NAIST-IS-DD1161015

Doctoral Dissertation

On the Power and Limitations of Quantum Computing Models: Quantum Walks and Communication Complexity

Marcos Villagra

February 14, 2013

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of SCIENCE

Marcos Villagra

Thesis Committee:

Professor Yasuhiko Nakashima	(Supervisor)
Professor Hiroyuki Seki	(Co-supervisor)
Professor Shigeru Yamashita	(Co-supervisor, Ritsumeikan University)
Associate Professor Masaki Nakanishi	(Co-supervisor, Yamagata University)

On the Power and Limitations of Quantum Computing Models: Quantum Walks and Communication Complexity*

Marcos Villagra

Abstract

To understand the physical limits of computation it is necessary to shift our classical computer models to ones that take into account physical considerations. The best current theory of physical reality is quantum mechanics, which take us to think on computer models based on it. Quantum computation is the study of the power and limitations of computer models that consider quantum mechanical effects like interference and entanglement. Several of the classical computing models like boolean circuits, Turing machines, etc., can be extended to quantum models of computation. In this research we focus on two particular models: the decision tree complexity and communication complexity.

This research is divided in two parts. First we consider Quantum Walks, a very powerful paradigm for the design and analysis of quantum algorithms. Clear mathematical foundations are still lacking for this paradigm. Hence, as a step toward this objective, the following question is being addressed: *Given a graph, what is the probability that a quantum walk arrives at a given vertex after some number of steps?* This is a very natural question, and for classical random walks it can be answered by different combinatorial arguments. For quantum walks this is a highly non-trivial task. Furthermore, this was only achieved before for one specific coin operator (Hadamard operator) for walks on the line. Even considering only walks on lines, generalizing these computations to a general $SU(2)$ coin operator is a complex task. The main contribution of this part is

* Doctoral Dissertation, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1161015, February 14, 2013.

a closed-form formula for the question above for a general symmetric $SU(2)$ operator for walks on lines. As the second contribution, this thesis presents how some basic properties of the walk can be deducted by means of weak convergence theorems for quantum walks.

The second part of this research considers communication complexity; in particular, quantum nondeterministic multiparty communication. There are three different types of nondeterminism in quantum computation: i) strong, ii) weak with quantum proofs, and iii) weak with classical proofs. This thesis is focused on strong quantum nondeterministic protocols where a correct input is accepted with positive probability, and an incorrect input is rejected with probability 1. By extending the definition proposed by de Wolf to nondeterministic tensor-rank ($nrank$), this thesis shows that for any boolean function f, when there is no prior shared entanglement, the strong quantum nondeterministic communication complexity 1) is upper-bounded by the logarithm of $nrank(f)$ in the Number-On-Forehead model; and, 2) in the Number-In-Hand model it is lower-bounded by the logarithm of $nrank(f)$. One application is a new lower bound for the generalized inner product function on the Number-In-Hand model. As another application of the main result this thesis shows that when the number of players in the protocol is $o(\log \log n)$ we have $\mathbf{NQP} \not\subseteq \mathbf{BQP}$ in the Number-On-Forehead model.

Keywords:

Quantum Computing, Decision Trees, Quantum Walks, Communication Complexity, Complexity Classes, Nondeterministic Communication

To Maggie and Val

Preface

This thesis is the culmination of four years of work. It started in April 2009 together with my graduate studies at Nara Institute of Science and Technology. However, the path to my graduation can be traced back even further. In June 2006 when I finished my undergraduate studies at the Catholic University of Asuncion in Paraguay I knew very well that I wanted to continue my studies in Japan. The next year I applied to the MEXT scholarship at the Embassy of Japan in Paraguay and received the good news at the end of the same year. I arrived in Japan on April 3rd, 2008 and went on to study Japanese language at Osaka University for six months. From October 2008 to March 2009 I was a research student at Nara Institute of Science and Technology, and it was during that time that the first ideas for this thesis started to be developed.

Before starting my graduate studies I had some experience on doing research. I was involved mainly in artificial intelligence and combinatorial optimization, in particular, local search heuristics for SAT which involved a lot of programming and experiments. However, my interests always were on theoretical computer science, so I chose quantum complexity theory as the main subject for my graduate studies.

I have to admit that the transition from experimental algorithmics to a field rooted in mathematics was not as smooth as I thought it would be. I had the skills needed but the intuition for mathematical work was not there. That required for me to start from scratch with math and learn how to do research all over again. That road had a lot bumps but I strived to the end.

The process for learning how to do research in mathematics was a very gratifying experience (as it should be for any endeavor). Sometimes it was frustrating, specially when I was stuck and completely loss on how to do progress. However, when I solved a problem, that feeling overwhelmed any other failures I had in the past. During this learning process I received a lot help from my advisors, colleagues and friends.

First of all I want to thank my thesis committee Yasuhiko Nakashima, Hiroyuki Seki, Shigeru Yamashita, and Masaki Nakanishi for all the hard work on reviewing this thesis. Prof. Nakashima always supported me and gave me confi-

dence and freedom to do my research the way I wanted. He also taught me how to write grant proposals, which is essential in science today. I want to thank Prof. Seki for all the wonderful classes on theoretical computer science and the fun book reading sessions. My advisors Prof. Shigeru Yamashita and Prof. Masaki Nakanishi were accompanying me from the start. They shared their experience and taught me how to do mathematics.

From NAIST I would also like to thank Prof. David Sell. I was his teaching assistant on the different english courses offered at the Graduate School of Information Science. I enjoyed our many discussions on philosophy. I also thank Prof. Jun Yao for his friendship and advice. We had many interesting and fun discussions on research practices of theorists vs the rest of computer science. Many thanks also go to the International Students Affairs Division from which I received guidance and invaluable advice during my stay at NAIST.

From the Polytechnic School of the National University of Asuncion I want to thank Benjamín Barán, Mariano Bordas, Carlos Brizuela, María Elena García, Pedro Gardel, Diego Pinto, Christian Schaerer, for all the support I received from the distance before and during my graduate studies. In particular, I thank Mariano Bordas for all his help during the last five years.

Special thanks go to Benjamín Barán. He was my mentor and the one who introduced me to science during my college years. An outstanding human being whose enthusiasm and devotion to science continues to be an example to me and to his many students.

I also benefited from conversations and discussions with several colleagues. Here I would like to thank Abdulrahman Al-lahham, Gilles Brassard, Andrew Childs, Kassem Kalach, Hoi Kwan (Kero) Lau, Youngrong Lim, François Le Gall, Michele Mosca, Pavithran Sridharan Iyer, Xiaoming Sun, Seiichiro Tani, Chao Wang, and Tomoyuki Yamakami. Here I would like to specially thank Xiaoming Sun for receiving me as an intern in his research group.

Big thanks also goes to the CSTheory StackExchange community. This web site is turning into a very important source of knowledge for the theory community worldwide. Here I was able to learn different fields in theory and very important research practices. Thanks go to the moderators Dave Clark, Kaveh Ghasemloo and Suresh Venkatasubramanian for their hard work on maintaining the site.

Deep thanks to MEXT (Ministry of Education, Culture, Sports, Science & Tecnology) of Japan for granting me a scholarship that allowed me to finnish my graduate studies. My research was also partly supported by the NEC C&C Foundation to whom I am grateful. I also thank the JSPS (Japan Society for the Promotion of Science) for trusting me with a fellowship to continue my research as a postdoc.

I would like to thank my parents for supporting me through all these years. From my father in particular I learned how to think freely and have a critical eye.

This thesis would have not been possible without the support of my loving family, my wife Maggie and my daughter Valery. They kept my feet on the ground and prevented me from losing my mind during this learning process.

Finally I thank you, the reader, for your interest on this writing. Even though I haven't put too much effort to make it an introductory and self-contained reading, I hope you find it interesting and serves as a guidance to further work on quantum computation and computational complexity theory.

Contents

List of Figures

List of Tables

List of Symbols

$\hat{D}(f)$ Deterministic decision tree complexity of f

$\hat{R}_\epsilon$(f) Randomized decision tree complexity of f

$\hat{U}_\epsilon(f)$ distributional complexity with error ϵ of f

$\left|\psi_x^T\right\rangle$ State of a quantum query algorithm after T queries

$\wedge$ Bit-wise AND

$\mathbb{C}$ Set of complex numbers

$\mathcal{A}_f$ Set of deterministic algorithms for a Boolean function f that fails to give a correct answer on some inputs

$\mathcal{I}$ Finite set of inputs

$\mathcal{T}_f$ Set of all deterministic decision trees computing f

$\mathcal{T}_{f,\epsilon}$ Set of deterministic decision trees computing f that err on at most a fraction ϵ of the inputs

μ_k^α k-party approximate μ-norm

$\Upsilon(\epsilon)$ Subset of $\mathcal{A}_f$ given by $\Upsilon(\epsilon) = \{A : A \in \mathcal{A}_f, \sum_{x\in\mathcal{I}} d(x) \cdot \varphi(A, x) \leq \epsilon\}$

$\widetilde{deg}(f)$ Approximate degree of f

C Combinatorial object which could be either a cube or a cylinder intersection

$Cov(f)$ Cover number of f

$Cov^z(f)$ z-cover of f

$D_k(f)$ k-party deterministic communication complexity of f

$Disc(f)$ Generalized discrepancy of f

EQ_k Equality function on k inputs

GIP_k Generalized inner product function on k inputs

M_f Communication matrix of f

$N_k(f)$ k-party nondeterministic communication complexity

$N_k^{NIH}(f)$ k-party nondeterministic NIH communication complexity

$N_k^{NOF}(f)$ k-party nondeterministic NOF communication complexity

$nrank(f)$ Nondeterministic rank of the communication tensor T_f

O_x Quantum oracle on input x

P_i Player i in a communication protocol

$Pr[\cdot]$ Probability of some event

$Q_k(f)$ Exact k-party quantum communication complexity

$Q_{\epsilon,k}(f)$ 2-sided bounded-error k-party quantum communication complexity with error probability upper-bounded by ϵ of f

$R_{\epsilon,k}^{NOF}$ k-party bounded-error NOF communication complexity with error probability upper-bounded by ϵ

$R_{k,\epsilon}(f)$ k-party bounded-error communication complexity of f for both NIH and NOF models

$rank_i(T)$ Rank of the i-mode unfolding of tensor T

$SQ_k^{NIH}(f)$ k-party strong nondeterministic NIH communication complexity of f

$SQ_k^{NOF}(f)$ k-party strong nondeterministic NOF communication complexity of f

$|\psi\rangle$ quantum state

$\mathcal{H}$ Hilbert space

$rank(f)$ Rank of the communication matrix M_f of some boolean function f

α, β complex amplitudes in a quantum state

$span\{\cdot\}$ vector space spanned by a set of vectors

$|u\rangle \otimes |v\rangle$ tensor between vectors $|u\rangle$ and $|v\rangle$

NOT not operation for 1 qubit

H Hadamard operation for 1 qubit

$CNOT$ controlled NOT operation for 2 qubits

$SWAP$ swap operation between 2 qubits

W Walsh-Hadamard transform

(V, E) graph with set of vertices V and set of edges E

$|\Psi_t\rangle$ state of the walk at time t

$|\psi_t(v)\rangle$ projection of the state of the walk onto position v at time t

$|d, v\rangle$ basis state of the walk at position v in direction d

NIH Number-In-Hand model

NOF Number-On-Forehead model

$\alpha_t^d(v)$ amplitude at time t of direction d and position v

$P(v, t)$ probability of finding the walk at position v at time t

U time-evolution operator of the walk

S	shift operation
C	coin operation
G	Grover operation
I	identity operation
F	Discrete Fourier Transform
X_t	random variable for the position of the walk at time t
$\left\|\widetilde{\psi}_t(k)\right\rangle$	basis vector of the walk in Fourier space
$\lambda_d(k)$	eigenvalue of the time-evolution in Fourier space
$\rightsquigarrow$	weak convergence

Chapter 1

Introduction

In the mid-90s one of the greatest discoveries in computer science was a polynomial-time algorithm for factoring composite numbers. Given that the security of cryptographic systems depend on the difficulty of factoring large composite numbers, this discovery implies that all the security of online transactions can be broken using this algorithm. But there was a catch, this algorithm only runs on a quantum computer.

A quantum computer is a computing model that directly uses quantum mechanical effects for computation. The original idea can be traced back to the physicist Richard Feynman [Fey82], and was later formalized in the mid-80s by another physicist, David Deutsch [Deu85]. At that time, people saw the model as another curious computer model until the seminal paper by Peter Shor, on the polynomial-time factoring algorithm [Sho94]. This gave a clear evidence that there could be practical problems that a quantum computer can solve faster than classical computers. In fact, as a plausible model of computation, it even challenges the strong version of the Church-Turing Thesis[1]. This fact was reinforced when researchers discarded it as a model of analog computation by showing that a quantum computer is tolerant against a finite amount of noise [NC00].

Since the discovery of Shor's algorithm, the scientific community started to get interested and several other quantum algorithms were discovered. Among those, one of the most important is the algorithm for searching an unstructured

[1]**Strong Church-Turing Thesis:** Any algorithmic process can be simulated efficiently using a probabilistic Turing machine.

search space discovered by Lov Grover [Gro96]. Given a set of n elements, the algorithm finds a marked element in $\mathcal{O}(\sqrt{n})$ steps (a quadratic speed-up with respect to classical search). This bound turned out to be tight [BBBV97] and presented evidence against the possibility that **NP**-complete problems could be efficiently solved by quantum computers.

1.1 Quantum Computing Models and Paradigms

To be able to say anything formal about quantum computation it is necessary to develop formal models of it. In the seminal paper of Bernstein and Vazirani [BV97] the Quantum Turing Machine model was introduced in its current form. Previous models were developed by Benioff [Ben80], Deutsch [Deu85] and Yao [Yao93]. In [BV97] it was also shown that the Quantum Turing Machine is universal with only a polynomial overhead on the simulation of any other machine. The authors also showed several properties like $\mathbf{BQP} \subseteq \mathbf{P}^{\#\mathbf{P}}$ where **BQP** is the class of languages with efficient quantum Turing machines.

Another important model is the Quantum Circuit Model originally introduced by Deutsch [Deu85] which was later developed by Yao [Yao93]. Yao also showed in the same piece of work that Quantum Turing Machines and Quantum Circuits are indeed equivalent in power. Shor's factoring algorithm was given in this model of computation [Sho94].

The Quantum Turing machine and Quantum Circuits are the models that could be considered the most significant when researching about the power and limitations of quantum computation. However, the existing problems in classical complexity theory carry over to the quantum world including the known barriers for proving class separations [AB09]. In particular, it is hard to prove lower bounds on these two models. Furthermore, since quantum mechanics requires reversible operations, also proving upper bounds turns out to be hard.

In order to prove facts about quantum computation, and at the same time, to be able to say significant things about it, researchers focused on more simple models of computation. Two of the more popular and widely studied models are quantum decision trees and quantum communication.

1.1.1 Quantum Decision Trees

A Quantum Decision Tree is the quantum counterpart of the well studied decision tree model. In the quantum computing community, it is common to refer to this model as quantum query complexity. This thesis uses both names interchangeably.

Normally it is easy to see a quantum query algorithm as an algorithm that has access to a black-box or oracle. Access to the input is only possible by making queries to this black-box, hence the name. This way, the complexity is measured in terms of the number of queries to the oracle in order to compute some function. Other computations, beside the oracles queries, can be made free of cost.

The first algorithm under this model was the celebrated Deutsch-Jozsa algorithm. Also, Shor's quantum algorithm has in its structure a quantum decision tree in operation. One lower bound technique is the *polynomial method* discovered by Beals, Buhrman, Cleve, Mosca and de Wolf [BBC$^+$01]. This technique allows to lower bound the quantum query complexity by computing the minimum degree of an approximating polynomial of the boolean function. An alternative technique is the *quantum adversary method* discovered by Ambainis [Amb00]. This technique is based on upper-bounding the amount of information obtained from each query. The quantum adversary method is the most researched technique in recent years with several works improving it to obtain better lower bounds. This culminated recently in a breakthrough result by Reichardt [Rei10]: a variant called The Negative Adversary method is optimal for quantum query complexity.

In the upper bounds realm it is generally easier to construct quantum query algorithms. This is mostly due to the fact that we can ignore computations costs that are not queries to the quantum oracle. However, a "real" quantum algorithm, i.e., one based on quantum Turing machines or quantum circuits, needs to take into considerations all computations. Hence, to actually construct quantum algorithms researchers turned to a new paradigm known as *quantum walks*.

Quantum walks are the quantum counterpart of classical random walks and Markov chains and they served as a paradigm for the design and analysis of quantum algorithms. It is expected that this tool would make the job easier for quantum algorithm designers. As an example of the power of the paradigm,

Ambainis [Amb07] was able to give an optimal algorithm for the element distinctness problem (see Chapter 2 of this thesis), and Magniez, Santha and Szegedy showed an almost-optimal algorithm for finding triangles in graphs [MSS07] (later generalized by Childs and Kothari [CK11] also using quantum walks).

1.1.2 Quantum Communication

The communication model for boolean functions was introduced by Yao [Yao79]. Originally it was motivated by the study of parallel computation and boolean circuits. However, the wide range of applicability to different areas like data structures, streaming algorithms, VLSI, etc, made it one of the more studied models of computation [KN97].

The model is basically the following. There are two or more players with unlimited computational resources seeking to compute some boolean function. The input to the function is distributed among the players in a way that any one player is not able to compute the function by itself, and thus, the party is forced to communicate. Each player can send a message (string of bits) to any other player. The *communication complexity* is defined as the minimum amount of communication required to compute the function. If we allow the players to send quantum messages (or quantum bits) to each other we have the quantum communication model [Yao93].

Most of the research in communication complexity is focused on lower bounds. We have a tight relation between classical and quantum decision trees and communication: lower bounds in communication imply lower bounds on decision trees, and, upper bounds for decision trees imply upper bounds for communication [KN97, BCW98]. Also, by proving lower bounds in communication we can obtain lower bounds on the size and depth of boolean circuits and Turing machines [KN97].

There are many lower bound techniques in the literature touching all aspects of mathematics (for quantum and classical communication). These techniques range from combinatorial, algebraic, information-theoretic, analytical, geometrical, to name a few. A few year ago the hardest kind of lower bounds to come by were on quantum communication. This changed upon the discovery of two important techniques: the *norm bound* by Linial and Shraibman [LS09c] and the

pattern matrix method by Sherstov [She08][2]. Previous techniques worked fine for 2-party communication, however, for three or more players (multiparty communication) they delivered weak bounds, normally exponentially decreasing in the number of players. The norm bound and the pattern matrix method, with their generalization to tensors, delivered new and stronger lower bounds for a variety of problems [LS09b]. Recently, there is a new technique, which is purely information-theoretic, known as *information complexity*. This is a very promising novel technique with new outstanding results [KLL$^+$12].

1.2 Contributions of this Thesis

This work studies some fundamental questions about quantum walks as a tool for the construction of quantum query algorithms and the communication model for quantum computation. This thesis can be divided in two big parts. The first part deals with quantum walks and studies its dynamics as a random process. The second part of this thesis looks into the communication model, in particular, quantum nondeterministic communication. Even though nondeterminism is not a realistic model of communication, it is still important mainly due to the fact that lower bounds on nondeterministic communication imply lower bounds for randomized and deterministic communication.

The main contributions can be summarized as follows.

1. A new closed-form formula to compute the induced probability distribution of quantum walks on lines (Chapter 2).
2. A new lower bound for multiparty strong quantum nondeterministic communication based on tensor rank (Chapter 3).

In the following, a more detailed summary of the contributions of this thesis is explained.

[2]Actually, the norm bound and pattern matrix techniques were inspired by Razborov's method [Raz03] which used a multidimensional generalization of discrepancy to prove an optimal lower bound on set disjointness.

1.2.1 Quantum Walks on the Line with Phase Parameters

Here a study of discrete-time coined quantum walks is presented. Clear mathematical foundations are still lacking for this quantum walk model. For example, in random walks we can relate exactly the eigenvalues of a graph to its hitting and mixing times [Str05]; on the other hand, in coined quantum walks it is not known if such a relation exists.

As a step toward finding mathematical foundations of quantum walks, here the following question is being addressed: *What is the probability that a quantum walk arrives at a given vertex after some number steps?* This is a very natural question, and for random walks it can be answered by several different combinatorial arguments [Str05]. For quantum walks, this is a highly non-trivial task. Furthermore, this was only achieved for one specific coin operator (Hadamard operator) for walks on the line [ABN$^+$01, Kon03, GJS04, CSL08]. Even considering only walks on lines, generalizing these computations to a general $SU(2)$ coin operator is a complex task.

The main contribution of this part of the thesis is a closed-form formula for the question above for a general symmetric $SU(2)$ operator for walks on lines (theorem 2.5.1). To this end, first a coin operator with parameters that alters the phase of the state of the walk on the line is proposed. Then, the spectrum of the unitary evolution operator of the walk is computed by means of Fourier analysis. Finally, closed-form solutions can be approximated using an asymptotic approximation method known as the steepest descent method [Won01, Mil06] from complex analysis. The error terms for this approximation can be derived from the steepest descent method itself and the Euler-Maclaurin summation formula [Apo99].

As the second contribution of this part, some basic properties of the walk are examined by means of weak convergence theorems for quantum walks [GJS04]. First, the support of the induced probability distribution of the walk is computed. Then, it is shown how changing the parameters in the coin operator affects the resulting probability distribution.

1.2.2 Tensor Rank and Strong Quantum Nondeterminism in Multiparty Communication

This part of the thesis studies quantum nondeterminism in multiparty communication. There are three (possibly) different types of nondeterminism in quantum computation: i) strong, ii) weak with quantum proofs, and iii) weak with classical proofs. It is common to refer to these notions as **NQP**-communication, **QMA**-communication, and **QCMA**-communication respectively [RS04, Kla11]. This work is focused on the first one. A strong quantum nondeterministic protocol accepts a correct input with positive probability, and rejects an incorrect input with probability 1.

