, number 1807 in Lecture Notes in Computer Science, Berlin — Heidelberg — New York, 2000. Springer Verlag.

[37] N. Dunford and J. Schwartz. *Linear Operators*, volume 1. Interscience, 1958.

[38] A. Edalat. Domains for computation in mathematics, physics and exact real arithmetic. *Bulletin of Symbolic Logic*, 3(4), 1997.

[39] A. Edalat and R. Heckmann. A computationa model for metric spaces. *Theoretical Computer Science*, 1996.

[40] A. Edalat and P. Sünderfhauf. A domain theoretic approach to computability on the real line. *Theoretical Computer Science*, 1997.

[41] E.H.Knill. Conventions for quantum pseucode. *LANL report LAUR-96-2724*, 1996.

[42] A. Ekert. Quantum cryptography based on bell's theorem. *Phys. Rev. Lett.*, 67, 1991.

[43] Y.L. Ershov. Computable functions of finite types. *Algebra and Logic*, 11(4), 1972.

[44] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, 1968.

[45] R.P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7):467, 1982.

[46] M. Garey and D. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. Freeman, San Francisco, 1979.

[47] G.Brassardand, P.Höyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. *AMS Contemporary Mathematics Series Millennium*, Quantum Computation & Information, 2000.

[48] P. Di Gianantonio. Real number computability and domain theory. *Information and Computation*, 127(1), 1996.

[49] R. Giles. *Mathematical Foundations of Thermodynamics*. International Series of Monographs on Pure and Applied Mathematics. Pergamon Press, 1964.

[50] A.M. Gleason. Measures of closed subspaces of a Hilbert space. *Journal of Mathematics and Mechanics*, 6:885, 1957. reprint in [60].

[51] O. Goldreich. *Foundations of Cryptography - Fragment of a Book*. 1995.

[52] J. Grollmann and A.L. Selman. Complexity measures for public-key cryptosystems. *SIAM journal of Computing*, 17:309, 1988.

[53] L.K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of STOC'96 – Symposium on the Theory of Computing*, page 212, Philadelphia, Pennsylvania, 1996. ACM.

[54] S. Hallgren. Polynomial-time quantum algorithms for pell's equation and the principal ideal problem. In *Proceedings of the Annual Symposium on Theoretical Aspect of Computer Sciences*, Lecture Note in Computer Sceinece, 2002.

[55] P. Halmos. *Measure Theory*. Van Nostrand, 1950.

[56] R. Heckmann. Probabilistic domains. *CAAP '94*, 1994.

[57] R. Heckmann. Space of valuations. In *Papers on General Topology and its Applications*, volume 806. Annals of the Ney York Academy of Science, 1996.

[58] R. Heckmann. Approximation of metric spaces by partial metric spaces. *Applied Categorical Structures*, 7, 1999.

[59] L. Hemaspaandra and J. Rothe. Characterizing the existence of one-way permutation. *Theoretical Computer Science*, 244(1-2):257, 2000.

[60] C.A. Hooker, editor. *The Logico-Algebraic Approach to Quantum Mechanics*, volume I – Historical Evolution. Reidel, Dordrecht – Bosten, 1975.

[61] M. Horodecki. Entanglement measures. *Quantum information and Computation*, 1, 2001.

[62] M. Horodecki. Entanglement measures. *Quant. Inf. Comp.*, 1, 2001.

[63] C.J. Isham. *Lectures on Quantum Theory – Mathematical and Structural Foundations*. Imperial College Press, London, 1995.

[64] C. Jones and G. Plotkin. A probabilistic powerdomain of evaluations. *Logic in Computer Science*, 1989.

[65] J.A. Jones, M. Mosca, and R.H. Hansen. Implementation of a quantum search algorithm on a quantum computer. *Nature*, 393:344, 1998.

[66] R. Jozsa. Characterising classes of functions computable by quantum parallelism. In *Proceedings of the Royal Society of London*, volume A435, 1991.

[67] R. Jozsa. Entanglement and quantum computation. In S Huggett, L Mason, K P Tod, S T Tsou, and N Woodhouse, editors, *Geometric Issues in the Foundations of Science*. Oxford University Press, 1998.

[68] R. Jozsa. Quantum factoring, discrete logarithms and the hidden subgroup problem. *Special issue of IEEE Computing in Science and Engineering*, 3, 2001.

[69] R. Jozsa. Notes on hallgren's efficient quantum algorithm for solving pell's equation. *quant-ph/0302134*, 2003.

[70] B.E. Kane. A silicon-based nuclear spin quantum computer. *Nature*, 393:133, 1998.

[71] E. Kashefi, A. Kent, V. Vedral, and K. Banaszek. A comparison of quantum oracles. *Phys. Rev. A*, 65, 2002.

[72] E. Kashefi, H. Nishimura, and V. Vedral. On quantum one-way permutations. *Quantum Information and Computation*, 5, 2002.

[73] E. Kashefi and V. Vedral. Physical reversibility and one-way functions. In *Proceedings of QCMC'02 – The Sixth International Conference on Quantum Communication, Measurement and Computing*. Rinton Press, 2002.

[74] D. Kozen. Semantics of probabilistic programs. *Computer and System Sciences*, 22(3), 1981.

[75] E. H. Lieb and J. Yngvason. The mathematics of the second law of thermodynamics. *Geom. Funct. Anal.*, Part 1 Sp. Iss. SI, 2000.