The main result of this part relates the strong quantum nondeterministic multiparty communication complexity to the rank of the communication tensor in the Number-On-Forehead and Number-In-Hand models. In particular, by extending the definition proposed by de Wolf to *nondeterministic tensor-rank* ($nrank$), it is shown that for any boolean function f when there is no prior shared entanglement between the players, 1) in the Number-On-Forehead model the cost is upper-bounded by the logarithm of $nrank(f)$, and 2) in the Number-In-Hand model the cost is lower-bounded by the logarithm of $nrank(f)$. Furthermore, as another application, when the number of players is $o(\log\log n)$ we have $\mathbf{NQP} \not\subseteq \mathbf{BQP}$ for Number-On-Forehead communication, where **NQP** and **BQP** are the classes of boolean functions with efficient[3] strong quantum nondeterministic protocols and bounded-error randomized protocols respectively.

1.3 Outline of the Thesis

The first part of this research appears in Chapter 2 where the contributions related to quantum walks is presented. As motivation, first the chapter starts with a general definition of a quantum walk with some applications for upper-bounding decision tree depth. Then the chapter proceeds with developing the main analysis technique. For the reader with no previous experience with quantum computation

[3] A protocol is efficient in communication complexity if the cost is polylogarithmic on the size of the input.

and decision trees, a brief introduction is presented in appendices A and B.

The second part is related to lower bounds on quantum communication. In Chapter 3, this thesis studies the notion of strong quantum nondeterministic communication. Then a brief introduction to tensors is presented. After the explanation of the main result of the chapter, two applications are given: 1) a new lower bound on the generalized inner product function in the Number-In-Hand model in Section 3.4, and 2) the $\mathbf{NQP} \not\subseteq \mathbf{BQP}$ proof for Number-In-Forehead communication in Section 3.6. A brief introduction to communication complexity can be found in Appendix C.

Chapter 2

Quantum Walks

Contents

2.1 Background

The design of quantum algorithms is nowadays one of the major problems in the quantum computing community. The strategies for writing classical algorithms

as divide and conquer, dynamic programming, etc, are not easily adapted to the quantum paradigm. Strategies for designing quantum algorithms are phase amplification, phase estimation, to name a few. As an example of the applications of these strategies, Grover's algorithm uses the amplitude amplification technique, and Shor's algorithm relies in reductions to order finding and phase estimation [NC00]. Therefore, it becomes necessary the study of different approaches to improve the efficiency of the search.

One of the emergent alternatives for the design of algorithms are quantum walks, in direct analogy to random walks in classical computing. Random walks showed to be a successful tool for designing algorithms, and the same success is expected in the quantum realm. Results in this field showed that quantum walks can outperform its classical counterpart by exploiting quantum mechanical effects such as interference and superposition, giving and exponential speedup for certain types of graphs, and polynomial speedup for some practical applications [Amb04, Kem03].

The field of quantum walks is very recent and still lacks a solid mathematical foundation. Markov chain quantum walks already started to build these foundations by establishing a direct connection to classical Markov chains using algebraic techniques [Sze04]. However, coined quantum walks are not having the same luck, and it seems that mathematical techniques for random walks simply do not work.

Quantum walks are defined by the application of two unitary operators S and C, where C (coin operator) decides which vertex to move onto, and S (shift operator) performs the actual movement of the walk given the direction decided by C. Ambainis [Amb04], Kempe [Kem03] and Konno [Kon08] give good surveys of this model. There are several studies of this walk for specific graphs. On the line, Ambainis et al. [ABN$^+$01] and Chandrashekar, Srikanth, and Laflamme [CSL08] show that the variance of the induced probability distribution has a quadratic improvement over the classical walk (i.e. for t steps is $\mathcal{O}(t^2)$ and classically $\mathcal{O}(t)$). Konno computed the induced probability distribution using path integrals [Kon03] and via a weak limit theorem [Kon05]. In the hypercube, Kempe [Kem05] shows that the hitting time from one corner to its opposite is exponentially faster, while Moore and Russell [MR02] gives the same speed-up for the

mixing time. For practical applications there are algorithms for hypercubes and grids. For the hypercube, Shenvi et al. [SKW03] gives an algorithm for solving SAT with a quadratic improvement, while Potoček et al. [PGKJ09] gives an improvement of the same algorithm on the success probability. For grids, Ambainis et al. [AKR05] show a quadratic speed-up and presents a general framework for analyzing quantum walks. Also, Ambainis [Amb07] gives an optimal algorithm for element distinctness over the Jhonson graph with a quadratic speed-up.

Quantum walks on the line is probably the most studied quantum walk model. Interest on this matter started in computer science with Ambainis, Bach, Nayak, Vishwanath, and Watrous [ABN$^+$01], where notions of hitting and mixing times were introduced. In the same piece of work, they computed a closed-form formula for the induced probability distribution of a Hadamard walk (i.e. a quantum walk with a Hadamard operator as coin). Furthermore, their formula gives a complete characterization of the amplitudes in the state of the walk in the asymptotic limit.

It is known that the dynamics of the walk is controlled by the coin operator [Kem03]. Thus, depending on the application, a good choice of the coin could make a great difference. This motivated the study of quantum walks on the line moved by a general $SU(2)$ operator, which has four independent variables. However if we consider only the resulting probability distribution, one variable is enough; i.e. any probability distribution resulting from a quantum walk on the line can be simulated by a general rotation around the z axis with parameter θ. Nayak and Vishwanath [NV00] gave an intuitive description of the probability distribution based on the stationary phase method without giving an explicit formula for it and without considering the amplitudes of the state of the walk. Chandrashekar, Srikanth, and Laflamme [CSL08] studied generalized walks using a $SU(2)$ coin operation. They present an approximate formula for the amplitudes of the state of the walk. However, their results were based in numerical experiments rather than a complete analytically deducted formula. Grimmet, Janson, and Scudo [GJS04] showed a ballistic spreading of the walk and they gave an expression for the limit distribution using weak convergence theorems.

Table 2.1: Known results of different coins for walks on the line.

Coin	**Amplitudes of the state**	**Probability distribution**
Hadamard	closed-form [NV00]	closed-form [NV00, Kon03]
$SU(2)$	numerical results [CSL08]	numerical results [CSL08], closed-form [NV00]
Symmetric $SU(2)$	closed-form [this work]	closed-form [this work]
$U(2)$	explicit formula (not closed-form) [Kon03, Kon05]	explicit formula (not closed-form) [Kon05]

2.2 Overview of the Chapter

As a step toward finding mathematical foundations of quantum walks, in this work the following question is being addressed: *Given a graph, what is the probability that a quantum walk arrives at a given vertex after some number steps?* This is a very natural question, and for random walks it can be answered by several different combinatorial arguments [Str05].

The main contribution of this chapter is a closed-form formula[1] for the question above for a general symmetric $SU(2)$ operator for walks on the line (Theorem 2.5.1). Furthermore, the formula characterizes the amplitudes of the state of the walk in the asymptotic limit. In comparison to the previous works mentioned before (Nayak and Vishwanath [NV00], Chandrashekar et al. [CSL08]), the closed-form formulas derived in this work were analytically computed for the amplitudes of the state of the walk (including the induced probability distribution) for a symmetric $SU(2)$ operator (Table 2.1 shows more clearly these differences).

In a seminal work, Konno [Kon03, Kon05] gave explicit expressions for the amplitudes of a $U(2)$ coin, using a discrete path integral method in a clever way. However, these expressions were not in closed-form, as we claim in this work. Furthermore, we show how to compute the errors in the asymptotic approximation, something that was missing from previous works in the literature. To

[1] A quantity $f(n)$ is in closed-form if we can compute it using at most a fixed number of "well-known" standard operations, independent of n [GKP94].

this end, in Section 2.4 a coin operator with parameters that alters the phase of the state of the walk on the line is proposed. The coin operator is inspired by the quantum algorithm for SAT proposed by Hogg [Hog00]. In that work, in order to implement heuristics for quantum algorithms, the author proposed to add parameters to the unitary operation of a search algorithm. This way, the situation is similar to classical algorithms where a tunable set of parameters are adjusted according to the problem. After defining the coin operation, we compute the spectrum of the unitary evolution operator of the walk using Fourier analysis. In Section 2.5, after having obtained the eigenspectrum of the walk, we apply the inverse Fourier transform to obtain the state of the walk in terms of Fourier coefficients. To compute a closed-form solution in the asymptotic limit from the Fourier coefficients, we applied the Euler-Maclaurin formula [Apo99] and the steepest descent method for asymptotic approximation of integrals [Won01]. This method is in fact stronger than the stationary point method from [NV00] and [SKJ08], where the authors use it to study the asymptotics of the resulting probability distribution from coin operators with real eigenvalues. With the steepest descent method we can compute the amplitudes of the state of the walk resulting from any complex unitary operator. In Section 2.5.3, we compute the error terms for the approximations made, which can be derived from the employed methods. Finally, some basic properties of the walk are examined by means of weak convergence theorems [GJS04]. The support of the induced probability distribution of the walk is computed, and then we argue how changing the parameters in the coin operator affects the resulting probability distribution.

2.3 A General Definition

A particle doing a random walk on a graph (V, E) starts in some vertex, and at every time step it chooses randomly to move to one of the neighboring vertices. A quantum walk does the same type of movement, but with an additional degree of freedom known as "chirality" [ABN$^+$01], or just "direction of the walk". The direction guides the walk through the edges of the graph, but it can do so in superposition of all possible directions. The space of directions of the walk is called coin space, in analogy to a coin toss. At each time step the walker chooses

a direction, or superposition of directions, and moves accordingly. If the walker chooses a superposition of directions, it also moves in a superposition of position states on the graph.

In the following the formal definition of discrete time quantum walks on general graphs is presented. First, let $\mathcal{H}_s$ be the Hilbert space of positions with basis states $\{|u\rangle : u \in V\}$, i.e., the walk is moving over the vertices. The coin space $\mathcal{H}_c$ is spanned by basis states $\{|i\rangle : i = 1, 2, \ldots, d\}$ where d is the degree of graph (V, E). For example, let u be the current position of the walk with a set of neighbors $N(u)$ with $|N(u)| \leq d$. Each neighboring vertex is labeled with a number between 1 and d. The quantum walk first selects a neighboring vertex with the coin operation C, e.g., if the state of the walk is $|u\rangle \otimes (|2\rangle + |6\rangle)$ it means that the walk selected neighbors 2 and 6 denoted as $|u_2\rangle$ and $|u_6\rangle$. Then with operator S it moves and the resulting state is $(|u_2\rangle + |u_6\rangle) \otimes (|2\rangle + |6\rangle)$.

Definition 2.3.1. The state of the quantum walker $|\Psi_t\rangle = \sum_v |\psi_t(v)\rangle$ at time t is defined over the joint space $\mathcal{H}_c \otimes \mathcal{H}_s$ with basis states $\{|d, v\rangle : |d\rangle \in \mathcal{H}_c, |v\rangle \in \mathcal{H}_s\}$, where $|\psi_t(v)\rangle = \sum_d \alpha_t^d(v)|d, v\rangle$ and $\alpha_t^d(v)$ is the amplitude at time t in direction d and position v. Also $\sum_{d,v} |\alpha_t^d(v)|^2 = 1$.

Definition 2.3.2. Let $|\Psi_t\rangle$ be the state of the walk at time t as defined in 2.3.1. The probability of finding the walk on vertex v at time t is given by

$$P(v, t) = \langle \psi_t(v)| \, \psi_t(v)\rangle = \sum_d |\alpha_t^d(v)|^2.$$

The quantum walk is defined by the way it moves at each time step. This is captured by the following definition.

Definition 2.3.3. The time evolution of the walk is given by

$$|\Psi_t\rangle = U|\Psi_{t-1}\rangle, \text{ or equivalently } |\Psi_t\rangle = U^t \, |\Psi_0\rangle \, ,$$

where $U = S(C \otimes I)$ is a unitary operator defined on the Hilbert space of the whole system $\mathcal{H}_c \otimes \mathcal{H}_s$, I is the identity matrix acting on $\mathcal{H}_s$, C is the coin operator acting solely in $\mathcal{H}_c$, and S is the shift operator in charge of moving the walker.

According to this definition, the walk first chooses a direction of movement using C, and then moves according to the result with operator S. In order to move, operator S needs to be conditioned on the coin space in the following way,

$$S = \sum_{d,v} |d\rangle\langle d| \otimes |v_d\rangle\langle v|,$$

whose action can be described as $|d, n\rangle \xrightarrow{S} |d, v_d\rangle$ with v_d as the d-th neighbor of v.

The dynamics is handled by the coin operator. Essentially, C is a rotation in $\mathcal{H}_c$ and it is called a "coin" in analogy to random walks. Operator C is chosen arbitrarily and we can define walks with different behavior by modifying C. The most common coins being studied are

$$H = \frac{1}{\sqrt{2}} \sum_{x,y} (-1)^{xy} |x\rangle\langle y| \quad \text{(Hadamard Coin)},$$

$$G = 2|\psi\rangle\langle\psi| - I \quad \text{(Grover Coin)},$$

$$F = \frac{1}{\sqrt{d}} \sum_{x,y} \omega^{xy} |x\rangle\langle y| \quad \text{(Discrete Fourier Transform)},$$

where $|\psi\rangle = F|0\rangle$, d is the dimension of $\mathcal{H}_c$ and $\omega = e^{2\pi/d}$ is a d-th root of unity. Normally the Hadamard coin is used for walks on the line [ABN+01], and Grover's and DFT coins were used to study walks on the hypercube [Kem05, KB06, MR02].

2.3.1 Convergence

Let X_t be a random variable representing the position of the walk at time t distributed according to $P(n, t)$ (Definition 2.3.2). Quantum walks gives rise to a sequence of random variables $\{X_t : t \geq 1\}$ similar to a stochastic process. We say that the sequence converges weakly to a random variable Z if $\lim_{t\to\infty} X_t = Z$ given that $\lim_{t\to\infty} \mathbf{E}[h(X_t)] = \mathbf{E}[h(Z)]$ for all bounded continuous function $h : \mathbb{R} \to \mathbb{R}$. Convergence of random variables indicates that random events settle into a fixed pattern. This concept plays an important role in results such as the

weak law of large numbers and the central limit theorem, and therefore it is of great importance for statistics and stochastic processes.

Weak convergence theorems for lines and d-dimensional grids were developed by Grimmett, Janson, and Scudo [GJS04]. With the help of weak convergence theorems we can calculate the asymptotic probability of the density function associated to the walk when $t \to \infty$. For this thesis, only the convergence theorem for walks on lines is needed [GJS04, Theorem 1].

Let $\mathcal{H} = span\{|d, n\rangle : d \in \{\leftarrow, \rightarrow\} \text{ and } n \in \mathbb{Z}\}$ be the Hilbert space of the walk on the line. The basis states of $\mathcal{H}$ are transformed to Fourier space, and denote this new space as $\mathcal{H}_k$. Define the variable $k \in \mathbb{K} = [0, 2\pi)$ which denotes the Fourier transform of position n on the line. In $\mathcal{H}_k$ a basis state at time t is denoted as $\left|\widetilde{\psi}_t(k)\right\rangle = \sum_d \left|\widetilde{\psi}_t^d(k)\right\rangle$ for some $k \in \mathbb{K}$, where $\left|\widetilde{\psi}_t^d(k)\right\rangle$ is the component of the walk going in direction d. Now define a probability space $\Omega = \mathbb{K} \times \{1, 2\}$ with probability measure $\Delta = |\left\langle \widetilde{\psi}_0(k) \right| \left. \widetilde{\psi}_t^d(k) \right\rangle|^2 dk/2\pi$ on $\mathbb{K} \times \{d\}$. Let U_k be the unitary operation of the walk in $\mathcal{H}_k$ with eigenvalues $\lambda_d(k)$, one for each possible direction over the line. Define a function $h : \Omega \to \mathbb{R}$ as $h(k, d) = \lambda_d(k)^{-1} \frac{d}{dk} \lambda_d(k)$.

Theorem 2.3.1 (Grimmett et al. [GJS04])**.** *Let Z be a random element of Ω with distribution Δ, then*

$$\frac{X_t}{t} \rightsquigarrow h(Z),$$

where the symbol $\rightsquigarrow$ denotes weak convergence.

The support of h is exactly in the range $[\min h, \max h]$ [GJS04]. As an example from [GJS04], consider as coin operation the Hadamard matrix H. The eigenvectors in Fourier space are

$$\lambda_d(k) = \frac{i}{\sqrt{2}} \sin k \pm \sqrt{1 - \frac{1}{2} \sin^2 k}.$$

Then

$$h(k, d) = \frac{-i\lambda_d'(k)}{\lambda_d(k)} = \pm \frac{\cos k}{\sqrt{2 - \sin^2 k}}.$$

The support of h, i.e., the points in the line where the probability is not 0, is concentrated in

$$[\min h, \max h] = \left[-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right].$$

Furthermore, these results can be used to directly compute the asymptotic density function of the walk on the line as shown in [GJS04].

2.3.2 Quantum Walks and Decision Trees

This section presents some algorithmic applications of quantum walks. For a more general analysis with applications see Appendix D.

SAT

We start by describing a quantum walk algorithm for SAT discovered by Shenvi, Kempe, and Whaley [SKW03]. This was possibly the first real application of quantum walks to a computational problem.

The graph used for the walk is a hypercube with 2^n vertices for a SAT formula with n variables. Each vertex is connected to its neighbors if and only if it has Hamming distance 1 from it. The objective is to find a target vertex $|x_{target}\rangle$ that it is promised to exists.

Theorem 2.3.2 (Shenvi, Kempe, Whaley [SKW03])**.** *Given a SAT formula with n variables and a unique solution, there exists a coined quantum walk algorithm that finds the solution in $\mathcal{O}(\sqrt{2^{n/2}})$ steps with success probability $\frac{1}{2} - \mathcal{O}(1/n)$.*

In order to obtain a search algorithm, the walk needs to amplify the amplitude of the target state. To do so, the authors introduce a "perturbed" coin operation C'. The idea is that operator C' acts by applying a "marking" coin C_1 to the target state, and the original coin C_0 to the rest. Each time this operation is used, it is making an oracle call and flipping the phase of the target state.

The algorithm goes as follows:

1. Create a superposition on all vertices;
2. Apply operator $U' = S(C' \otimes I)$, $\mathcal{O}(\sqrt{2^n})$ times;
3. Measure the position register.

One of the key aspects of the proof is the projection of the hypercube graph onto a line. The authors then proceed with the analysis of the walk on the line to prove their claims. For details refer to [SKW03]. The success probability was recently improved to $1 - \mathcal{O}(1/n)$ [PGKJ09].

Element Distincness

Another interesting application is to the *Element Distinctness Problem*. Given a set of positive integers $S = \{x_1, \ldots, x_n\}$, determine if there exists indices $i \neq j$ such that $x_i = x_j$, i.e., a collision. Ambainis [Amb07] proved, using quantum walks, an upper bound of $\mathcal{O}(n^{2/3})$ queries for this problem, matching the lower bound given by Aaronson and Shi [AS04].

In general, this is a walk on the Johnson graph. This graph have as vertices subsets $A \subseteq [n]$ with a fixed size $|A| = r$. A vertex A is connected to another vertex B if and only if they differ in exactly one element, i.e. $|A \cap B| = 1$. Denote this graph by $J_{n,r}$.

Theorem 2.3.3 (Ambainis [Amb07])**.** *There exists a quantum walk algorithm that solves the element distinctness problem with $\mathcal{O}(n^{2/3})$ queries and constant success probability.*

This is a walk on $J_{n,r}$ with $r = n^{2/3}$. The algorithms goes as follows:

1. Create a superposition on all vertices;
2. Make r queries to the current vertex;
3. Repeat $\mathcal{O}((n/r)^{1/2})$ times:
   (a) Apply the conditional phase flip to check for collisions;
   (b) Perform $\mathcal{O}(\sqrt{r})$ steps of the quantum walk.
4. Measure the final state.

This is a very rough idea of the algorithm. For details refer to [Amb07]. The major difference with the walk for SAT is that the flipping operation takes place outside the quantum walk (step 3.a).

2.4 Walks on the Line with Phase Parameters

This section gives the definition of quantum walks on the line and introduces the coin operator used in this research.

Definition 2.4.1. Let $\mathcal{H}_c = span\{|\leftarrow\rangle, |\rightarrow\rangle\}$ and $\mathcal{H}_s = span\{|n\rangle : n \in \mathbb{Z}\}$. The state of the walk $|\Psi_t\rangle = \sum_n |\psi_t(n)\rangle$ at time t is defined over the joint space $\mathcal{H}_c \otimes \mathcal{H}_s$ with basis states $\{|d, n\rangle : |d\rangle \in \mathcal{H}_c, |n\rangle \in \mathcal{H}_s\}$, where $|\psi_t(n)\rangle =$

$\sum_d \alpha_t^d(n)|d,n\rangle$ and $\alpha_t^d(n)$ is the amplitude at time t in direction d and position n. Also $\sum_{d,n} |\alpha_t^d(n)|^2 = 1$.

For the analysis of the walk on the line we consider the projection at time t onto position n as a 2 dimensional vector, i.e.,

$$\begin{bmatrix} \alpha_t^{\leftarrow}(n) \\ \alpha_t^{\rightarrow}(n) \end{bmatrix}$$

with $\alpha_t^{\leftarrow}(n)$ and $\alpha_t^{\rightarrow}(n)$ representing the amplitude of the walker at position n at time t going left and right respectively. The probability of being at position n at time t is thus given by

$$P_t(n) = \langle \psi_t(n)|\psi_t(n)\rangle = |\alpha_t^{\leftarrow}(n)|^2 + |\alpha_t^{\rightarrow}(n)|^2. \tag{2.1}$$

Throughout the paper, the initial condition is considered as $|\psi_0(0)\rangle = [\alpha_0^{\leftarrow}, \alpha_0^{\rightarrow}]^T$ and $|\psi_0(n)\rangle = [0,0]^T$ for $n \neq 0$, with $|\alpha_0^{\leftarrow}|^2 + |\alpha_0^{\rightarrow}|^2 = 1$.

The quantum walk is defined by the way it moves at each time step. This is captured by the following definition.

Definition 2.4.2. The time evolution of the walk on the line is given by

$$|\Psi_t\rangle = U\,|\Psi_{t-1}\rangle\,, \text{ or equivalently, } |\Psi_t\rangle = U^t\,|\Psi_0\rangle\,,$$

where $U = S(C \otimes I)$ is a unitary operator defined on the Hilbert space of the whole system $\mathcal{H}_c \otimes \mathcal{H}_s$, I is the identity matrix acting on $\mathcal{H}_s$, C is the coin operator acting solely on $\mathcal{H}_c$, and S is the shift operator in charge of performing the walk.

According to this definition, the walk first choses a direction of movement using C, and then moves with operator S. In order to move, operator S needs to be conditioned on the coin space in the following way,

$$S = \sum_n |{\leftarrow}\rangle\langle{\leftarrow}| \otimes |n-1\rangle\langle n| + |{\rightarrow}\rangle\langle{\rightarrow}| \otimes |n+1\rangle\langle n|\,. \tag{2.2}$$

Definition 2.4.3. The coin operator is defined by $C = HTH$, where H is the Hadamard operator[2] in charge of mixing amplitudes among states, and $T = e^{i\pi\tau_1}|{\leftarrow}\rangle\langle{\leftarrow}| + e^{i\pi\tau_2}|{\rightarrow}\rangle\langle{\rightarrow}|$ is the diagonal phase adjustments with $\tau_1, \tau_2 \in [0,1]$.

[2]The Hadamard operator is defined as $H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$.

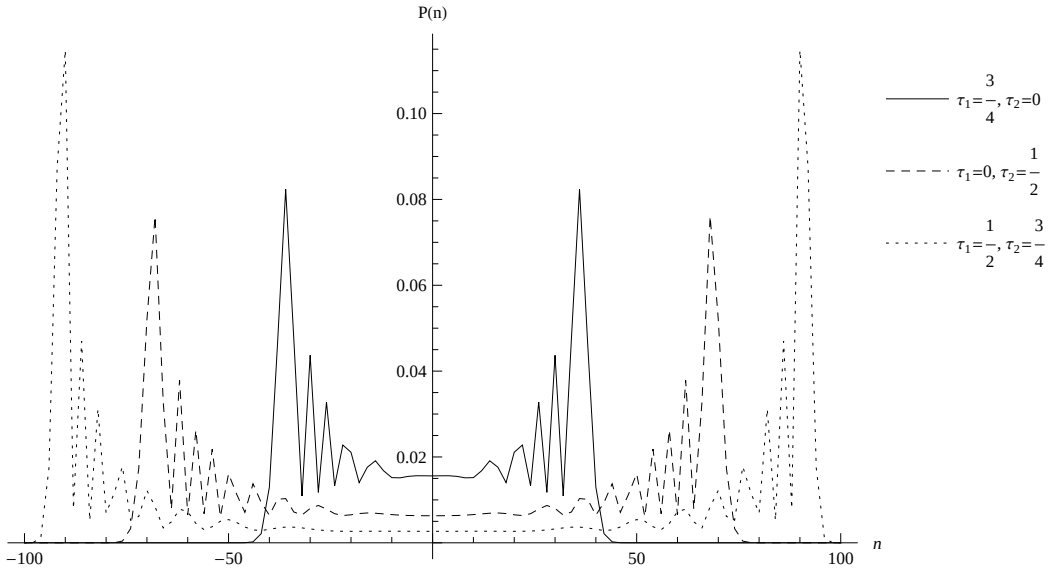

Figure 2.1: Quantum walk on the line with different values of phase parameters. The variance of the walk changes depending on τ_1 and τ_2. Since the probabilities at odd positions are 0, those points are not plotted.

Let $a \equiv e^{i\pi\tau_1} + e^{i\pi\tau_2}$ and $b \equiv e^{i\pi\tau_1} - e^{i\pi\tau_2}$. Then, the resulting operator can be written as

$$C = \frac{1}{2}\begin{bmatrix} a & b \\ b & a \end{bmatrix},$$

which have the following effect on $\mathcal{H}_C$

$$|\leftarrow\rangle \longrightarrow (1/2)a|\leftarrow\rangle + (1/2)b|\rightarrow\rangle,$$
$$|\rightarrow\rangle \longrightarrow (1/2)b|\leftarrow\rangle + (1/2)a|\rightarrow\rangle.$$

Figure 2.1 shows the dynamics of a walk using C as coin. For different values of the phase parameters τ_1 and τ_2 the variance of the induced probability distribution changes.

The state of the walk at time t can be related to the state at time $t+1$ according to the following lemma.

Lemma 2.4.1.

$$|\psi_{t+1}(n)\rangle = M_+|\psi_t(n-1)\rangle + M_-|\psi_t(n+1)\rangle \tag{2.3}$$

where

$$M_+ = \begin{bmatrix} 0 & 0 \\ (1/2)b & (1/2)a \end{bmatrix} \text{ and } M_- = \begin{bmatrix} (1/2)a & (1/2)b \\ 0 & 0 \end{bmatrix}.$$

Proof. Let $|\Psi_t\rangle = \sum_n \alpha_t^{\leftarrow}(n)|\leftarrow, n\rangle + \alpha_t^{\rightarrow}(n)|\rightarrow, n\rangle$ be the state at time t. Also denote the amplitudes after applying operators C and S as

$$(C \otimes I)\,|\Psi_t\rangle = \sum_n \alpha_t^{\leftarrow}(n)'|\leftarrow, n\rangle + \alpha_t^{\rightarrow}(n)'|\rightarrow, n\rangle,$$
$$S(C \otimes I)|\Psi_t\rangle = \sum_n \alpha_t^{\leftarrow}(n)''|\leftarrow, n\rangle + \alpha_t^{\rightarrow}(n)''|\rightarrow, n\rangle.$$

Now let $|\Psi_{t+1}\rangle = \begin{bmatrix} \alpha_{t+1}^{\leftarrow}(n) \\ \alpha_{t+1}^{\rightarrow}(n) \end{bmatrix}$ be the state at time $t+1$. The amplitudes of this state are related to the amplitudes of $|\Psi_t\rangle$ in the following way

$$\begin{bmatrix} \alpha_{t+1}^{\leftarrow}(n) \\ \alpha_{t+1}^{\rightarrow}(n) \end{bmatrix} = \begin{bmatrix} \alpha_t^{\leftarrow}(n)'' \\ \alpha_t^{\rightarrow}(n)'' \end{bmatrix} = \begin{bmatrix} \alpha_t^{\leftarrow}(n+1)' \\ \alpha_t^{\rightarrow}(n-1)' \end{bmatrix}.$$

The contributions to the amplitudes of state $|\Psi_{t+1}\rangle$ come from position $n+1$ for the upper component, and from $n-1$ for the lower component by definition of operator S. The amplitudes corresponding to the state after applying C are computed as follows:

$$\begin{aligned} C|\psi_t(n+1)\rangle &= \begin{bmatrix} (1/2)a\alpha_t^{\leftarrow}(n+1) + (1/2)b\alpha_t^{\rightarrow}(n+1) \\ (1/2)b\alpha_t^{\leftarrow}(n+1) + (1/2)a\alpha_t^{\rightarrow}(n+1) \end{bmatrix} \\ &= \begin{bmatrix} \alpha_t^{\leftarrow}(n+1)' \\ \alpha_t^{\rightarrow}(n+1)' \end{bmatrix}, \end{aligned}$$

and the same for $C\,|\psi_t(n-1)\rangle$. Thus

$$\begin{aligned} |\psi_{t+1}(n)\rangle &= \begin{bmatrix} (1/2)a\alpha_t^{\leftarrow}(n+1) + (1/2)b\alpha_t^{\rightarrow}(n+1) \\ (1/2)b\alpha_t^{\leftarrow}(n-1) + (1/2)a\alpha_t^{\rightarrow}(n-1) \end{bmatrix} \\ &= M_+|\psi_t(n-1)\rangle + M_-|\psi_t(n+1)\rangle, \end{aligned}$$

where

$$M_+ = \begin{bmatrix} 0 & 0 \\ (1/2)b & (1/2)a \end{bmatrix} \text{ and } M_- = \begin{bmatrix} (1/2)a & (1/2)b \\ 0 & 0 \end{bmatrix}.$$

□

2.4.1 Analysis

One approach to the analysis of quantum processes is the path integral approach. This method explicitly computes the amplitude of a certain state as the sum over all possible paths leading to that state [ABN+01, Kon03]. Solving a path integral is known to be hard, and we avoid this by following the steps of [ABN+01, MR02, Kem05] known as the *Schrödinger approach*. Given the translational invariance of the walk, it has a simple description in Fourier space [ABN+01]. The Fourier transform of the walk is analyzed and then transformed back to the original domain.

The quantum Fourier transform [NC00] of a wave equation is defined by

$$\left|\widetilde{\psi}_t(k)\right\rangle = \sum_n e^{ikn} \left|\psi_t(n)\right\rangle, \tag{2.4}$$

and the corresponding inverse Fourier transform is then

$$\left|\psi_t(n)\right\rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-ikn} \left|\widetilde{\psi}_t(k)\right\rangle dk. \tag{2.5}$$

Applying (2.4) to (2.3) we get

$$\begin{aligned}
\left|\widetilde{\psi}_{t+1}(k)\right\rangle &= \sum_n e^{ikn} M_+ |\psi_t(n-1)\rangle + e^{ikn} M_- |\psi_t(n+1)\rangle \\
&= e^{ik} M_+ \sum_n e^{ik(n-1)} \left|\psi_t(n-1)\right\rangle \\
&\qquad + e^{-ik} M_- \sum_n e^{ik(n+1)} \left|\psi_t(n+1)\right\rangle \\
&= e^{ik} M_+ \left|\widetilde{\psi}_t(k)\right\rangle + e^{-ik} M_- \left|\widetilde{\psi}_t(k)\right\rangle \\
&= \left(e^{ik} M_+ + e^{-ik} M_-\right) \left|\widetilde{\psi}_t(k)\right\rangle.
\end{aligned}$$

Then, the time-evolution in Fourier space is given by

$$\left|\widetilde{\psi}_{t+1}(k)\right\rangle = M_k \left|\widetilde{\psi}_t(k)\right\rangle \tag{2.6}$$

where $M_k = e^{ik} M_+ + e^{-ik} M_-$. In matrix form

$$M_k = \frac{1}{2} \begin{bmatrix} a e^{-ik} & b e^{-ik} \\ b e^{ik} & a e^{ik} \end{bmatrix}. \tag{2.7}$$

In general, the state at time t is given by the t-th power of operator M_k applied to the initial state

$$\left|\widetilde{\psi}_t(k)\right\rangle = M_k^t \left|\widetilde{\psi}_0(k)\right\rangle. \tag{2.8}$$

The following lemma shows the eigenspectrum of operator M_k.

Lemma 2.4.2. *Let M_k be a unitary matrix as in (2.7). The eigenvalues and eigenvectors of M_k are*

$$\lambda_j(k) = 1/2\left(a\cos k \pm \sqrt{b^2 - a^2\sin^2 k}\right)$$

and

$$|\lambda_j(k)\rangle = N_j(k)\begin{bmatrix} -ia\sin k \pm \sqrt{b^2 - a^2\sin^2 k} \\ be^{ik} \end{bmatrix}$$

respectively, with $j = 1, 2$. Furthermore, $N_j(k)$ is a normalization coefficient given by

$$N_j(k) = \left(\left|-ai\sin k \pm \sqrt{b^2 - a^2\sin^2 k}\right|^2 + |b|^2\right)^{-1/2}.$$

Proof. The characteristic polynomial of M_k is determined by $det(M_k - \lambda I) = 0$. Then

$$det(M_k - \lambda I) = \lambda^2 - a\lambda\cos k + \frac{a^2}{4} - \frac{b^2}{4}.$$

Solving the equation gives the eigenvalues

$$\lambda_j(k) = \frac{a\cos k \pm \sqrt{b^2 - a^2\sin^2 k}}{2},$$

for $j = 1, 2$. In order to find the eigenvectors, we solve the following system of linear equations

$$(M_k - \lambda_j(k)I)\begin{bmatrix} x_j \\ y_j \end{bmatrix} = \begin{bmatrix} x_j\left(\frac{a}{2}e^{-ik} - \lambda_j(k)\right) + y_j\frac{b}{2}e^{-ik} \\ x_j\frac{b}{2}e^{ik} + y_j\left(\frac{a}{2}e^{ik} - \lambda_j(k)\right) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

By letting $y_j = 1$, we get $x_j = (-a + 2\lambda_j e^{-ik})/b$. Given that any multiple of this vector is still an eigenvector, multiply y_j and x_j by be^{ik} and obtain

$$be^{ik}\begin{bmatrix} x_j \\ y_j \end{bmatrix} = \begin{bmatrix} -ae^{ik} + 2\lambda_j \\ be^{ik} \end{bmatrix} = \begin{bmatrix} -ai\sin k \pm \sqrt{b^2 - a^2\sin^2 k} \\ be^{ik} \end{bmatrix}.$$

Then $N_j(k)$ is 1 divided by the ℓ_2-norm of this vector, and multiply the eigenvectors by $N_j(k)$ to normalize them. □

Diagonalize (2.7) to obtain

$$M_k^t = \sum_{j\in\{1,2\}} \lambda_j(k)^t |\lambda_j(k)\rangle\langle\lambda_j(k)|,$$

where $\lambda_1(k)$ and $\lambda_2(k)$ are the eigenvalues with corresponding eigenvectors $|\lambda_1(k)\rangle$ and $|\lambda_2(k)\rangle$. Now apply the diagonalized operator to the time evolution (2.8) and obtain the following form

$$\begin{aligned} |\widetilde{\psi}_t(k)\rangle &= \sum_j \left(\lambda_j(k)^t \, |\lambda_j(k)\rangle \, \langle\lambda_j(k)|\right) \left|\widetilde{\psi}_0(k)\right\rangle \\ &= \sum_j \left\langle \lambda_j(k)\right| \left.\widetilde{\psi}_0(k)\right\rangle \lambda_j(k)^t \, |\lambda_j(k)\rangle \, . \end{aligned} \tag{2.9}$$

The initial state is $[\alpha_0^{\leftarrow}, \alpha_0^{\rightarrow}]^T$, and in Fourier space becomes $|\widetilde{\psi}_0(k)\rangle = [\alpha_0^{\leftarrow}, \alpha_0^{\rightarrow}]^T$ for all $k \in [-\pi, \pi]$. To write equation (2.9) in a simpler way define

$$\begin{aligned} \xi_j(k) &= \left\langle \lambda_j(k)\right| \left.\widetilde{\psi}_0(k)\right\rangle \\ &= \alpha_0^{\leftarrow} N_j(k) \left(-ia\sin k \pm \sqrt{b^2 - a^2\sin^2 k}\right)^* + \alpha_0^{\rightarrow} N_j(k) b^* e^{-ik}, \end{aligned} \tag{2.10}$$

where $*$ is the complex conjugate. This can be expressed in matrix form as

$$\begin{bmatrix} \xi_1(k) \\ \xi_2(k) \end{bmatrix} = \begin{bmatrix} (-ia\sin k + \sqrt{b^2 - a^2\sin^2 k})^* & b^* e^{-ik} N_1(k) \\ (-ia\sin k - \sqrt{b^2 - a^2\sin^2 k})^* & b^* e^{-ik} N_2(k) \end{bmatrix} \cdot \begin{bmatrix} \alpha_0^{\leftarrow} \\ \alpha_0^{\rightarrow} \end{bmatrix}.$$

The state of the walk at time t can be expressed by

$$\left|\widetilde{\psi}_t(k)\right\rangle = M_k^t \left|\widetilde{\psi}_0(k)\right\rangle = \sum_j \lambda_j^t(k)\xi_j(k) \, |\lambda_j(k)\rangle \, . \tag{2.11}$$

Let $\widetilde{\alpha}_t^{\leftarrow}(k)$ and $\widetilde{\alpha}_t^{\rightarrow}(k)$ be the amplitudes of the state $\left|\widetilde{\psi}_t(k)\right\rangle$ in Fourier space going left and right respectively. Then, by equation (2.11) and Lemma 2.4.2 these amplitudes are

$$\widetilde{\alpha}_t^{\leftarrow}(k) = \sum_j \lambda_j(k)^t \xi_j(k) N_j(k) \left(-ia\sin k \pm \sqrt{b^2 - a^2\sin^2 k}\right) \tag{2.12}$$

and

$$\widetilde{\alpha}_t^{\rightarrow}(k) = \sum_j \lambda_j(k)^t \xi_j(k) N_j(k) b e^{ik}. \tag{2.13}$$

The final step is to reverse back to the original domain of the walk. This is done by applying (2.5) to (2.12) and (2.13),

$$\alpha_t^{\leftarrow}(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\sum_j \xi_j(k)N_j(k)\lambda_j^t e^{-ikn}\left(-ia\sin k \pm \sqrt{b^2 - a^2\sin^2 k}\right)dk \quad (2.14)$$

and

$$\alpha_t^{\rightarrow}(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\sum_j \xi_j(k)N_j(k)be^{ik}\lambda_j^t e^{-ikn}dk, \quad (2.15)$$

Note that a discrete walk is being approximated by an integral. The Euler-Maclaurin summation formula[3] gives the error term for these approximations.

Equations (2.14) and (2.15) can be solved by the steepest descent method from complex analysis, obtaining this way closed-form solutions. This is done in the next section.

2.5 Asymptotic Approximation

In this section it is shown how to find close-form solutions to the integrals (2.14) and (2.15). First, in Section 2.5.1 the technique used in this research known as the steepest descent method is briefly explained. Then, in Section 2.5.2 the same technique is applied to the integral-forms of the walk (equations (2.14) and (2.15)).

2.5.1 Steepest Descent Method

Here one of the most powerful methods for asymptotic approximation of integrals is briefly explained. The method is known as Steepest Descent Approximation or Saddle Point Method. For a deeper understanding on this technique refer to [Won01].

The method of steepest descent is an asymptotic approximation method for certain types of exponential integrals of the form

$$I_t = \int_{\mathcal{C}} g(z)e^{tf(z)}dz \quad (2.16)$$

[3] $\sum_{n=a}^{b} f(n) = \int_a^b f(x)dx + \frac{f(a)+f(b)}{2} + \sum_{k=1}^{\infty}\frac{B_{2k}}{(2k)!}(f^{(2k-1)}(b) - f^{(2k-1)}(a))$, where each B_i is a Bernoulli number [Apo99].

where $\mathcal{C}$ is a contour in the complex z-plane and $g(z)$ and $f(z)$ are complex-valued analytic functions. The parameter t is taken to be real and positive, and we are interested in the asymptotic behavior of (2.16) as $t \to \infty$ with $t > 0$. Laplace's and stationary phase methods are just instances of this general procedure. The integral is dominated by the highest stationary points of f, i.e., if $f(z) = u(x, y) + iv(x, y)$ with $z = x + iy$ we expect the integral to be dominated by points where u is maximum and v is constant. The only possible extrema for f are the *saddle points* where $f'(z) = 0$. Since f is analytic, u and v satisfy the Cauchy-Riemann equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0,$$

and from the maximum principle [Won01] we have that if $\frac{\partial^2 u}{\partial x^2} > 0$ then $\frac{\partial^2 u}{\partial y^2} < 0$ or vice versa. If z_0 is the saddle point, then we can deform the contour to $\mathcal{C}'$ (by Cauchy's theorem) so that it passes through z_0. From the Taylor expansion of $f(z)$ about z_0 we have

$$f(z) \sim f(z_0) + \frac{1}{2} f''(z_0)(z - z_0)^2,$$

where $\sim$ means "is close up to additive error to". Then $g(z) \sim g(z_0)$, because for large t the main contribution to the integral comes from f. Then I_t becomes

$$I_t \sim g(z_0) e^{tf(z_0)} \int_{\mathcal{C}'} e^{\frac{1}{2} t f''(z_0)(z - z_0)^2} dz.$$

Setting

$$z - z_0 = re^{i\phi} \quad \text{and} \quad f''(z_0) = |f''(z_0)|\, e^{i\theta}$$

it can be seen that

$$I_t \sim g(z_0) e^{tf(z_0)} \int_{\mathcal{C}'} \exp(\frac{1}{2} t\, |f''(z_0)|\, e^{i\theta + 2i\phi} r^2) e^{i\phi} dr.$$

Note that ϕ is the angle of inclination of the oriented tangent to $\mathcal{C}$ at point z_0, i.e., $\phi = \arg(z_0)$ on $\mathcal{C}$ [Won01]. Choosing $\theta + 2\phi = \pi$, i.e., $\phi = (\pi - \theta)/2$ then

$$I_t \sim g(z_0) e^{tf(z_0)} e^{i\phi} \int_{\mathcal{C}'} e^{-\frac{1}{2} t |f''(z_0)| r^2} dr$$

and solving this as a Gaussian integral[4] yields

$$I_t = g(z_0)e^{tf(z_0)}e^{i\phi}\left(\frac{2\pi}{t\,|f''(z_0)|}\right)^{1/2} + O(t^{-1}). \tag{2.17}$$

The deformation of the contour chosen to make the integration Gaussian corresponds to the steepest descent path from the saddle point, hence the name of the method [Won01]. Taking this path is not essential, other methods like stationary point and Perron's method take another path with similar results [Won01].