[76] K. Martin. Powerdomains and zero finding. *QAPL*, 59(3), 2001.

[77] M. Mosca and A. Ekert. The hidden subgroup problem and eigenvalue estimation on a quantum computer. In *Proceedings of the 1st NASA International Conference on Quantum Computing and Quantum Communication*, 1999.

[78] M. A. Nielsen. Computable functions, quantum measurements, and quantum dynamics. *Phys.Rev.Lett.*, 79, 1997.

[79] M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2000.

[80] H. Nishimura. *Computational Complexity Theory of Quantum Turing Machines and Quantum Circuits*. PhD thesis, Nagoya University, 2000.

[81] B. Ömer. A procedural formalism for quantum computing. Master's thesis, Department of Theoretical Physics – Technical University Vienna, Vienna, 1998.

[82] B. Ömer. *Quantum Programming in QCL*. PhD thesis, Institute of Information Systems – Technical University Vienna, Vienna, 2000.

[83] M. Ozawa. Measurability and computability. *quant-ph/9809048*, 1998.

[84] C.H. Papadimitriou. *Computational Complexity*. Addision-Wesley, 1994.

[85] O. Penrose. *Foundations of Statistical Mechanics*. Pergamon Press, 1970.

[86] M. B. Plenio and V. Vedral. Entanglement in quantum information theory. *Cont. Phys.*, 39, 1998.

[87] S. Popescu and D. Rohrlich. Thermodynamics and the measure of entanglement. *Phys Rev. A*, 56, 1997.

[88] R. Raussendorf and H. Briegel. Computational model underlying the one-way quantum computer. *QIC*, 2(6), 2002.

[89] R. Raussendorf, D.E. Browne, and H.-J. Briegel. The one-way quantum computer - a non-network model of quantum computation. *Journal of Modern Optics*, 49, 2002.

[90] O. Rudolph. A new class of entanglement measures. *J. Math. Phys.*, 42, 2001.

[91] J.W. Sanders and P. Zuliani. Quantum programming. In R. Backhouse and J.S. Nuno Oliveira, editors, *Proceedings of MPC'00 – Mathematics of Program Construction*, volume 1837 of *Lecture Notes in Computer Science*. Springer-Verlag, 2000.

[92] E. Schmidt. *Math. Annalen*, 63, 1907.

[93] D. Scott and C. Strachey. Towards a mathematical semantics for computer languages. *Tech. mono. PRC6*, 1971.

[94] D. S. Scott. Outline of a mathematical theory of computation. In *4th Annual Princeton Conference on Information Science and Systems*, 1970.

[95] P. Selinger. Towards a quantum programming language. *submited*, 2003.

[96] E. Shannon and W. Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, 1949.

[97] P.W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings of FOCS'94 – Symposium on Foundations of Computer Science*, page 124, Santa Fe, New Mexico, 1994. IEEE Press.

[98] D.R. Simon. On the power of quantum computation. *SIAM J. Computing*, 26:1474, 1997.

[99] M. B. Smyth. Effectivly given domains. *Theoretical Computer Science*, 5, 1977.

[100] W. Thirring. *A course in mathematical physics*. Springer-Verlog, 1979.

[101] Q.A. Turchette. *Phys. Rev. Lett.*, 81, 1998.

[102] V. Vedral V and M. B. Plenio. Entanglement measures and purification procedures. *Phys. Rev. A*, 57, 1998.

[103] W. van Dam. Quantum oracle interrogation: Getting all information for almost half the price. In *Proceedings of FOCS'98 – Symposium on Foundations of Computer Science*, page 362, 1998.

[104] Wim van Dam. Quantum oracle interrogation: Getting all information for almost half the price. In *Proceedings FOCS'98 – Symposium on Foundations of Computer Science*, page 362, quant-ph/9805006, 1998.

[105] V. Vedral. The role of relative entropy in quantum information theory. *Rev. Mod. Phys.*, 2002.

[106] V. Vedral and E. Kashefi. A unified axiomatic approach to information content of physical states. In *Proceedings of QCMC'02 – The Sixth International Conference on Quantum Communication, Measurement and Computing*. Rinton Press, 2002.

[107] V. Vedral and E. Kashefi. Uniqueness of entanglement measure and thermodynamics. *Phys. Rev. Lett.*, 89, 2002.

[108] J. von Neumann. *Mathematical Foundations of Quantum Mechanics*. Princeton University Press, Princeton, 1955.

[109] J. Watrous. Succinct quantum proofs for properties of finite groups. In *Proceedings of FOCS'2000 – Symposium on Foundations of Computer Science*, page 537. IEEE Press, 2000.

[110] J. Watrous. Quantum algorithms for solvable groups. In *Proceedings of STOC'01 – Symposium on Theory of Computing*, page 60, 2001.

[111] K. Weihrauch. *Computable Analysis*. Springer, 2000.

[112] K. Weihrauch and U. Schreiber. Embedding metric spaces into cpo's. *Theoretical Computer Science*, 16, 1981.

[113] T. Yamakami. Quantum np and a quantum hierarchy. In *Proceedings of 2nd IFIP International Conference on Theoretical Computer Science*, page 323, 2002.

[114] A. C. C. Yao. Quantum circuit complexity. In *Proceedings of FOCS'93 – Symposium on Foundations of Computer Science*. IEEE Press, 1993.

[115] C. Zalka. Grover's quantum searching algorithm is optimal. *Phys. Rev. A*, 60:2746, 1999.