2.5.2 Asymptotic Approximation of the Walk on the Line

Left Amplitude

First the integral-form corresponding to equation (2.14) is solved. Write the integral in the form of equation (2.16) by setting $n = \gamma t$ ($\gamma = n/t$) and writing

$$\alpha_t^{\leftarrow}(\gamma t) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\sum_j g_j(k)e^{tf_j(k)} \tag{2.18}$$

where

$$f_j(k) = \log\lambda_j(k) - ik\gamma, \tag{2.19}$$

$$g_j(k) = N_j(k)\xi_j(k)\left(-ia\sin k \pm \sqrt{b^2 - a^2\sin^2 k}\right). \tag{2.20}$$

The saddle points θ_j of $f_j(k)$ are defined by the equation

$$f_j'(\theta_j) = -i\gamma \mp \frac{a\sin\theta_j}{\sqrt{b^2 - a^2\sin^2\theta_j}} = 0.$$

This equation has a solution at

$$\theta_j = \pm\arcsin\left(\frac{b\gamma}{a\sqrt{\gamma^2 - 1}}\right). \tag{2.21}$$

Also note that $|\lambda_j(\theta_j)| = 1$. Moreover

$$f_j(\theta_j) = -i\gamma\theta_j + \log\left(\frac{\pm b + \sqrt{a^2(1-\gamma^2) + b^2\gamma^2}}{2\sqrt{1-\gamma^2}}\right) \tag{2.22}$$

[4]The Gaussian integral or probability integral is given by $\int_{-\infty}^{\infty} e^{-x^2}dx = \sqrt{\pi}$.

and

$$f_j''(\theta_j) = \frac{\pm(\gamma^2-1)\sqrt{b^2\gamma^2 + a^2(1-\gamma^2)}}{b}. \tag{2.23}$$

Another solution to the equation $f'(\theta_j) = 0$ is at $-\pi - \theta_j$ in the interval $[-\pi, \pi]$. However, since $f''(\theta_j)$ and $f''(-\pi - \theta_j)$ have similar behavior, the computations do not change.

The contour is the real line in $[-\pi, \pi]$ and has no imaginary part, therefore $\phi = \arg\theta_j = 0$ in equation (2.17).

Now using (2.17), the asymptotic expansion can be obtained

$$\begin{aligned}
\alpha_t^{\leftarrow}(\gamma t) &= \frac{1}{2\pi}\sum_j g_j(\theta_j) e^{tf_j(\theta_j)} \left(\frac{2\pi}{t|f_j''(\theta_j)|}\right)^{1/2} + O(t^{-1}) \\
&= \frac{1}{2\pi}\sum_j N_j(\theta_j)\xi_j(\theta_j) \left[\frac{\pm b(1-\gamma)}{\sqrt{1-\gamma^2}}\right] \\
&\quad \times \left(\frac{\pm b + \sqrt{a^2(1-\gamma^2)+b^2\gamma^2}}{2\sqrt{1-\gamma^2}}\right)^t e^{-i\gamma\theta_j t} \\
&\quad \times \left(\frac{2\pi|b|}{t|\gamma^2-1|\sqrt{b^2\gamma^2+a^2(1-\gamma^2)}}\right)^{1/2} + O(t^{-1}).
\end{aligned}$$

Right Amplitude

Next is the solution of equation (2.15). Following the same steps as above, write the integral as

$$\alpha_t^{\rightarrow}(\gamma t) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\sum_j h_j(k) e^{tf_j(k)}, \tag{2.24}$$

where f_j is defined in the same way as in (2.19), and

$$h_j(k) = N_j(k)\xi_j(k)be^{ik}. \tag{2.25}$$

Reusing the previous calculations for f_j (equations (2.21), (2.22) and (2.23)),

the asymptotic expansion is

$$\begin{aligned}\alpha_t^{\rightarrow}(\gamma t) &= \frac{1}{2\pi}\sum_j h_j(\theta_j)e^{tf_j(\theta_j)}\left(\frac{2\pi}{t|f_j''(\theta_j)|}\right)^{1/2} + O(t^{-1}) \\ &= \frac{1}{2\pi}\sum_j N_j(\theta_j)\xi_j(\theta_j)be^{i\theta_j} \\ &\quad \times \left(\frac{\pm b + \sqrt{a^2(1-\gamma^2)+b^2\gamma^2}}{2\sqrt{1-\gamma^2}}\right)^t e^{-i\gamma\theta_j t} \\ &\quad \times \left(\frac{2\pi|b|}{t|\gamma^2-1|\sqrt{b^2\gamma^2+a^2(1-\gamma^2)}|}\right)^{1/2} + O(t^{-1})\end{aligned}$$

2.5.3 Closed-form Formulas and Convergence

Formulas

Approximate closed-forms for the amplitudes of the state of the walk on the line were given. Now the main contribution of this paper can be stated formally.

Theorem 2.5.1. *Let $\gamma = n/t$ and $a \equiv e^{i\pi\tau_1} + e^{i\pi\tau_2}, b \equiv e^{i\pi\tau_1} - e^{i\pi\tau_2}$. If the state of the walk is*

$$|\Psi_t\rangle = \sum_n |\psi_t(n)\rangle \;\; \textit{with} \;\; |\psi_t(n)\rangle = \begin{bmatrix}\alpha_t^{\leftarrow}(n) \\ \alpha_t^{\rightarrow}(n)\end{bmatrix}$$

then,

$$\begin{aligned}\alpha_t^{\leftarrow}(\gamma t) &\sim \frac{1}{2\pi}\sum_j N_j\xi_j A_j\left[\frac{\pm b(1-\gamma)}{\sqrt{1-\gamma^2}}\right], \\ \alpha_t^{\rightarrow}(\gamma t) &\sim \frac{1}{2\pi}\sum_j N_j\xi_j A_j be^{i\theta_j},\end{aligned}$$

where the terms A_j, N_j, ξ_j and θ_j are given by

$$
\begin{aligned}
A_j &= \left(\frac{\pm b + \sqrt{a^2(1-\gamma^2) + b^2\gamma^2}}{2\sqrt{1-\gamma^2}} \right)^t \\
&\quad \times \left(\frac{2\pi|b|}{t|\gamma^2-1|\sqrt{b^2\gamma^2 + a^2(1-\gamma^2)}} \right)^{1/2} e^{-i\gamma\theta_j t}, \\
N_j &= \left(\left| -ia\sin\theta_j \pm \sqrt{b^2 - a^2\sin^2\theta_j} \right|^2 + |b|^2 \right), \\
\xi_j &= \alpha_0^{\leftarrow}(0) \left(-ia\sin\theta_j \pm \sqrt{b^2 - a^2\sin^2\theta_j} \right)^* \\
&\quad + \alpha_0^{\rightarrow}(0) b^* e^{-i\theta_j}, \\
\sin\theta_j &= \pm \left(\frac{b\gamma}{a\sqrt{\gamma^2-1}} \right),
\end{aligned}
$$

with $\alpha_0^{\leftarrow}(0)$ and $\alpha_0^{\rightarrow}(0)$ as the initial amplitudes of the walk for $n = 0$, and $\alpha_0^{\leftarrow}(n) = \alpha_0^{\rightarrow}(n) = 0$ for $n \neq 0$.

In a seminal work, Konno [Kon03, Kon05] gave explicit expressions for the amplitudes of a $U(2)$ coin using a discrete path integral method. However, these expressions were not in closed-form, as it is claimed in this work.

In order to assess the quality of the approximation, Figures 2.2 and 2.3 show a comparison between the probability distributions given by Theorem 2.5.1, and a numerical simulation of walks that start with an equal superposition of directions for different values of the parameters. It can be seen that the approximation gives some errors, however the asymptotic agrees with the simulation. The figures show that Theorem 2.5.1 is close to the real values of the probability distribution, in particular in the middle part of the plots.

The errors in the approximation made by Theorem 2.5.1 can be computed from two parts, the Euler-Maclaurin formula and the steepest descent method [Won01]. Denote these errors by ϵ and ε respectively. Let $B_i = \sum_{r=0}^{i} \binom{i}{r} B_{i-r}$ be a Bernoulli number [Apo99], and let $d \in \{\leftarrow, \rightarrow\}$. Then, the error for $\alpha_t^d(\gamma t)$ is $\sum_j \epsilon_{j,d} + \varepsilon_{j,d}$, where

$$
\epsilon_{j,d} = \sum_{m=1}^{\infty} \frac{B_{2m}}{(2m)!} \left(\frac{\partial^{2m-1}}{\partial k^{2m-1}} \widetilde{\alpha}_t^d(\pi) - \frac{\partial^{2m-1}}{\partial k^{2m-1}} \widetilde{\alpha}_t^d(-\pi) \right) \tag{2.26}
$$

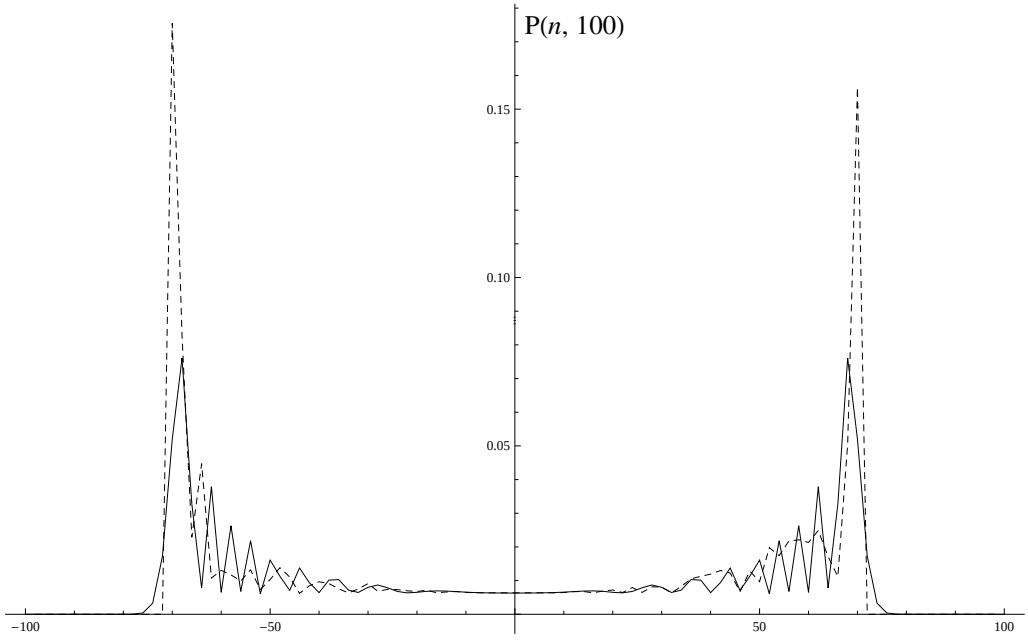

Figure 2.2: Comparison between the probability distributions of numerical simulation (dark) and Theorem 2.5.1 (dashed) with $\tau_1 = 1/2$ and $\tau_2 = 0$, $t = 100$, and initial state in equal superposition of directions.

and

$$\varepsilon_{j,d} = \frac{1}{2\pi} \sum_j e^{tf_j(\theta_j)} \left(\frac{2\pi}{t|f_j''(\theta_j)|} \right)^{1/2} \times \left(\sum_{m=1}^{\infty} \frac{(-1)^m}{m!} \left(\frac{1}{2t|f_j''(\theta_j)|} \right)^m \frac{\partial^{2m}}{\partial k^{2m}} \rho_j(\theta_j) \right), \tag{2.27}$$

where ρ_j is either equation (2.20) if $d = \leftarrow$, or (2.25) if $d = \rightarrow$. It can be seen that if we take m terms from each summation $\epsilon_{j,d} = \mathcal{O}(2^{-m})$ and $\varepsilon_{j,d} = \mathcal{O}(t^{-m})$.

Convergence and Properties

For quantum walks on the line and n-dimensional grids there exists weak convergence theorems [GJS04]. In this section, we state the weak convergence of quantum walks on the line with phase parameters using these previous results. Then we show some applications of the convergence to compute the support of the probability density function.

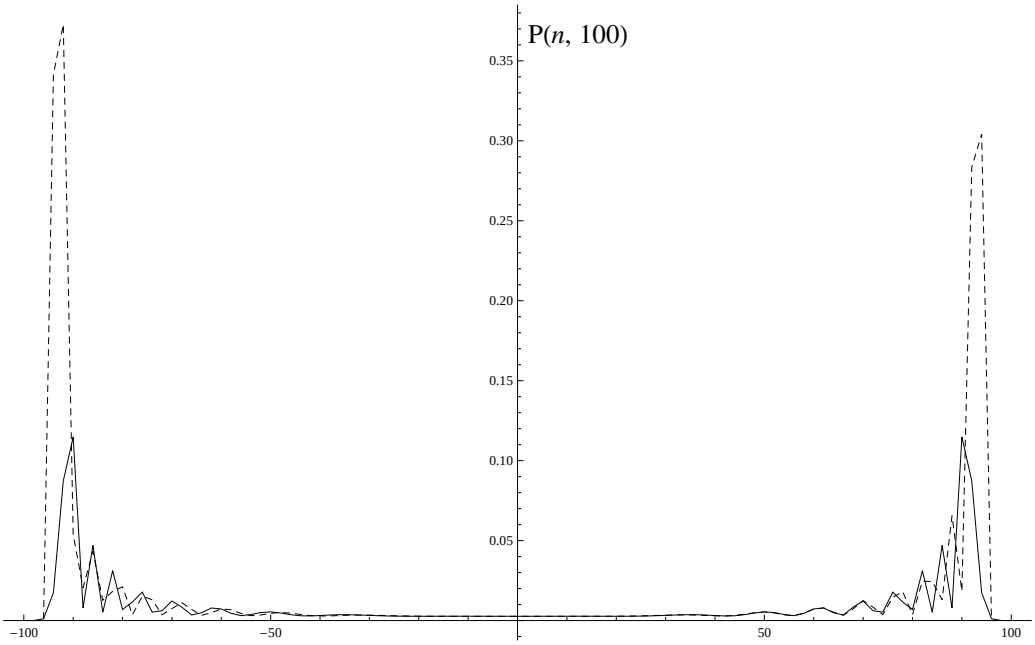

Figure 2.3: Comparison between the probability distributions of numerical simulation (dark) and Theorem 2.5.1 (dashed) with $\tau_1 = 3/4$ and $\tau_2 = 1/2$, $t = 100$, and initial state in equal superposition of directions.

Theorem 2.5.2. *Let $\Omega = [-\pi, \pi] \times \{1, 2\}$ be a probability space with probability measure $\Delta = |\langle\widetilde{\psi}_0(k)|\lambda_j(k)\rangle|^2 dk/2\pi$ for $k \in [-\pi, \pi]$ and $j = 1, 2$. Define a map $h : \Omega \to \mathbb{R}$ such that for $(k, j) \in \Omega$*

$$h(k, j) \equiv h_j(k) = (-1)^j \frac{\sin k}{\sqrt{\sin^2 k + \tan^2 \frac{\pi}{2}(\tau_1 - \tau_2)}}.$$

Let X_t be a position of the quantum walk at time t with distribution given by (2.1), and Z be a random variable of Ω with distribution Δ. Then we have as $t \to \infty$

$$\frac{X_t}{t} \rightsquigarrow h(Z),$$

where $\rightsquigarrow$ denotes weak convergence.

Proof. Consider the theorem that states the weak convergence of quantum walks on the line (Theorem 2.3.1). Let $\lambda_j(k)$ be as in Lemma 2.4.2. Then

$$\lambda_j'(k) = \frac{-a \sin k}{2} - \frac{a^2 \cos k \sin k}{2\sqrt{b^2 - a^2 \sin^2 k}}.$$

Dividing this by $\lambda_j(k)$ we obtain

$$\frac{-i\lambda_j'(k)}{\lambda_j(k)} = (-1)^{j+1}\frac{ai\sin k}{\sqrt{b^2 - a^2\sin^2(k)}}.$$

Then, after some algebra and observing that $\frac{b}{a} = e^{i\pi/2}\tan\frac{\pi}{2}(\tau_1 - \tau_2)$, the theorem follows. □

As an application of Theorem 2.5.2, we can calculate the position of the two peaks of the walk for large time.

Corollary 2.5.3. *The limit distribution of X_t/t is concentrated on the interval $\left[-\frac{|a|}{2}, \frac{|a|}{2}\right]$.*

Proof. Theorem 2.5.2 have its maximum and minimum values for $k = \pm\pi/2$ and the corollary follows. □

The maximum probability of $P_t(n)$ is found at the top of these two peaks, i.e., where $n = \pm|a|/2$ [GJS04]. Considering $|n/t|$ as the speed of the peaks, it can be seen that by setting $\tau_1 = \tau_2$ it gets its maximum value, i.e., the fastest spreading of the walk. This corresponds exactly to an identity operator, and the walk does not mix at all inside the range of Corollary 2.5.3. In order to get high speed and maximum randomness (i.e. the best mixing for positions inside the range) for $P_t(n)$, we can set any value such that $|\tau_1 - \tau_2| = 1/2$. This implies that the support of h is in $[-1/\sqrt{2}, 1/\sqrt{2}]$. In this case, the operator simulates exactly the probability distribution of a Hadamard operator [GJS04].

As another application of Theorem 2.5.2, we can compute the density function of the random variable $Y = X_t/t$ in the asymptotic limit when $t \to \infty$. Following the steps of [GJS04] for the Hadamard coin, we differentiate the quantity

$$P(Y \le y) = \sum_j \int_{k\in[-\pi,\pi]:h_j(k)\le y} \left|\left\langle \widetilde{\psi}_0(k)\right| \lambda_j(k)\right\rangle\Big|^2 \frac{dk}{2\pi}, \tag{2.28}$$

which yields the density function

$$f(y) = \frac{|b|/2}{\pi(y^2-1)\sqrt{(|a|/2)^2 - y^2}} \tag{2.29}$$

for $y \in (-|a|/2, |a|/2)$, under the assumption of $Im(\alpha_0^{\leftarrow}\cdot\alpha_0^{\rightarrow *})\sin(\tau_1-\tau_2)\pi = 0$ and $|\alpha_0^{\leftarrow}| = |\alpha_0^{\rightarrow}| = 1/\sqrt{2}$, which agrees with [Kon03, Kon05].

2.6 Concluding Remarks of the Chapter

This chapter presented a study of discrete-time quantum walks on the line. A symmetric $SU(2)$ coin operation was proposed and analyzed as a step towards an understanding of quantum walks. Using Fourier analysis and asymptotic approximation methods, we computed a closed-form formula for the amplitudes of the state of the walk. With this formula, we have a direct way to compute the amplitudes at any time step without recurring to time-consuming simulations or numerical integration. This also give us a complete characterization of the induced probability distribution of general quantum walks on the line.

One important question that remains unanswered is the relation between Theorems 2.5.1 and 2.5.2. Theorem 2.5.1 is based on the computation of saddle points of the high oscillatory kernel of Fourier coefficients. On the other hand, Theorem 2.5.2 is based on the method of moments (see [GJS04] for details). A relation between these two density functions could set a common ground for the analysis of coined quantum walks.

Chapter 3

Strong Quantum Nondeterministic Communication

Contents

3.1 Background

Nondeterminism plays a fundamental role in complexity theory. For instance, the **P** vs **NP** problem asks if nondeterministic polynomial time is strictly more powerful than deterministic polynomial time. Even though nondeterministic models are unrealistic, they can give insights into the power and limitations of realistic models (i.e., deterministic, random, etc.).

There are two ways of defining a nondeterministic machine, using randomness or as a proof system: a nondeterministic machine *i*) accepts a correct input with positive probability and rejects an incorrect input with probability one; or *ii*) is a deterministic machine that receives besides the input, a proof or certificate which exists if and only if the input is correct. For classical machines (i.e., machines based on classical mechanics), these two notions of nondeterminism are equivalent. However, in the quantum setting they can be different. In fact, these two notions give rise to three different kinds of quantum nondeterminism. In *strong quantum nondeterminism*, the quantum machine accepts a correct input with positive probability. In *weak quantum nondeterminism*, the quantum machine outputs the correct answer when supplied with a correct proof, which could be either classical or quantum. Indeed, as efficient computation is concerned, the corresponding complexity classes are exactly **NQP**, **QMA**, and **QCMA** respectively[1].

The study of quantum nondeterminism in the context of query and communication complexities started with de Wolf [dW00]. In particular, de Wolf [dW00, dW03] introduced the notion of *nondeterministic rank* of a matrix, which was proved to completely characterize strong quantum nondeterministic communication. In the same piece of work it was proved that strong quantum nondeterministic protocols are exponentially stronger than classical nondeterministic protocols. Similarly, Le Gall [LG06] studied weak quantum nondeterministic communication with classical proofs and showed a quadratic separation for a total function.

Weak nondeterminism seems a more suitable definition mainly due to the requirement of the existence of a proof, a concept that plays fundamental roles in

[1] `http://qwiki.stanford.edu/index.php/Complexity_Zoo`

complexity theory. In contrast, strong nondeterminism lends itself to a natural mathematical description in terms of matrix rank. Moreover, strong nondeterminism is a more powerful model capable of simulating weak nondeterminism with classical and quantum proofs. However, if weak nondeterminism is strictly a less powerful model or not is still an open problem.

The previous results by de Wolf [dW03] and Le Gall [LG06] were on the context of 2-party communication complexity, i.e., there are two players with two inputs $x, y \in \{0,1\}^n$ each and they want to compute a function $f(x,y)$. Let $rank(f)$ be the rank of the communication matrix M_f where $M_f[x,y] = f(x,y)$. A known result by [BdW01] is $\lceil \frac{1}{2} \log rank(f) \rceil \leq Q(f) \leq D(f)$, where $D(f)$ is the deterministic communication complexity of f and $Q(f)$ the quantum exact communication complexity[2]. It is conjectured that $D(f) = O(\log^c rank)$ for some arbitrary constant c. This is the *log-rank conjecture* in communication complexity, one of the biggest open problems in the field. If it holds, it will imply that $Q(f)$ and $D(f)$ are polynomially related. This is in stark contrast to the characterization given by de Wolf [dW03] in terms of the nondeterministic matrix-rank, which is defined as the minimal rank of a matrix (over the complex field) whose (x,y)-entry is non-zero if and only if $f(x,y) = 1$.

3.2 Overview of the Chapter

This work continues the study of strong quantum nondeterminism in the context of multiparty protocols. Let $k \geq 2$ be the number of players evaluating a function $f(x_1, \ldots, x_k)$ where each $x_i \in \{0,1\}^n$. The players take turns predefined at the beginning of the protocol. Each time a player sends a bit (or qubit if it is a quantum protocol), he sends it to the player who follows next. The computation of the protocol ends when the last player computes f. The communication complexity of the protocol is defined as the minimum number of bits that need to be transmitted by the players in order to compute $f(x_1, \ldots, x_k)$. There are two common ways of communication: The Number-On-Forehead model (NOF) where player i knows all inputs except x_i; and, Number-In-Hand model (NIH) where player i knows only x_i. Also, any protocol naturally defines a *communication*

[2]All logarithms in this thesis are base 2.

tensor T_f where $T_f[x_1, \ldots, x_k] = f(x_1, \ldots, x_k)$.

Tensors are natural generalizations of matrices. They are defined as multi-dimensional arrays while matrices are 2-dimensional arrays. In the same way, the concept of matrix rank extends to *tensor rank*. However, the nice properties of matrix rank do not hold anymore for tensors; for instance, unlike matrix rank for which there exist polynomial-time algorithms, computing tensor rank is **NP**-hard [Ha90]. See the survey paper by Kolda and Bader [KB09] for more differences.

This work extends the concept of nondeterministic matrices to *nondeterministic tensors*. The *nondeterministic tensor rank*, denoted $nrank(f)$, is the minimal rank of a tensor (over the complex field) whose $(x_1, \ldots, x_k)$-entry is non-zero if and only if $f(x_1, \ldots, x_k) = 1$.

Let SQ_k^{NOF} and SQ_k^{NIH} denote the k-party strong quantum nondeterministic communication complexity without prior shared entanglement for the NOF and NIH models respectively.

Theorem 3.2.1. *Let* $f : (\{0,1\}^n)^k \to \{0,1\}$*, then* $SQ_k^{NOF}(f) \leq \lceil \log nrank(f) \rceil + 1$*, and* $SQ_k^{NIH}(f) \geq \lceil \log nrank(f) \rceil + 1$.

This theorem generalizes previous results by de Wolf [dW03]. Also, since SQ_k^{NIH} is a lower bound for exact NIH quantum communication[3], denoted Q_k^{NIH}, we obtain the following corollary:

Corollary 3.2.2. $\lceil \log nrank(f) \rceil + 1 \leq Q_k^{NIH}(f)$.

The proof of Theorem 3.2.1 is given in Section 3.4. Even though it is a generalization of the techniques of [dW03], it requires technical insight. The proof does not generalize in an straightforward manner and it does not yield the same characterization as in the 2-player case. For example, SQ_k^{NOF} cannot be lower-bounded in general by the tensor rank. To see this consider the k-party equality function EQ given by $EQ_k(x_1, \ldots, x_k) = 1$ if and only if $x_1 = \cdots = x_k$. A nondeterministic tensor for EQ_k is superdiagonal[4] with non-zero entries in the main diagonal, and 0 anywhere else. Thus, it has 2^n rank and implies by Theorem 3.2.1 that $SQ_k^{NOF}(EQ_k) \leq n + 1$ and $SQ_k^{NIH}(EQ_k) \geq n + 1$. In particular, the

[3] An exact quantum protocol accepts a correct input and rejects an incorrect input with probability 1.

[4] An order-k tensor is *superdiagonal* when $T[x_1, \ldots, x_k] \neq 0$ if and only if $x_1 = \cdots x_k$.

communication complexity of EQ_k is upper-bounded by $\mathcal{O}(n)$ in the NOF model. However, it is easy to show that in the NOF model there exists a classical protocol for EQ_k with a cost of 2 bits[5]. Hence, the characterization for the 2-player case does not extends to the multiplayer case. In contrast, the lower bound on $SQ_k^{NIH}(EQ_k)$ that follows from Theorem 3.2.1 is not that loose; using the trivial protocol, where all players send their inputs, we have $SQ_k^{NIH}(EQ_k) = \mathcal{O}(kn)$. Thus, Theorem 3.2.1 yields a tight bound for EQ_k whenever $k = O(1)$. However, whether the same phenomenon extends to all functions in the NIH model is unknown. See below in this section for some consequences on constructing tensors with high rank.

A more interesting function is the generalized inner product $GIP_k(x_1, \ldots, x_k) = (\sum_{i=1}^{k} \bigwedge_{j=1}^{n} x_{ij}) \mod 2$. Section 3.5 shows that $nrank(GIP_k) \geq (k-1)2^{n-1} + 1$. Thus, the following result follows.

Proposition 3.2.3. $SQ_k^{NIH}(GIP_k) \geq n + \lceil \log(k-1) \rceil$.

In NIH, using the trivial protocol, we obtain (with Corollary 3.2.2) a bound in quantum exact communication of $n + \lceil \log(k-1) \rceil - 1 \leq Q_k^{NIH}(GIP_k) \leq (k-1)n + 1$. Improving the lower bound will require new techniques for explicit construction of linear-rank tensors with important consequences to circuit lower bounds; see for example Raz [Raz10a] and the paper by Alexeev, Forbes and Tsimerman [AFT11] for state-of-the-art tensor constructions. In general, it is open whether $SQ_k^{NIH}(f)$ can be upper-bounded in terms of $\log nrank$. This yields a new *log-rank conjecture* for strong quantum nondeterministic communication complexity.

Although the bounds given by Theorem 3.2.1 could be loose for some functions, they are good enough for other applications. For instance, Section 3.6 shows a separation between the NOF models of strong quantum nondeterminism and bounded-error quantum communication. This is proved by applying Theorem 3.2.1 to a total function previously studied by de Wolf [dW03]. This result

[5]In the *blackboard model* (explained in Section 3.3) for $k \geq 3$ let the first player check if $x_2, \ldots, x_k$ are equal. If they are, he sends a 1 bit to the second player who will check if $x_1, x_3, \ldots, x_k$ are equal. If his strings are equal and he received a 1 bit from the first player, he sends a 1 bit to all players indicating that all strings are equal. In the *message-passing* model the same protocol has a cost of $\mathcal{O}(k)$ bits.

could be considered as the quantum analog of a separation previously proved in [DPV09, CA08, GS10] between classical nondetermistic and randomized NOF communication.

3.3 Preliminaries

This section presents a small review of tensors and quantum communication.

3.3.1 Tensors

A *tensor* is a multi-dimensional array defined over some field. An order-d tensor is an element of the tensor product of d vector spaces.

Definition 3.3.1 (Simple Tensor)**.** Let $|v_i\rangle \in V^{n_i}$ be an n_i-dimensional vector for $1 \leq i \leq d$ on some vector space V^{n_i}. The j_i-th component of $|v_i\rangle$ is denoted by $v_i(j_i)$ for $1 \leq j_i \leq n_i$. The tensor product of $\{|v_i\rangle\}$ is the tensor $T \in V^{n_1} \otimes \cdots \otimes V^{n_d}$ whose $(j_1, \ldots, j_d)$-entry is $v_1(j_1) \cdots v_d(j_d)$, i.e., $T[j_1, \ldots, j_d] = v_1(j_1) \cdots v_d(j_d)$. Then $T = |v_1\rangle \otimes \cdots \otimes |v_d\rangle$ and we say T is a rank-1 or simple order-d tensor. We also say that a tensor is of high order if $d \geq 3$.

From now on, we will refer to high-order tensors simply as tensors, and low-order tensor will be matrices, vectors, and scalars as usual.

It is important to note that the set of simple tensors spans the space $V^{n_1} \otimes \cdots \otimes V^{n_d}$, and hence, there exist tensors that are not simple. This leads to the definition of rank.

Definition 3.3.2 (Tensor Rank)**.** The rank of a tensor is the minimum r such that $T = \sum_{i=1}^{r} A_i$ for simple tensors A_i.

This agrees with the definition of matrix rank. The complexity of computing tensor rank was studied by Håstad [Ha90] who showed that it is **NP**-complete for any finite field, and **NP**-hard for the rational numbers.

The process of arranging the elements of an order-k tensor into a matrix is known as *matrization*. Since there are many ways of embedding a tensor into a matrix, in general the permutation of columns is not important, as long as the corresponding operations remain consistent; see Kolda and Bader[KB09].

3.3.2 Strong Quantum Nondeterministic Communication

In a multiparty communication protocol there are $k \geq 2$ players trying to compute a function f. Let $f : (\{0,1\}^n)^k \to \{0,1\}$ be a function on k strings $x = (x_1, \ldots, x_k)$. There are two common ways of communication between the players: the Number-In-Hand (NIH) and the Number-On-Forehead (NOF) models. In NIH, player i only knows x_i, and in NOF, player i knows all inputs except x_i.

Definition 3.3.3 (Classical Nondeterministic Protocol)**.** Let k be the number of players. In order to communicate, the players take turns in an order predefined at the beginning of the protocol. Each player sends exactly one bit to the player that follows next. The computation of the protocol ends when the last player computes f. If $f(x) = 1$ then the protocol accepts x with positive probability; if $f(x) = 0$ the protocol rejects x with probability 1. The cost of the protocol is the total number of bits communicated.

Hence, the *classical nondeterministic multiparty communication complexity*, denoted $N_k(f)$, is defined as the minimum number of bits required to compute $f(x)$. If the model is NIH or NOF, we add a superscript $N_k^{NIH}(f)$ or $N_k^{NOF}(f)$ respectively. Note that the definition of the multiparty protocols in this thesis (classical and quantum) are by *message-passing*, i.e., a player sends a bit only to the player that follows next. This is in contrast to the more common *blackboard model*. In this latter model, when a player sends a bit, he does so by broadcasting it and reaching all players immediately. Clearly, any lower bound on the blackboard model is a lower bound for the message-passing model in this paper.

To model NOF and NIH in the quantum setting, we follow the work of Lee, Schechtman, and Shraibman [LSS09], originally defined by Kerenidis [Ker09].

Definition 3.3.4 (Quantum Multiparty Protocol)**.** Let k be the number of players in the protocol. Define the Hilbert space by $\mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_k \otimes \mathcal{C}$, where each $\mathcal{H}_i$ is the Hilbert space of player i and $\mathcal{C}$ is the one-qubit channel. To communicate the players take turns predefined at the beginning of the protocol. On the turn of player i:

1. in NIH, an arbitrary unitary that only depends on x_i is applied on $\mathcal{H}_i \otimes \mathcal{C}$ and acts as the identity anywhere else;

2. in NOF, an arbitrary unitary that depends on all inputs except x_i is applied on $\mathcal{H}_i \otimes \mathcal{C}$ and acts as the identity anywhere else.

The cost of the protocol is the number of rounds.

The initial state is a pure state $|0\rangle \otimes \cdots \otimes |0\rangle |0\rangle$ without any prior entanglement. If the final state of the protocol on input $x_1, \ldots, x_k$ is $|\Psi\rangle$, it outputs 1 with probability $p(x_1, \ldots, x_k) = \langle\Psi| \Pi |\Psi\rangle$, where Π is a projection onto the $|1\rangle$ state of the channel.

We say that T is a *nondeterministic communication tensor* if $T[x_1, \ldots, x_k] \neq 0$ if and only if $f(x_1, \ldots, x_k) = 1$. Thus, T can be obtained by replacing each 1-entry in the original communication tensor by a non-zero complex number. We also define the *nondeterministic rank* of f, denoted $nrank(f)$, to be the minimum rank over the complex field among all nondeterministic tensors for f.

Definition 3.3.5 (Strong Quantum Nondeterministic Protocol)**.** A *k-party strong quantum nondeterministic communication protocol* outputs 1 with positive probability if and only if $f(x) = 1$.

The k-party quantum nondeterministic communication complexity, denoted $SQ_k(f)$, is the cost of an optimum (i.e., minimal cost) k-party quantum nondeterministic communication protocol. If the model is NIH or NOF, we add a superscript $SQ_k^{NIH}(f)$ or $SQ_k^{NOF}(f)$ respectively. From the definition it follows that SQ_k is a lower bound for the exact quantum communication complexity Q_k for both NOF and NIH.

The following lemma, given in Lee, Schechtman, and Shraibman [LSS09], generalizes a previous observation made by Yao [Yao93] and Kremer [Kre95] on 2-party protocols.

Lemma 3.3.1. *After ℓ qubits of communication on input $(x_1, \ldots, x_k)$, the state of a quantum protocol without prior shared entanglement can be written as*

$$\sum_{m \in \{0,1\}^\ell} \left|A_m^1(x^1)\right\rangle \left|A_m^2(x^2)\right\rangle \cdots \left|A_m^k(x^k)\right\rangle |m_\ell\rangle ,$$

where m_ℓ is the ℓ-th bit in m, and each vector $|A_m^t(x^t)\rangle$ corresponds to the t-th player which depends on m and the input x^t. If the protocol is NOF then $x^t = (x_1, \ldots, x_{t-1}, x_{t+1}, \ldots, x_k)$; if it is NIH then $x^t = (x_t)$.

3.4 Proof of Theorem 3.2.1

3.4.1 Lower Bound

The arguments in this section are generalizations of a previous result by [dW03] from 2-party to k-party communication for $k \geq 3$. First we need the following technical lemma (see below for a proof).

Lemma 3.4.1. *If there exist k families of vectors such that $\{|A_1^i(x_i)\rangle, \ldots, |A_r^i(x_i)\rangle\} \subseteq \mathbb{C}^d$ for all i with $1 \leq i \leq k$ and $x_i \in \{0,1\}^n$ given that*

$$\sum_{i=1}^{r} \left|A_i^1(x_1)\right\rangle \otimes \cdots \otimes \left|A_i^k(x_k)\right\rangle = 0 \text{ iff } f(x_1, \ldots, x_k) = 0,$$

then $nrank(f) \leq r$.

Now we proceed to prove the lower bound as stated in Theorem 3.2.1.

Lemma 3.4.2. $SQ_k^{NIH}(f) \geq \lceil \log nrank(f) \rceil + 1$

Proof. Consider a NIH ℓ-qubit protocol for f. By Lemma 3.3.1 its final state is

$$|\psi\rangle = \sum_{m \in \{0,1\}^\ell} \left|A_m^1(x_1)\right\rangle \cdots \left|A_m^k(x_k)\right\rangle |m_\ell\rangle \,. \tag{3.1}$$

Assume all vectors have the same dimension d. Let $S = \{m \in \{0,1\}^\ell : m_\ell = 1\}$, and consider only the part of the state that is projected onto the 1-state of the channel,

$$|\phi(x_1, \ldots, x_k)\rangle = \sum_{m \in S} \left|A_m^1(x_1)\right\rangle \cdots \left|A_m^k(x_k)\right\rangle |1\rangle \,. \tag{3.2}$$

The vector $|\phi(x_1, \ldots, x_k)\rangle$ is 0 if and only if $f(x_1, \ldots, x_k) = 0$. Thus, by Lemma 3.4.1, we have that $nrank(f) \leq |S| = 2^{\ell-1}$, which implies the lower bound. □

Proof of Lemma 3.4.1

Let $k \geq 3$. We divide the proof in two cases, when k is odd and even.

Even k: There are k size-r families of d-dimensional vectors. We will construct two new families of vectors denoted $\mathscr{D}$ and $\mathscr{F}$. First, divide the k families in two groups of size $k/2$. Then, tensor each family in one group together in the following way: for each family $\{|A_1^i(x_i)\rangle, \ldots, |A_r^i(x_i)\rangle\}$ for $1 \leq i \leq k/2$ construct a new family

$$\begin{aligned}\mathscr{D} &= \left\{ \bigotimes_{j=1}^{k/2} \left|A_1^j(x_j)\right\rangle, \ldots, \bigotimes_{j=1}^{k/2} \left|A_r^j(x_j)\right\rangle \right\} \\ &= \left\{ \left|A_1(y)\right\rangle, \ldots, \left|A_r(y)\right\rangle \right\},\end{aligned}$$

where $y = (x_1, \ldots, x_{k/2})$. Do the same to construct $\mathscr{F}$ for $k/2 + 1 \leq i \leq k$ obtaining

$$\begin{aligned}\mathscr{F} &= \left\{ \bigotimes_{j=k/2+1}^{k} \left|A_1^j(x_j)\right\rangle, \ldots, \bigotimes_{j=k/2+1}^{k} \left|A_r^j(x_j)\right\rangle \right\} \\ &= \left\{ \left|B_1(z)\right\rangle, \ldots, \left|B_r(z)\right\rangle \right\},\end{aligned}$$

where $z = (x_{k/2+1}, \ldots, x_k)$. Thus, $\mathscr{D}$ and $\mathscr{F}$ will become two size-r family of vectors, each vector with dimension $dk/2$. Then apply the theorem for $k = 2$ from [dW03] on these two families and the lemma follows.

Odd k: Here we can use the same approach by constructing again two new families $\mathscr{D}$ and $\mathscr{F}$ by dividing the families in two groups of size $\lfloor k/2 \rfloor$ and $\lceil k/2 \rceil$. However, although both families will have the same number of elements r, the dimension of the vectors will be different. In fact, the dimension of the vectors in one family will be $d' = d\lfloor k/2 \rfloor$ and in the other $d' + 1$. So, in order to prove the theorem we will consider having two families $\{|A_1(y)\rangle, \ldots, |A_r(y)\rangle\} \subseteq \mathbb{C}^{d'}$ and $\{|B_1(z)\rangle, \ldots, |B_r(z)\rangle\} \subseteq \mathbb{C}^{d'+1}$, both with cardinality r.

Denote the entry of each vector $|A_i(y)\rangle, |B_i(z)\rangle$ by $A_i(y)_u$ and $B_i(z)_v$ respectively for all $(u, v) \in [d'] \times [d'+1]$. Note that, if $f(y, z) = 0$ then $\sum_{i=1}^r A_i(y)_u B_i(z)_v = 0$ for all (u, v); if $f(y, z) = 1$ then $\sum_{i=1}^r A_i(y)_u B_i(z)_v \neq 0$ for some (u, v). This

holds because each vector $|A_i(y)\rangle$ and $|B_i(z)\rangle$ are the set of vectors $|A_i^t(x^t)\rangle$ tensored together and separated in two families of size $\lfloor k/2 \rfloor$ and $\lceil k/2 \rceil$ respectively.

The following lemma was implicitly proved by de Wolf [dW03] for families of vectors with the same dimension. However, we show that the same arguments hold even if the families have different dimensionality (see below for a proof).

Lemma 3.4.3. *Let I be an arbitrary set of real numbers of size 2^{2n+1}. Let $\alpha_1, \ldots, \alpha_{d'}$ and $\beta_1, \ldots, \beta_{d'+1}$ be numbers from I, and define the quantities*

$$a_i(y) = \sum_{u=1}^{d'} \alpha_u A_i(y)_u \quad \textit{and} \quad b_i(z) = \sum_{v=1}^{d'+1} \beta_v B_i(z)_v.$$

Also let

$$v(y,z) = \sum_{i=1}^{r} a_i(y) b_i(z) = \sum_{u=1}^{d'} \sum_{v=1}^{d'+1} \alpha_u \beta_v \left(\sum_{i=1}^{r} A_i(y)_u B_i(z)_v \right).$$

There exists $\alpha_1, \ldots, \alpha_{d'}, \beta_1, \ldots, \beta_{d'+1} \in I$ such that for every $(y,z) \in f^{-1}(1)$ we have $v(y,z) \neq 0$.

Therefore, by the lemma above we have that $v(y,z) = 0$ if and only if $f(y,z) = 0$. Now let $|a_i\rangle$ and $|b_i\rangle$ be 2^n-dimensional vectors indexed by elements from $\{0,1\}^n$, and let $M = \sum_{i=1}^{r} |a_i\rangle \langle b_i|$. Thus M is a nondeterministic order-k tensor of rank r.

Proof of Lemma 3.4.3

If $f(y,z) = 0$ then $v(y,z) = 0$ for all α_u, β_v. If $f(y,z) \neq 0$ there exists (u', v') such that $v(y,z) \neq 0$. Here we use the same arguments given by [dW03], i.e., we show that $v(y,z) = 0$ happens with small probability. In fact, having families of vectors with different dimensions does not affect the argument. Consider the situation where all α_u and β_v were chosen except $\alpha_{u'}$ and $\beta_{v'}$. Write $v(y,z)$ in terms of these two coefficients

$$v(y,z) = c_0 \alpha_{u'} \beta_{v'} + c_1 \alpha_{u'} + c_2 \beta_{v'} + c_3,$$

where $c_0 = \sum_{i=1}^{r} A_i(y)_{u'} B_i(z)_{v'} \neq 0$. If we fix $\alpha_{u'}$ then, $v(y,z)$ is a linear equation with at most one zero for each $\alpha_{u'}$. Therefore, we have at most $2^{2n+1} + 2^{2n+1} - 1 =$

$2^{2n+2} - 1$ ways of choosing $\alpha_{u'}$ and $\beta_{v'}$ such that $v(y,z) = 0$. Thus

$$Pr[v(y,z) = 0] \leq \frac{2^{2n+1}}{(2^{2n+1})^2} < \frac{2^{2n+2}}{(2^{2n+1})^2} = 2^{-2n}.$$

By the union bound

$$\begin{aligned} &Pr[\exists (y,z) \in f^{-1}(1) \text{ s.t. } v(y,z) = 0] \\ &\qquad \leq \sum_{(y,z)\in f^{-1}(1)} Pr[v(y,z) = 0] < 2^{2n} \cdot 2^{-2n} = 1. \end{aligned}$$

The following is a probabilistic method argument. Since the above probability is strictly less than 1, there exists sets $\{a_1(y), \ldots, a_r(y)\}$ and $\{b_1(z), \ldots, b_r(z)\}$ such that for every $(y,z) \in f^{-1}(1)$ we have $v(y,z) \neq 0$.

3.4.2 Upper Bound

The proof of the upper bound follows by fixing a proper matrization (separating the cases of odd and even k) of the communication tensor, and then applying the 2-party protocol by de Wolf [dW03].

Lemma 3.4.4. $SQ_k^{NOF}(f) \leq \lceil \log nrank(f) \rceil + 1$.

Proof. Let T be a nondeterministic tensor for f with $nrank(f) = r$. We divide the proof in two cases.

Even k: Fix two players, say P_1 (Alice) and P_k (Bob). Also fix some matrization of T, i.e., let M be such matrization and consider it as an operator $M : \mathcal{H}_{k/2+1} \otimes \cdots \otimes \mathcal{H}_k \to \mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_{k/2}$. Thus M is a $2^{kn/2} \times 2^{kn/2}$-matrix that maps elements from the $\mathcal{H}_{k/2+1} \otimes \cdots \otimes \mathcal{H}_k$ subspace to the $\mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_{k/2}$ subspace. Let also $M = U\Sigma V$ be the singular value decomposition of M such that U, V are $2^{kn/2} \times 2^{kn/2}$ unitary matrices, and Σ is a $2^{kn/2} \times 2^{kn/2}$ diagonal matrix containing the singular values of M in the diagonal. The number of singular values is at most $rank(M) \leq r$.

Bob computes the state $\left|\phi_{1\cdots k/2}\right\rangle = c_{1\cdots k/2}\Sigma V \left|x_1, \ldots, x_{k/2}\right\rangle$ where $c_{1\cdots k/2}$ is some normalizing constant that depends on $x_1, \ldots, x_{k/2}$. Since only the first entries of Σ are non-zero, the vector $\left|\phi_{1\cdots k/2}\right\rangle$ has at most r non-zero entries,

so the state can be compressed using $\log r$ qubits[6]. Bob sends these qubits to Alice. Alice then computes $U\left|\phi_{1\cdots k/2}\right\rangle$ and measures that state. If Alice observes $x_{k/2+1},\ldots,x_k$ then she puts a 1 on the qubit channel, and otherwise she puts a 0. The probability of Alice putting a 1 on the channel is

$$\begin{aligned}
&\left|\left\langle x_{k/2+1},\ldots,x_k\right| U\left|\phi_{1\cdots k/2}\right\rangle\right|^2 \\
&\quad = |c_{1\ldots,k/2}|^2\left|\left\langle x_{k/2+1},\ldots,x_k\right| U\Sigma V\left|x_1,\ldots,x_{k/2}\right\rangle\right|^2 \\
&\quad = |c_{1\ldots,k/2}|^2\left|\left\langle x_{k/2+1},\ldots,x_k\right| M\left|x_1,\ldots,x_{k/2}\right\rangle\right|^2 \\
&\quad = |c_{1\ldots,k/2}|^2\left|M[x_1,\ldots,x_k]\right|^2 \\
&\quad = |c_{1\ldots,k/2}|^2\left|T[x_1,\ldots,x_k]\right|^2.
\end{aligned}$$

Since $T[x_1,\ldots,x_k]$ is non-zero if and only if $f(x_1,\ldots,x_k)=1$, this probability will be positive if and only if $f(x_1,\ldots,x_k)=1$. Thus, this is a nondeterministic protocol with total cost $\log r+1$.

Odd k: To use the protocol given in the even case, we add an extra degree of freedom to T.

Lemma 3.4.5. *If T is an order-k tensor with rank r then there exists a tensor T' of order $k+1$ with rank r where $T[x_1,\ldots,x_k]=T'[x_1,\ldots,x_kx_{k+1}]$ for all x_{k+1}.*

See below for a proof. By the above lemma above we have that $T'[x_1,\ldots,x_kx_{k+1}]=0$ if and only if $f(x_1,\ldots,x_k)=0$ for any given x_{k+1}.

Before the protocol starts, each player knows T' (which has even order) and its matrization M'. We fix two players, P_1 (Alice) and P_k (Bob), and they can now use the protocol for even k. □

Proof of Lemma 3.4.5

Let $T=\sum_{i=1}^r\left|v_1^i\right\rangle\cdots\left|v_k^i\right\rangle$ for some family of d-dimensional vectors. Define the tensor $T'=\sum_{i=1}^r\left|v_1^i\right\rangle\cdots\left|v_k^i\right\rangle\left|v_{k+1}^i\right\rangle$ where each $\left|v_{k+1}^i\right\rangle$ is the all-1 vector. Thus, component-wise we have that

$$T[x_1,\ldots,x_k]=\sum_{i=1}^r v_1^i(x_1)\cdots v_k^i(x_k),$$

[6] A n dimensional vector can be encoded as a quantum state with $\log n$ qubits by observing that a k-qubit state is a 2^k-dimensional vector. This fact was used by Raz [Raz99] to show an exponential separation between classical and quantum 2-party communication.

and

$$T'[x_1, \ldots, x_k x_{k+1}] = \sum_{i=1}^{r} v_1^i(x_1) \cdots v_k^i(x_k) v_{k+1}^i(x_{k+1}),$$

where $v_{k+1}^i(x_{k+1}) = 1$ for all i and for all inputs x_{k+1}. Then $T'[x_1, \ldots, x_k x_{k+1}] = \sum_{i=1}^{r} v_1^i(x_1) \cdots v_k^i(x_k)$ and $T'[x_1, \ldots, x_k x_{k+1}] = T[x_1, \ldots, x_k]$ for any x_{k+1}.

3.5 Rank Lower Bound for the Generalized Inner Product Function

This section shows a lower bound on the nondeterministic rank of the Generalized Inner Product (GIP) function.

Lemma 3.5.1. *$nrank(GIP_k) \geq (k-1)2^{n-1} + 1$.*

Proof. First, we start by generalizing the concept of rows and columns for tensors. Define a *fiber* to be a vector obtained by fixing every index except one. In general, a mode-i fiber is a vector obtained by fixing all except the i-th index. Thus, a matrix column is a mode-1 fiber, and a row is a mode-2 fiber. For order-3 tensors, we have columns, rows and tubes, and so on for higher order tensors. In the same way we define a *slice* to be a two-dimensional section of T obtained by fixing all but two indices.

Here we will consider a particular form of matrization. Let $T \in \mathbb{C}^{n_1 \times \cdots \times n_k}$ be an order-k tensor, with $n_i = 2^n$ for every i. The *i-mode unfolding* of T, denoted $T_{(i)}$, is the matrix obtained by arranging the i-mode fibers as columns. The permutations of the columns of $T_{(i)}$ is not important, as long as the corresponding operations remain consistent; see Kolda and Bader [KB09]. Define the *i-rank* of T as $rank_i(T) = rank(T_{(i)})$. It is trivial that $rank_i(T) \leq rank(T)$ for every i; see Lathauwer, de Moore, and Vandewalle [dLdMV00].

Now we proceed with the proof. Let T be the order-k nondeterministic communication tensor for GIP_k. Let M_{IP_n} be the boolean communication matrix for GIP_2, i.e., the 2-party inner product function on n bits. It is well known that $rank(M_{IP_n}) = 2^n - 1$; see Example 1.29 in Kushilevitz and Nisan[KN97]. The same holds even if M_{IP_n} is defined over $\mathbb{C}$.

Let $\mathbf{1}$ denote the string of length n with only 1s in it, and let T' be the $(x'_3, \ldots, x'_k)$-slice of T where $x'_i = \mathbf{1}$ for $i = 3, \ldots, k$. In this way $T'[x_1, x_2] \neq 0$ whenever $\langle x_1|\ x_2\rangle = 1$ and hence $rank(T') = rank(M_{IP_n}) = 2^n - 1$.

Let $x^{(i)}$ denote the string x with the i-th bit flipped. For $i = 3, \ldots, k$ consider the $(x'_3, \ldots, x'^{(i)}_k)$-slice of T denoted T'_i where $x'^{(i)}_k$ is the string $\mathbf{1}$ with the i^{th} bit flipped to 0. Then,

$$T'_i[x_1, x_2] \neq 0 \text{ whenever } \langle x_1|\ x_2\rangle - x_{1i}x_{2i} = 1. \tag{3.3}$$

Note that the non-zero entries of T'_i for any i agrees with the non-zero entries of $M_{IP_{n-1}}$, where $M_{IP_{n-1}}$ is obtained by deleting the i-th bits of x_1 and x_2 in M_{IP_n} for all x_1 and x_2. Thus, $rank(T'_i) = 2^{n-1} - 1$ for all $i = 3, \ldots, k$.

The 1-mode unfolding of T is obtained by fixing every index except x_1. Thus

$$T_{(1)} = \begin{bmatrix} T' & T'_3 & \cdots & T'_k & \cdots \end{bmatrix},$$

with $2^{(k-1)n}$ columns, and the right part of $T_{(1)}$ (after T'_k) is filled with the remaining slices of T that are different to T' and each T'_i. We known that T' and each T'_i have $(2^n - 1)$ and $2^{n-1} - 1$ linearly independent columns respectively. Also, each of these columns are pair-wise linearly independent. To see this, just take take any two slices T'_i and T'_j for any $i \neq j$, fix one column in each and compute the inner product according to Equation 3.3. Thus, $rank(T) \geq rank_1(T) \geq 2^n - 1 + (k-2)(2^{n-1} - 1) = (k-1)2^{n-1} + 1$. □

3.6 Some Separations for Complexity Classes

In this section we take a complexity-theoretic view of quantum multiparty communication complexity. Remember that "efficient communication" means that a protocol computes a function with $polylog(n)$ bits [BFS86].

Definition 3.6.1. We define the following communication complexity classes:

1. **BPP** is the class of boolean functions with a classical bounded-error protocol of cost $polylog(n)$;
2. **BQP** is the class of boolean functions with a quantum bounded-error protocol of cost $polylog(n)$;

3. **NQP** is the class of boolean functions with a quantum strong nondeterministic protocol of cost $polylog(n)$.

In the following two theorems are presented that give separations between the complexity classes defined above. First, for better understanding, we start by showing a weaker nevertheless easier to prove result, a separation between **NQP** and **BPP**. Then, that result is used to separate **NQP** from **BQP**. Although this latter result can be proved without the need of the former, starting with the separation from **BPP** seems easier to understand.

Theorem 3.6.1. *For NOF communication we have* $\mathbf{NQP} \not\subseteq \mathbf{BPP}$ *whenever the number of players* $k = o(\log\log n)$.

Proof. To prove this we exhibit a function $f : (\{0,1\}^n)^k \to \{0,1\}$ such that $SQ_k^{NOF}(f) = O(\log n)$ and $R_{\epsilon,k}^{NOF}(f) = \Omega(n^{1/(k+1)}/(k2^{2^k}))$, where $R_{\epsilon,k}^{NOF}$ denotes the k-party bounded-error NOF communication complexity with error probability upper-bounded by ϵ. This will give the separation whenever $k = o(\log\log n)$.

In particular, we analyze the following total function. Let $x_1, \ldots, x_k \in \{0,1\}^n$, then

$$w_n(x_1, \ldots, x_k) = \begin{cases} 1 & \text{if } |x_1 \wedge \cdots \wedge x_k| \neq 1 \\ 0 & \text{if } |x_1 \wedge \cdots \wedge x_k| = 1 \end{cases}, \tag{3.4}$$

where $\wedge$ denotes the bit-wise AND and $|x|$ is the Hamming weight of x. This function was previously studied by de Wolf [dW03] in the 2-player case.

Upper Bound: For each i let $x_i = x_{ij_1} \ldots x_{ij_n}$ and let T_j be an order-k tensor where $T_j[x_1, \ldots, x_k] = 1$ if $x_{1j} \wedge \cdots \wedge x_{kj} = 1$ and $T_j[x_1, \ldots, x_k] = 0$ otherwise. Note that for each j the tensor T_j has rank 1. Define the order-k tensor T by

$$T[x_1, \ldots, x_k] = \sum_{j=1}^{n} T_j[x_1, \ldots, x_k] - 1.$$

This tensor has rank n. Also T is a nondeterministic communication tensor for f since $T[x_1, \ldots, x_k] = 0$ if and only if $|x_1 \wedge \cdots \wedge x_k| = 1$. Hence, by Theorem 3.2.1 the upper bound follows.

Lower Bound: To prove the lower bound we will use, without loss of generality,

the sign version of Equation (3.4), i.e.,

$$f(x_1, \ldots, x_k) = \begin{cases} 1 & \text{if } |x_1 \wedge \cdots \wedge x_k| \neq 1 \\ -1 & \text{if } |x_1 \wedge \cdots \wedge x_k| = 1 \end{cases} . \tag{3.5}$$

We make use of a result by Lee and Shraibman [LS09a]. Let μ^α be the *approximate cylinder intersection norm* as defined in [LS09a] and let $\widetilde{deg}(f)$ be the *approximate degree* of a boolean function f [NS92] (see also Appendix C).

Lemma 3.6.2. *Let $f_n : \{0,1\}^n \to \{-1,1\}$ be a symmetric[7] function, and let $F_f : (\{0,1\}^n)^k \to \{-1,1\}$ be a function (not necessarily symmetric) defined by $F_f(x_1, \ldots, x_k) = f(x_1 \wedge \cdots \wedge x_k)$. Let $\alpha > 1/(1-2\epsilon)$ and set $c = 2e(k-1)2^{2^{k-1}}$, then*

$$R_{1/4,k}(F_{f_n}) = \Omega(\log \mu^\alpha(F_{f_n})) = \Omega\left(\frac{\widetilde{deg}(f_m)}{2^k}\right),$$

where $n = (c/\widetilde{deg}(f_m))^{k-1} m^k$.

Note that Lemma 3.6.2 is a generalization of [LS09a, Corollary 6.1] to symmetric functions. However, as pointed by the authors of [LS09a], this generalization is straightforward and can be easily proved by following the proof of [LS09a, Corollary 6.1], and it is therefore omitted from this work.

Define the following Hamming weight function:

$$\varphi(x) = \begin{cases} 1 & \text{if } |x| \neq 1 \\ -1 & \text{if } |x| = 1 \end{cases} .$$

This way we can write Equation 3.5 as $f(x_1, \ldots, x_k) = \varphi(x_1 \wedge \cdots \wedge x_k)$. Also note that φ is symmetric and we can apply Lemma 3.6.2. Together with the characterization given by Paturi [Pat92] of the approximate degree of symmetric functions we have that

$$\log \mu^\alpha(f) = \Omega\left(\frac{n^{1/(k+1)}}{k2^{2^k}}\right). \tag{3.6}$$

□

Theorem 3.6.3. *For NOF communication we have that* $\mathbf{NQP} \not\subseteq \mathbf{BQP}$ *whenever the number of players $k = o(\log \log n)$.*

[7] A function is called symmetric if it only depends on the number of 1s in the input.

Proof. To prove this we rely again in Equation (3.5) and the fact that $SQ_k^{NOF}(f) = O(\log n)$. Here we show that $Q_{\epsilon,k}(f) = \Omega(n^{1/(k+1)}/(k2^{2^k}) - k)$, where $Q_{\epsilon,k}$ denotes the bounded-error NOF communication complexity with error probability upper-bounded by ϵ.

Note that to prove Theorem 3.6.1 we derived a lower bound on μ^α (Equation 3.6). We can use the same lower bound to prove the separation for **BQP**. In order to do that we make use of the following two results by Lee, Schechtman, and Shraibman [LSS09]. Let γ^α be the *approximate quantum norm* as defined in [LSS09].

Lemma 3.6.4. *Let T be an order-k sign-tensor, then $Q_{\epsilon,k}(T) = \Omega(\log \gamma^\alpha(T))$.*

Lemma 3.6.5. *For every order-k tensor T, $\gamma(T) \leq \mu(T) \leq C^k\gamma(T)$, for some absolute constant C.*

Thus, by these two lemmas above and Equation 3.6 we have that

$$\log \gamma^\alpha(f) = \Omega\left(\frac{n^{1/(k+1)}}{k2^{2^k}} - k\right).$$

□

3.7 Concluding Remarks of the Chapter

In this chapter we studied strong quantum nondeterministic communication complexity in multiparty protocols. In particular, it was shown that i) strong quantum nondeterministic NOF communication complexity is upper-bounded by the logarithm of the rank of the nondeterministic communication tensor; ii) strong quantum nondeterministic NIH communication complexity is lower-bounded by the logarithm of the rank of the nondeterministic communication tensor. These results naturally generalizes previous work by de Wolf [dW03]. Moreover, the lower bound on NIH is also a lower bound for quantum exact NIH communication. This fact was used to show a $\Omega(n + \log k)$ lower bound for the generalized inner product function.

We also showed that $\mathbf{NQP} \not\subseteq \mathbf{BQP}$ when the number of players is $o(\log \log n)$. It remains as an open problem to prove the same separations with an increased number of players.

In order to prove strong lower bounds using tensor-rank in NIH, we need stronger construction techniques for tensors. The fact that computing tensor-rank is **NP**-complete suggests that this could be a very difficult task. Alternatives for finding lower bounds on tensor-rank include computing the norm of the communication tensor, or a hardness result for approximating tensor-rank.

Chapter 4

Concluding Remarks of the Thesis

4.1 Summary

This work presented fundamental studies on quantum computation in two very important models in computer science, quantum walks and communication complexity. Both models are intimately connected, in particular, lower bounds in communication implies lower bounds for decision trees, and upper bounds in decision tree depth (which can be found using quantum walks) implies upper bounds for communication.

In Chapter 2 we saw some examples of how quantum walks can be used to construct quantum query algorithms. Proving the correctness of a quantum algorithm (also randomized algorithms) requires computing the error probability for correct and incorrect inputs. Hence, in this thesis the problem of computing the probability distribution induced by quantum walks moving over an infinite line was tackled. To that end, a technique from complex analysis known as the steepest descent method was applied to compute a closed-form formula for the probability distribution. Previous work on the same problem only computed the probability distribution for specific kinds of quantum walks. In this work, we analyzed a quantum walk moved by general $SU(2)$ coin operator. Since operators from this group make use of arbitrary complex number as long as the determinant is one, it was necessary to make use of the steepest descent method.

In Chapter 3 we focused on bounds in communication complexity. The main contribution of this part was a generalization of a lower bound technique proposed by de Wolf [dW03] known as the nondeterministic rank. His technique only worked for 2-party quantum strong nondeterministic communication, and therefore, it was necessary to study the behavior of tensor-rank in multiparty quantum strong nondeterministic protocols. We saw that in the Number-In-Hand model, strong quantum nondeterministic complexity is lower-bounded by the logarithm of the nondeterministic tensor-rank of the communication tensor. In the Number-On-Forehead model, strong quantum nondeterministic communication complexity is upper-bounded by the logarithm of the nondeterministic tensor-rank of the communication tensor. Furthermore, this result was later applied to show the first nontrivial lower bound on the Generalized Inner Product function for quantum exact multiparty communication in the Number-In-Hand model. As a second application, we were able to prove a separation between quantum strong nondeterministic communication and bounded 2-sided error quantum communication, i.e., $\mathbf{NQP} \not\subseteq \mathbf{BQP}$ for the Number-On-Forehead communication whenever the number of players is $o(\log \log n)$.

In summary, this thesis presented techniques that deepen our knowledge of quantum walks and quantum multiparty communication. Both parts of this thesis unearthed nontrivial connections between unexplored parts of complex analysis and abstract algebra. Currently these models are thoroughly studied by several researchers due to the promising applications to quantum computation. Quantum walks have been recently proved capable of universal computation [CGW12] and communication technologies based on quantum physics are striving [MHS+12]. Therefore, a deep understanding of the theoretical foundations of these models is extremely important.

4.2 Open Problems

To conclude this thesis, a list of what could be (arguably) the most important open problems left is presented.

1. **Quantum Walks**
    (a) A generalization of the techniques of this thesis to other graphs, e.g,

hypercube, grids, Cayley graphs, etc.

(b) Relation between the moment of methods [GJS04] and the Fourier coefficients of quantum walks.

(c) A closed-form formula for the probability distribution of Quantum Markov Chains [Sze04].

(d) An exact computation of the induced probability distribution.

2. **Quantum Nondetermistic Communication**

(a) Upper bound on quantum strong nondeterministic communication in terms of the nondeterministic tensor-rank.

(b) Relation between the norm-bound and nondeterministic rank. Also its relation to information complexity.

(c) Hardness of approximating tensor-rank.

(d) New techniques for explicit construction of tensors with high rank.

(e) Lower bounds for other models of quantum nondeterministic communication, i.e., **QMA** and **QCMA** communication.

(f) Gap between quantum exact and quantum strong nondeterministic communication.

Appendix A

Quantum Computation

Building a working quantum computer is one of the grand challenges of the 21st century. A quantum computer exploits quantum phenomena in order to rapidly solve complex computational problems in nature. It is not believed to help in solving efficiently **NP**-complete problems [BBBV97], nevertheless, it does gives an speed-up in the solution of several computational problems.

In this chapter, a review of the basic mathematical concepts of quantum computation is given. Section A.1 argues about the main motivations for the study of this field. In Section A.2 the basic unit of quantum information is defined, the qubit. In sections A.2 and A.3 we explain the basic buildings blocks for quantum computation. Finally, in Section A.4 we present a very general algorithm for search known as amplitude amplification. For a complete introduction to quantum computation, we refer the interested reader to the books by Nielsen and Chuang [NC00] and Kaye, Laflamme and Mosca [KLM07].

A.1 Why Quantum Computing

Quantum computing is a field that mixes three different sciences. Two of the oldest fields of science, mathematics and physics, and a third and more recent: computer science. Today, computer science has become an interdisciplinary field with influences from economics, biology, physics, to name a few. Quantum computing is one of the hottest topics for its potential to solve more rapidly computational problems, and it also seems to hold the key to answer questions about

the universe and our existence [Aar04, Aar05]. There are techniques from computer science (and quantum computing) that were used to prove open problems in mathematical physics (cf. Aaronson [Aar05]).

In the following we will try to argue in favor of the main motivations behind its study.

A.1.1 Building Quantum Computers

The original idea on quantum computing was proposed by Feynman in 1982 [Fey82] mainly motivated by the necessity of simulating quantum physics using a computer. Later in 1985, Deutsch [Deu85] formalized the notion of quantum computation generalizing the Turing machine model. Besides being an interesting model by itself, the quantum Turing machine did not woke up any immediate interest. One clear example that showed that quantum computers could be more powerful than classical computers was the Deutsch-Jozsa algorithm [DJ92]. Given a boolean function f on n bits, the algorithm decides if f is constant (outputs 0 on all inputs or 1 on all inputs) or balanced (outputs 0 for half of the inputs and 1 for the other half). To solve this problem with a classical computer, $\Theta(2^n)$ evaluations of f are needed. The Deutsch-Jozsa algorithm only requires linear time. This was the first exponential separation between a classical computer and a quantum computer.

The problem solved by the Deutsch-Jozsa algorithm does not have any practical application and there was no enough motivation for the study of quantum computation. This was true until a breakthrough result: A polynomial-time algorithm for factoring large composite numbers [Sho94]. This result is simply known as Shor's Algorithm (by the name of its discoverer). Today, all the security of the world is based on the assumption that factoring is hard, and from that point on quantum computing started to bloom. Several fields inside quantum computing developed like Quantum Complexity Theory, Quantum Error Correcting Codes, Quantum Cryptography, etc. Also, people started to race for building the first quantum computer, however, this turned out to be more than a challenge. A recent survey on building quantum computers can be found in [LJL+10].

Besides all advantages that quantum computers could bring, the most important (arguably) reason for studying quantum computation is to test the theory

of quantum mechanics. If we fail in this attempt, it could give evidence (for the first time) that maybe there is something wrong in the theory.

A.1.2 Computational Hardness

After the discovery of Shor's algorithm, several researchers saw this as a computational hardness result. Another group of people not believing in the possibility of building a quantum computer took Shor's algorithm as evidence against.

For theoretical computer science, what Shor's algorithm brought was the necessity to rethink on the most famous universal computation model: The Turing Machine. The *Strong Church-Turing Thesis* reads: "Any computational model can be efficiently simulated by a Turing machine[1]". The fact that there exists a plausible model (Quantum Turing Machine) that cannot be simulated efficiently by a Turing machine hits hard on the theoretical foundations of the field. Shor's algorithm gives evidence of a computational model beyond the classical Turing machine, which everyone thought to be correct. From this point of view, quantum computing brings forward questions about the *physical limits* of computation [Aar05].

A.2 Quantum Bits and Registers

A.2.1 The Qubit

A *classical* bit holds two states: 1 or 0. To extend this notion to quantum bits, or just qubits, define it as a vector on a Hilbert space. We denote vectors using the Dirac notation, i.e., $|\psi\rangle$ is a vector in some vector space, and $\langle\psi|$ represents its dual in the dual vector space of functionals.

Definition A.2.1. Let $\mathcal{H} = span\{|0\rangle, |1\rangle\}$ be a Hilbert space equipped with a ℓ_2-norm. A qubit is a vector $|\psi\rangle \in \mathcal{H}$ over the complex field defined as

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle ,$$

where $|\alpha|^2 + |\beta|^2 = 1$.

[1]The Turing machine could be deterministic or probabilistic.

The set $\{|0\rangle, |1\rangle\}$ is known as the *computational basis* which is an orthonormal basis for $\mathcal{H}$. These vectors can be written in matrix notation as

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

which is a canonical basis for a 2 dimensional space.

A qubit could be considered as a 1-bit register holding the two states $|0\rangle$ and $|1\rangle$ at the same time. In general it is known as a *superposition state*. Contrarily to a classical bit, which could be queried obtaining a deterministic answer (either 0 or 1), when we query a qubit we get a probabilistic answer. This answer will be 0 with probability $|\alpha|^2$ or 1 with probability $|\beta|^2$.

A.2.2 Registers

In order to generalize a system to n qubits we need to introduce tensor vector spaces.

Definition A.2.2. Let $\mathcal{H} = span\{|x\rangle : x \in \{0,1\}^n\}$ be a Hilbert space with a ℓ_2-norm. An n qubit state is a vector $|\psi\rangle \in \mathcal{H}$ defined as

$$|\psi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle,$$

where $1 = \sum_x |\alpha_x|^2$, and $|x\rangle = |x_1\rangle \otimes \cdots \otimes |x_n\rangle$ with each $|x_i\rangle \in span\{|0\rangle, |1\rangle\}$.

Observe that the number of possible basis states for a n-qubit register scales exponentially faster. This seems to store an exponential amount of information. However, by the Holevo bound [NC00], in order to recover information faithfullly, n is the maximum number of bits one can store in an n-qubit register.

A.2.3 Operations

Algorithms for quantum computers are built as a sequence of quantum operations (or gates, in analogy to classical circuits) acting on qubit registers. These quantum operations are essentially unitary operations defined over some Hilbert space.

1-qubit Operations

Maybe the simplest operation is the NOT operator. Given a qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, it interchanges the amplitudes in the following way

$$NOT(\alpha|0\rangle + \beta|1\rangle) = \beta|0\rangle + \alpha|1\rangle.$$

This operation is easily defined as

$$\begin{aligned} NOT &\equiv |0\rangle\langle 1| + |1\rangle\langle 0| \\ &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \end{aligned}$$

Another important quantum gate is the Hadamard operation denoted with H. It is basically a rotation operation and agrees exactly with a Fourier transform on a 2-dimensional space. It acts in the following way

$$H|i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^i|1\rangle).$$

This operation is defined as

$$\begin{aligned} H &\equiv \frac{1}{\sqrt{2}} \sum_{i,j\in\{0,1\}} (-1)^{i\cdot j} |i\rangle\langle j| \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \end{aligned}$$

2-qubits Operations

A very popular generalization of the NOT operation is CNOT, which stands for controlled-NOT. This gate flips a target qubit if and only if another qubit is set to 1, and it is denoted by $CNOT$. Its actions are

$$\begin{aligned} CNOT : |00\rangle &\longrightarrow |00\rangle \\ |01\rangle &\longrightarrow |01\rangle \\ |10\rangle &\longrightarrow |11\rangle \\ |11\rangle &\longrightarrow |10\rangle. \end{aligned}$$

Here, the first qubit controls the fliping of the second qubit. It resembles exactly a conditional statement for quantum computation. Formally it is defined as

$$\begin{aligned} CNOT &\equiv \sum_{i,j\in\{0,1\}} |0\rangle\langle 0| \otimes |j\rangle\langle j| + |1\rangle\langle 1| \otimes |1-j\rangle\langle j| \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \end{aligned}$$

A second very important 2-quibts operation is SWAP. This operation interchanges the state of two different qubits

$$SWAP\,|i,j\rangle = |j,i\rangle\,.$$

The operation is formally defined as

$$\begin{aligned} SWAP &\equiv \sum_{i,j\in\{0,1\}} |i\rangle\langle j| \otimes |j\rangle\langle i| \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

n-qubits Operations

It is easy to see that for an n-qubit system, the number of rows and columns in the matrix representation of the operators grow exponentially faster. To define operations on n-qubits, the braket notation is a convenient tool.

A generalization of the Hadamard gate to n qubits is known as the Walsh-Hadamard transform denoted as W. It can be written as

$$W \equiv \frac{1}{\sqrt{2^n}} \sum_{x,y\in\{0,1\}^n} (-1)^{x\cdot y}\,|x\rangle\langle y|\,.$$

Further generalizations of other operations like CNOT, controlled-SWAP, etc is straightforward.

A.3 Measurements and Observables

Evolution in a closed quantum system is unitary. The previous section presented different types of unitary operators. However, to read-out the result of a quantum computation we need to be able to measure the state of the system. In this section we will see a basic introduction to the process of measurement of a quantum system.

First we start by defining an *observable*. In simple terms, this is the dynamic variable we want to measure, e.g., velocity, energy, spin, etc. Normally, in quantum computation we want to know if a qubit is in state $|0\rangle$ or $|1\rangle$. Formally, an observable is a Hermitian operator that acts on the Hilbert space of the system whose eigenvalues and eigenvectors correspond to the values and states of the dynamic variable.

To be able to measure an observable we need to make a *measurement*. Formally, a measurement is a set of linear operators $\{M_m\}$ that acts on the Hilbert space of the system being observed. The index m refers to the outcome of the measurement.

Say that the system is in state $|\psi\rangle$. When we measure it, the probability that m occurs is

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

and the state of the system is

$$\frac{M_m |\psi\rangle}{\sqrt{P(m)}}.$$

In the type of measurements we deal with in this thesis, called *projective measurements*, we want each m to be an eigenvalue of an observable. Given an obervable O, its spectral decomposition is

$$O = \sum_m m P_m,$$

where P_m is a projection onto the subspace with eigenvalue m. Thus, the probability of getting m is

$$P(m) = \langle\psi| P_m |\psi\rangle ,$$

and the state of the system is

$$\frac{P_m |\psi\rangle}{\sqrt{P(m)}}.$$

A.4 Quantum Search Algorithms

The most popular and also the first search algorithm was given by Grover [Gro96]. This is an algorithm for unstructured search, i.e., given n objects, with no information about how they are positioned in the search space, find a set of marked objects. Grover showed an upper bound of $\mathcal{O}(\sqrt{n})$ operations for a set of n objects. This is a quadratic speed-up with respect to classical algorithms (classically for a randomized algorithm $\Omega(n)$ steps are required). This bound is tight as showed by Bennett, Bernstein, Brassard, and Vazirani [BBBV97].

The field of quantum search algorithms is a very popular area. Several problems in the query model of computation were developed for graphs, matrices, groups, etc (cf. The Quantum Algorithms Zoo[2]). Also, there exist several lower bound methods, mainly based on adversary arguments [vS06] and polynomials [BBC+01].

There are basically two main techniques for making search algorithms: Amplitude Amplification and Quantum Walks. Here we briefly give an intuitive picture of how a general search algorithm works by amplitude amplification. Quantum walks can be seen as a generalization of this search procedure, where some structure is given to the search space.

Amplitude amplification [BHMT02] is an algorithm for unstructured search spaces. Its a generalization of Grover's algorithm and it has the same query complexity, i.e., $\mathcal{O}(\sqrt{n})$ queries for a set with n elements. The Hilbert space is decomposed in terms of a direct sum of good and bad subspaces $\mathcal{H} = \mathcal{H}_{good} \oplus \mathcal{H}_{bad}$. The good subspace $\mathcal{H}_{good}$ is spanned by the marked elements, and the bad subspace $\mathcal{H}_{bad}$ is the orthogonal complement. Denote a vector in $\mathcal{H}$ as $|\psi\rangle = \alpha |\psi_{good}\rangle + \beta |\psi_{bad}\rangle$, where $\mathcal{H}_{good} = span\{|\psi_{good}\rangle\}$ and $\mathcal{H}_{bad} = span\{|\psi_{bad}\rangle\}$. The amplification process is realized by the repeated application of the following operator,

$$Q = -AS_0A^{-1}S_x.$$

The operator S_x is the oracle, and it changes the sign of the amplitudes of good states, i.e., if $f(x) = 1$ then $|x\rangle$ is transformed to $-|x\rangle$. Operator S_0 changes only the sign of the zero state. Finally, A is any unitary operation in charge of

[2]http://www.its.caltech.edu/ sjordan/zoo.html

exploring the search space, e.g., A could be the Walsh-Hadamard operation as in Grover's algorithm.

Let $p = \langle \psi_{good} | \, \psi_{good} \rangle$ be the probability of measuring a vector from $\mathcal{H}_{good}$. In order to use Q to amplify p, we define the state of the algorithm as $|\psi\rangle = A\,|0\rangle$, and then apply t times operator Q, i.e.,

$$|\psi_t\rangle = Q^t \, |\psi\rangle \, .$$

If $t = O(\frac{1}{\sqrt{p}})$ with $p = m/n$ where m is the number of marked elements, we have that $t = O(\sqrt{n})$. For details refer to [BHMT02].

Appendix B

Decision Tree Complexity

Today it is still very difficult to answer many questions on the power and limitations of computational models like Turing machines, RAM, etc. Therefore, it is convenient to address these questions in more idealized models of computation. In this chapter we review the basics of the *decision tree model*.

B.1 Deterministic Decision Trees

Let $f : \{0,1\}^n \to \{0,1\}$ be a boolean function. A *decision tree* A for f on input is a binary tree where the nodes at level $i = 1, \ldots, n$ are labeled with x_i. Each node has two outgoing edges labeled 0 and 1. The computation on input $x = x_1 \cdots x_n$ starts at the root and proceeds down the tree by choosing one of the two children. If at the root we choose the edge labeled 1 we let $x_1 = 1$ and so on till all the variables have a value assigned. When the last variable x_n is assigned a value, we move to a leaf which will contain the value $f(x)$.

The assignment of values to nodes of the tree does not follow any particular order. At each level of the tree we could have given a value to any remaining unassigned variable. This way, we can view a computation on a decision tree as using a *black-box* or *oracle* at each level. This black-box has access to each bit of the input, and the algorithm needs to query the black-box in order to know the input. Hence, we can define the cost of a decision tree A on input x, denoted $cost(A, x)$, as the depth of the tree, or equivalently, as the number of queries made to the black-box.

Definition B.1.1 (Deterministic Decision Tree Complexity)**.** The decision tree complexity of a function $f : \{0,1\}^n \to \{0,1\}$ is defined as

$$\hat{D}(f) = \min_{A \in \mathcal{T}_f} \max_{x \in \{0,1\}^n} cost(A, x),$$

where $\mathcal{T}_f$ is the set of all decision trees computing f.

It is clear that $\hat{D}(f) \leq n$ for any f, since the full binary tree of depth n has 2^n leafs.

B.2 Randomized Decision Trees

In a randomized decision tree, every time we choose a child we do so randomly. A more convenient, but equivalent, way to define a randomized decision tree is as a probability distribution over deterministic decision trees.

We will put emphasis on decision trees that could err on inputs. Normally this type of algorithms are called Montecarlo (2-sided error or bounded-error) algorithms. Let $\mathcal{T}_{f,\epsilon}$ be the set of deterministic decision trees computing f that err on at most a fraction ϵ of the inputs.

Definition B.2.1 (Randomized Decision Tree Complexity)**.** Let $\mathcal{P}_f$ be a set of probability distributions over $\mathcal{T}_{f,\epsilon}$. The randomized decision tree complexity with error bound ϵ is defined as

$$\hat{R}_\epsilon(f) = \min_{P \in \mathcal{P}_f} \max_{x \in \{0,1\}^n} E_{A \in P}[cost(A, x)],$$

where $E_{A \in P}[cost(A, x)]$ denotes the expected value of the cost under the probability distribution P.

It is clear that $\hat{R}(f) \leq \hat{D}(f)$ for any f since a randomized decision tree is a generalization of a deterministic decision tree. Similarly, we can define the randomized decision tree complexity for Las Vegas (0-sided error) algorithms.

B.3 Lower Bounds for Classical Decision Trees

Here we give one (maybe the only one) technique for lower-bounding randomized decision tree complexity. For deterministic trees it is common to prove a lower

bound by *adversary arguments*. However, since any lower bound technique for $\hat{R}_\epsilon(f)$ is also good for $\hat{D}(f)$, we will concentrate on randomized trees. This will also be useful when studying quantum decision trees later on this chapter.

B.3.1 Yao's Minimax Principle

The technique which bares the name of Andrew Chi-Chih Yao appeared in [Yao77]. Yao showed that we can lower bound randomized decision trees by considering instead deterministic decision trees.

Let $\mathcal{I}$ be a finite set of inputs and let $\mathcal{A}_f$ be a set of deterministic algorithms for a boolean function f that fails to give a correct answer on some inputs. We denote by $cost(A, x)$ the cost incurred by algorithm $A \in \mathcal{A}_f$ on input $x \in \mathcal{I}$. Also let $\varphi(A, x) = 0$ if A gives the correct answer for x, and $\varphi(A, x) = 1$ otherwise.

Definition B.3.1 (Distributional Complexity)**.** Let $\epsilon \in [0, 1]$. For any distribution P on the inputs, let $\Upsilon(\epsilon)$ be the subset of $\mathcal{A}_f$ given by $\Upsilon(\epsilon) = \{A : A \in \mathcal{A}_f, \sum_{x\in\mathcal{I}} P(x) \cdot \varphi(A, x) \leq \epsilon\}$. The *Distributional complexity with error* ϵ for a boolean function f is defined as

$$\hat{U}_\epsilon(f) = \max_P \min_{A\in\Upsilon(\epsilon)} \sum_{x\in\mathcal{I}} P(x) \cdot cost(A, x).$$

Definition B.3.2 (Randomized Complexity)**.** We say that a distribution Q on the family $\mathcal{A}_f$ is ϵ-tolerant if $\max_{x\in\mathcal{I}} \sum_{A\in\mathcal{A}_f} Q(A) \cdot \varphi(A, x) \leq \epsilon$. Let $\epsilon \in [0, 1]$ and given an ϵ-tolerant distribution Q, the *randomized complexity with error* ϵ is

$$\hat{R}_\epsilon(f) = \min_Q \max_{x\in\mathcal{I}} \sum_{A\in\mathcal{A}_f} Q(A) \cdot cost(A, x).$$

Yao's minimax principle claims that $\frac{1}{2}\hat{U}(f)_{2\epsilon} \leq \hat{R}_\epsilon(f)$. However, the most common way to state it and use it in practice is the following.

Theorem B.3.1 (Minimax Principle for Montecarlo Algorithms)**.** *Given a probability distribution Q that is ϵ-tolerant on $\mathcal{A}_f$ and a probability distribution P on $\mathcal{I}$, for all $0 < \epsilon < 1/2$*

$$\frac{1}{2} \min_{A\in\Upsilon(2\epsilon)} \sum_{x\in\mathcal{I}} P(x) \cdot cost(A, x) \leq \max_{x\in\mathcal{I}} \sum_{A\in\mathcal{A}_f} Q(A) \cdot cost(A, x).$$

In simple terms, given a randomized algorithm, its worst-case running time can be lower-bounded by giving a hard distribution over the inputs on the best deterministic algorithm.

For Las Vegas algorithms we have $\hat{U}(f)_\epsilon = \hat{R}_\epsilon(f)$ and is proved by using the celebrated Von Neumann's Minimax Theorem. However, in Yao's original paper [Yao77], Theorem B.3.1 is given without proof. In the next section we give a proof by using a similar approach of Fich, Meyer auf der Heide, Radge and Widgerson [FMRW85] for Las Vegas algorithms.

B.3.2 Proof of Theorem B.3.1

As stated in the previous section, the proof will follow an approach given by Fich et al. [FMRW85, Lemma 4]. Their approach does not yield a characterization for Las Vegas algorithms (only the lower bound), but it is sufficient for our purposes.

Given that the probability distribution q is ϵ-tolerant on $\mathcal{A}_f$ we have that

$$\begin{aligned}
\epsilon &\geq \max_{x \in \mathcal{I}} \left\{ \sum_{A \in \mathcal{A}_f} Q(A) \cdot \varphi(A, x) \right\} \\
&\geq \sum_{x \in \mathcal{I}} P(x) \sum_{A \in \mathcal{A}_0} Q(A) \cdot \varphi(A, x) \\
&= \sum_{A \in \mathcal{A}_f} Q(A) \sum_{x \in \mathcal{I}} P(x) \cdot \varphi(A, x) \\
&\geq \min_{A \in \mathcal{A}_f} \left\{ \sum_{x \in \mathcal{I}} P(x) \cdot \varphi(A, x) \right\}.
\end{aligned}$$

If we replace the family $\mathcal{A}_f$ with $\Upsilon(2\epsilon)$ we see that

$$\begin{aligned}
\epsilon &\geq \max_{x \in \mathcal{I}} \left\{ \sum_{A \in \mathcal{A}_f} Q(A) \cdot \varphi(A, x) \right\} \\
&\geq \max_{x \in \mathcal{I}} \left\{ \sum_{A \in \Upsilon(2\epsilon)} Q(A) \cdot \varphi(A, x) \right\} \\
&\geq \min_{A \in \Upsilon(2\epsilon)} \left\{ \frac{1}{2} \sum_{x \in \mathcal{I}} P(x) \cdot \varphi(A, x) \right\},
\end{aligned}$$

where the second inequality follows from $\Upsilon(2\epsilon) \subseteq \mathcal{A}_f$, and the last inequality is given by the definition of $\Upsilon(2\epsilon)$ where the summation divided by 2 cannot be greater than ϵ. Hence,

$$\max_{x\in\mathcal{I}} \left\{ \sum_{A\in\mathcal{A}_f} Q(A)\cdot\varphi(A,x) \right\} \geq \frac{1}{2} \min_{A\in\Upsilon(2\epsilon)} \left\{ \sum_{x\in\mathcal{I}} P(x)\cdot\varphi(A,x) \right\}.$$

By noting that φ maps to $\{0,1\}$ and $cost(A,x)$ maps to $\mathbb{N}$, now we can safely replace the function φ in the inequality above by $cost(A,x)$ to obtain the desired inequality.

The proof given above appears in [Vil] and an alternative proof was given by Nikolov [Nik].

B.4 Quantum Decision Trees

From now on it will be more convenient to consider oracles instead of decision trees. In this model, we have a black-box (or oracle) which have access to the input, and the complexity of the algorithm is measured in terms of the number of *queries* made to this black-box in order to compute some function $f : \{0,1\}^n \to \{0,1\}$.

For $x \in \{0,1\}^n$, a classical query consists of an index $i \in [n]$ and the answer x_i. For quantum computation, the query needs to be done in a reversible manner. A natural way is to have a pair (i,d), where i is the index and $d \in \{0,1\}$, and the output will be another pair $(i, d \oplus x_i)$. Then, a quantum query is an unitary operator O_x that takes as input a quantum state $|i,d\rangle$ and produces another state $|i, d\oplus x_i\rangle$. More formally,

$$O_x |i,d,z\rangle = \begin{cases} |i,d,z\rangle & \text{if } i = 0 \text{ or } x_i = 0 \\ |i,d\oplus 1,z\rangle & \text{if } i \in [n] \text{ and } x_i = 1, \end{cases} \tag{B.1}$$

where $|z\rangle$ is an ancilla state which the oracle uses for any other computation not involved in the query.

A quantum query algorithm starts in some arbitrary state that is independent of the oracle, e.g., the state $|0\rangle$. Then proceeds on applying arbitrary unitary

operators alternated with calls to the oracle and ending with a measurement. More formally, a T-query quantum algorithm on input x computes the state

$$\left|\psi_x^T\right\rangle = U_T O_x U_{T-1} \dots U_1 O_x U_0 \left|0\right\rangle . \tag{B.2}$$

Then, this state is measured and we obtain the output which could be for example the leftmost qubit and will contain $f(x)$. The algorithm has error at most ϵ if we measure a 1 or 0 in the leftmost qubit with probability $1-\epsilon$ whenever $f(x)=1$ or $f(x)=0$ respectively.

There are mainly two lower bound techniques for quantum query complexity: Quantum adversary and polynomial methods. We recommend to the interested reader the survey by Høyer and Špalek [Hv05] for further reading. Here we will concentrate on the polynomial method which is explained next.

B.5 The Polynomial Method for Quantum Query Complexity

First we present some basic properties of polynomials. A boolean function f on n variables can be represented by an n-variate polynomial $p : \mathbb{R}^n \to \mathbb{R}$. Since $x^m = x$ for any m when restricted to boolean variables, we can deal exclusively with multilinear polynomials. Then there exists a unique multilinear polynomial such that $p(x) = f(x)$ for all $x \in \{0,1\}$. We use $deg(f)$ to denote the degree of the unique multilinear polynomial that represents f. We also define the approximate degree of f denoted by $\widetilde{deg}(f)$ whenever $|f(x) - p(x)| \leq 1/3$ for all $x \in \{0,1\}^n$ and p is of minimum degree.

Since multilinear polynomials could be difficult to handle, the following statement allow us to transform multilinear polynomials into univariate polynomials.

Definition B.5.1 (Symmetrization)**.** The *symmetrization* p^{sym} of p is defined as

$$p^{sym}(x_1, \dots, x_n) = \frac{\sum_{\pi \in S_n} p(x_{\pi(1)}, \dots, x_{\pi(n)})}{n!}$$

where S_n is the symmetric group.

Lemma B.5.1 (Minsky and Papert [MP88]). *If $p : \mathbb{R}^n \to \mathbb{R}$ is a multilinear polynomial of degree d then, there exists a polynomial $q : \mathbb{R} \to \mathbb{R}$ with degree at most d such that $q(x_1 + \cdots + x_n) = p^{symm}(x_1, \ldots, x_n)$.*

The following connection between polynomials and quantum query complexity was discovered by Beals, Buhrman, Cleve, Mosca and de Wolf [BBC+01]. The theorem and the proof presented here is by Andrew Childs.

Theorem B.5.2. *The acceptance probability of a t-query quantum algorithm for a problem with black-box input $x \in \{0,1\}^n$ is a polynomial in $x_1, \ldots, x_n$ of degree at most $2t$.*

Proof. The proof is by induction on t. When $t = 0$ the algorithm makes no queries and the success probability is independent of the input, i.e., a constant and the polynomial degree is 0.

For the induction step, note that a query maps $|i, b\rangle$ to $(-1)^{bx_i} |i, b\rangle = (1 - 2x_i) |i, b\rangle$. Hence, after each query the degree of the polynomial in the amplitude increases by at most 1. □

Appendix C

Communication Complexity

The communication model for boolean functions was proposed by Yao [Yao77]. In this model, two parties (say Alice and Bob) seek to evaluate a function $f(x, y)$ with minimal communication (i.e., minimal number of bits), where x is only known to Alice and y is only known to Bob. Today it is one of the most studied computing models with several applications spanning data structures, streaming algorithms, boolean circuits and more [KN97].

With the emergence of quantum computing, the communication model evolved naturally to a model where the parties can send qubits [Yao93]. Several authors showed the existence of exponential and quadratic gaps between the classical and quantum communication models (see the survey paper by de Wolf [dW02] for a good account of these separations).

In this thesis we focus on communication protocols where the number of players is three or more. This situation is normally called multiparty communication. In the next section we make a brief introductory overview of classical multiparty communication. For further reading on the communication model we recommend the reader the book by Kushilevitz and Nisan [KN97]. For recent results in the area see the survey papers by Lee and Schraibman [LS09b] and Razboroz [Raz10b].

C.1 Multiparty Communication

In a multiparty communication protocol there are $k \geq 2$ players seeking to compute a boolean function f. Let $f : (\{0,1\}^n)^k \to \{0,1\}$ be a function on k strings $x = (x_1, \ldots, x_k)$, where each $x_i \in \{0,1\}^n$. There are two common ways of communication between the players: The Number-In-Hand (NIH) and Number-On-Forehead (NOF) models. In NIH, player i knows only x_i, and in NOF, player i knows all inputs except x_i. Furthermore, the players can communicate in two different ways. In the *blackboard model*, we imagine that every time a player wants to send a message he does so by writing in a hypothetical black-board which all players can see. Therefore, when a player sends a message, it arrives at the same time to all players in the party. In the *message-passing* model, before the protocol starts there is a predefined fixed order of communication between the players. Thus, every time a player sends a message, he does so by sending it to only one player according to the fixed order.

The deterministic k-party communication complexity $D_k(f)$ of a boolean function f is defined as the minimum cost of a protocol, over all protocols for f, over the worst-case input. We can also define other modes of computation appropriately like 2-sided error, 1-sided error, etc.

Define the k-party communication tensor T_f of a boolean function f as an order-k tensor $T_f[x_1, \ldots, x_k] = f(x_1, \ldots, x_k)$. A NIH or NOF protocol partitions T_f in combinatorial objects called *cubes* or *cylinder intersections* respectively. We will denote both objects simply as C and the context will make explicit to which combinatorial object we are refering to.

Definition C.1.1 (Cube)**.** Define a combinatorial cube as a subset $C \subseteq (\{0,1\}^n)^k$ such that for some sets $A_1, \ldots, A_k \subseteq \{0,1\}^n$ we have $D = A_1 \times \cdots \times A_k$.

Definition C.1.2 (Cylinder Intersection)**.** Define a cylinder in the i-th dimension as a subset $C_i \subseteq (\{0,1\}^n)^k$ that does not depend on the i-th coordinate, i.e., if $(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_k) \in C_i$ then $(x_1, \ldots, x_{i-1}, x'_i, x_{i+1}, \ldots, x_k) \in C_i$ for all $x'_i \in \{0,1\}^n$. A cylinder intersection is defined as an intersection of cylinders in all dimensions $C = C_1 \cap \cdots \cap C_k$.

Let $z \in \{0,1\}$. We say that C is a z-cylinder intersection (z-cube) if $f(x) = z$ for all $x \in C$. Define a z-cover for f as a set of z-cylinder intersections (z-cubes)

that contain all z-inputs of f. Note that cylinder intersections (cubes) in a cover can be intersecting. Denote by $Cov^z(f)$ the minimal size of a z-cover of f.

Define the *cover number* $Cov(f) = Cov^0(f) + Cov^1(f)$. It is a well known fact that $\log Cov(f) \leq D_k(f)$.

C.2 Nondeterministic Communication

In this section we extend the notion of nondeterministic computation to communication. Remember that there are essentially two equivalent ways of defining a nondeterministic computation, using randomness or as a proof system. According to the first definition, a nondeterministic protocol accepts a correct input with positive probability, and rejects an incorrect input with probability one. In the second definition, a nondeterministic protocol is a deterministic protocol that receives besides the input a proof or certificate which exists if and only if the input is correct. We see in Chapter 3 of this thesis that for quantum protocols these two notions can be different.

Theorem C.2.1 ([KN97])**.** *Let $f : (\{0,1\}^n)^k \to \{0,1\}$. The k-party nondeterministic communication complexity of f is $N_k^1(f) = \log Cov^1(f)$. The k-party co-nondeterministic communication complexity of f is $N_k^0(f) = \log Cov^0(f)$. If the communication is NOF of NIH then the cover is made of cylinder intersections or cubes respectively.*

There is a tight lower bound (up to logarithmic additive factor) for $N_k^1(f)$. Put a hard distribution on the 1-inputs of f and measure the size of the largest 1-cylinder intersection (the same for cubes) [KN97, Proposition 2.15]. However, this quantity could be hard to compute or even estimate. We will see in the next section an alternative and powerful way of lower-bounding deterministic and nondeterministic communication complexity.

C.3 The Norm Bound

Linial and Shraibman [LS09c] introduced the use of factorization norms[1] as tools for proving lower bounds in randomized and quantum communication complexity in the 2-party setting. Later, their techniques were extended to multiparty communication in the works of Lee and Shraibman [LS09a] and Lee, Schecthman and Shraibman [LSS09].

In this section, we present only the generalization of the factorization norms for communication given by Lee and Shraibman [LS09a]. This generalization already covers the 2-party case at the expense of the loss of some intuition. We also make use w.l.o.g. of sign tensors (± 1 valued) instead of boolean tensors (0/1 valued).

Definition C.3.1 (Cylinder Intersection Norm)**.** Let T be an order-k sign tensor. The cylinder intersection norm is defined by

$$\mu(T) = \min\left\{\sum_i |\alpha_i| : T = \sum_i \alpha_i \chi(C_i)\right\},$$

where $\alpha_i \in \{-1, 1\}$, C_i is a cylinder intersection, and $\chi(C_i)$ is an order-k tensor where $\chi(C_i)[x_1, \ldots, x_k] = 1$ if $(x_1, \ldots, x_k) \in C_i$ and 0 otherwise.

Definition C.3.2 (Approximate Cylinder Intersection Norm)**.** Let T be an order-k sign tensor and $\alpha \geq 1$. The α-approximate cylinder intersection norm is defined as

$$\mu^\alpha(T) = \min_{T'}\{\mu(T') : 1 \leq T \circ T' \leq \alpha\},$$

where $\circ$ denotes the Hadamard (entry-wise) product. When $\alpha \to \infty$,

$$\mu^\alpha(T) = \min_{T'}\{\mu(T') : 1 \leq T \circ T'\}.$$

Lee and Shraibman showed the following lower bound on communication for any $k \geq 2$.

[1]Let M be a matrix that acts as a linear operator on two normed spaces $M : (X, \|\cdot\|_X) \to (Y, \|\cdot\|_Y)$. The operator norm $\|M\|$ is defined as the supremum of $\|Mx\|_Y$ over all $x \in X$ with $\|x\|_X = 1$. Factorization norms are defined by considering all possible ways of expressing M as the composition of two linear operators via a given middle normed space [LS09c].

Theorem C.3.1. *Let* $\alpha = 1/(1-2\epsilon)$ *and* $\epsilon \in (0, 1/2)$. $R_{k,\epsilon}(f) = \Omega(\mu^{\alpha_\epsilon}(T_f))$ *and* $N_k^1(f) = \Omega(\mu^\infty(T_f))$.

In the same piece of work it was also shown that $\mu^\infty(T_f) = 1/Disc(f)$ where $Disc(f)$ is the *generalized discrepancy* of f. This naturally implies $N_k^1(f) = \Omega(1/Disc(f))$.

Definition C.3.3 (Discrepancy)**.** Let λ be some probability measure on $(\{0,1\}^n)^k$. The discrepancy of f with respect to λ is

$$Disc_\lambda(f) = \max_C \langle T_f \circ \lambda, \chi(C) \rangle,$$

where the maximum is taken over all cylinder intersections (combinatorial cubes). The general discrepancy is

$$Disc(f) = \min_\lambda Disc_\lambda(f).$$

Appendix D

A General Approach to Coined Quantum Walk Analysis for Regular Graphs

This appendix presents a more general application of the spectral analysis approach used in Chapter2 for quantum walks on line.

D.1 General Analysis

Let (V, E) be a graph of degree d. Remember that a quantum walk on (V, E) is given by the time evolution

$$|\Psi_t\rangle = U^t |\Psi_0\rangle \tag{D.1}$$

where $U = S(C \otimes I)$. The shift operator is

$$S = \sum_{d,v} |d\rangle \langle d| \otimes |v_d\rangle \langle v| , \tag{D.2}$$

and the coin operator C is an element of $U(d)$[1]. We will assume that the initial state $|\Psi_0\rangle$ is some arbitrary superposition of directions and vertices of the graph, i.e.,

$$|\Psi_0\rangle = \sum_v |\psi_0(v)\rangle \tag{D.3}$$

[1] $U(d)$ is the group of $d \times d$ unitary matrices with matrix multiplication as the group operation.

where $|\psi_t(v)\rangle = \sum_d \alpha_t^{v,d} |d, v\rangle$ is the state at vertex v at step t.

Diagonalize operator U on the $|d, v\rangle$ basis to obtain

$$U = \sum_\lambda \lambda |\lambda\rangle \langle\lambda| . \tag{D.4}$$

Thus

$$\begin{aligned} |\Psi_t\rangle &= U^t |\Psi_0\rangle \\ &= \sum_\lambda \lambda^t \sum_{v'} \langle\lambda| \psi_0(v')\rangle |\lambda\rangle \\ &= \sum_\lambda \lambda^t \sum_{d',v'} \alpha_0^{d',v'} \langle\lambda| d', v'\rangle |\lambda\rangle , \end{aligned} \tag{D.5}$$

and the formulas for each amplitude is

$$\alpha_t^{d,v} = \sum_{\lambda,d',v'} \lambda^t \alpha_0^{d',v'} \langle\lambda| d', v'\rangle \lambda_{d,v} \tag{D.6}$$

where $\lambda_{d,v}$ is the (d, v)-th component of eigenvector $|\lambda\rangle$.

Here we can see that all what we need to compute the evolution of the quantum walk is completely determined by the eigenspectrum of operator U. In general, however, computing the eigenspectrum of a general unitary operator can be a daunting task. Hence, the need to exploit the translation symmetries of the walk with Fourier analysis.

D.2 An Application to Search

We will assume that the graph (V, E) has some marked vertex $\bar{v}$. To apply the general analysis given in the previous section, we make use of the approach developed by Shenvi et al. [SKW03] which was briefly explained in Section 2.3.2.

Let G be the Grover operator and $C' = -I_d$ where I_d is the $d \times d$ identity operator. We define the coin operator as

$$C = G \otimes I_{|V|} + (C' - G) \otimes |\bar{v}\rangle \langle\bar{v}| .$$

From this definition it is easy to see that the operator G is applied to all the unmarked nodes and only C' is applied to $\bar{v}$. This way, C' serves as a phase flip in order to amplify the amplitude on $\bar{v}$ during the search.

The search algorithm goes as follows:

1. Start with initial state $|\Psi_0\rangle$ on a superposition of all states and directions.
2. Apply $|\Psi_t\rangle = (SC)^t |\Psi_0\rangle$ for $t = \mathcal{O}(\sqrt{|V|})$.
3. Measure $|\Psi_t\rangle$ in the $|v\rangle$ basis. If the result of the measurement yields $\bar{v}$ then output "found", else "not found".

If we let $\Pi_{\bar{v}}$ be a projection operator on $\bar{v}$ the probability of finding $\bar{v}$ is thus given by

$$\begin{aligned} P_t(\bar{v}) &= \langle \Psi_t | \, \Pi_{\bar{v}} \, | \Psi_t \rangle \\ &= \sum_d |\alpha_t^{d,\bar{v}}|^2. \end{aligned} \tag{D.7}$$

D.2.1 Example: Walking the Line

For such a walk just let G be the line with n vertices. Here there are two options: 1) the line could have reflecting end-points, or 2) it could be a circle with vertices n and $-n$ connected as neighbors. These two kinds of walks were previously studied in [ABN+01].

D.2.2 Example: SAT

To obtain an algorithm for SAT just let G be a hypercube. For a SAT formula with n variables, an n-dimensional hypercube is a graph where each node corresponds to an assignment of boolean values to the variables. Two vertices are connected if and only if the Hamming distance between them is exactly one. See figure D.1.

The walk on the hypercube moves from vertex to vertex by flipping exactly one variable and checking if the formula is true on that assignment. The walk on the hypercube was thoroughly analyzed in [SKW03].

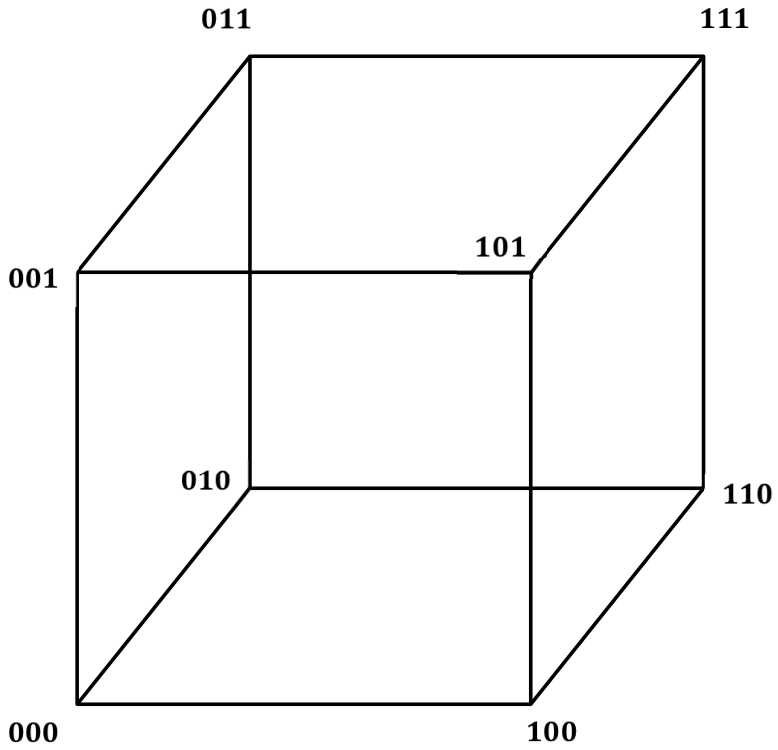

Figure D.1: Hypercube on 3 variables.

References

[Aar04] Scott Aaronson. Is Quantum Mechanics An Island In Theoryspace? Technical report, arXiv:quant-ph/0401062, January 2004.

[Aar05] Scott Aaronson. NP-complete Problems and Physical Reality. Technical report, Electronic Colloqium in Computataionl Complexity, 2005.

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[ABN+01] Andris Ambainis, Eric Bach, Ashwin Nayak, Ashvin Vishwanath, and John Watrous. One-Dimensional Quantum Walks. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 37–49, 2001.

[AFT11] Boris Alexeev, Michael Forbes, and Jacob Tsimerman. Tensor rank: Some lower and upper bounds. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity*, 2011.

[AKR05] Andris Ambainis, Julia Kempe, and Alexander Rivosh. Coins Make Quantum Walks Faster. In *Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms*, pages 1099–1102, 2005.

[Amb00] Andris Ambainis. Quantum lower bound by quantum arguments. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 636–643, 2000.

[Amb04] Andris Ambainis. Quantum Walks and their Algorithmic Applications. *International Journal of Quantum Information*, 1(4):507–518, 2004.

[Amb07] Andris Ambainis. Quantum Walk Algorithm for Element Distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007.

[Apo99] Tom Apostol. An Elementary View of Euler's Summation Formula. *The American Mathematical Monthly*, 106(5):409–418, 1999.

[AS04] Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51(4), 2004.

[BBBV97] Charles Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and Weaknesses of Quantum Computing. *SIAM Journal on Computing*, 26(5):1510–1523, October 1997.

[BBC+01] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald De Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4), 2001.

[BCW98] Harry Buhrman, Richard Cleve, and Avi Wigderson. Quantum vs. classical communication and computation. In *Proceedings of the 30th annual ACM symposium on Theory of computing*, pages 63–68, New York, New York, USA, May 1998. ACM Press.

[BdW01] Harry Buhrman and Ronald de Wolf. Communication complexity lower bounds by polynomials. In *Proceedings of the 16th Annual IEEE Conference on Computational Complexity*, pages 120–130, 2001.

[Ben80] Paul 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):563–591, 1980.

[BFS86] László Babai, Péter Frankl, and János Simon. Complexity classes in communication complexity theory. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, pages 337–347, 1986.

[BHMT02] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum Amplitude Amplification and Estimation. In Samuel Lomonaco and Howard Brandt, editors, *Quantum Computation & Information*, volume 305 of *AMS Contemporary Mathematics*, chapter 4. American Mathematical Society, May 2002.

[BV97] Ethan Bernstein and Umesh Vazirani. Quantum Complexity Theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997.

[CA08] Arkadev Chatopadhyay and Anil Ada. Multiparty communication complexity of disjointness. Technical Report TR08-002, Electronic Colloqium in Computational Complexity, 2008.

[CGW12] Andrew M. Childs, David Gosset, and Zak Webb. Universal computation by multi-particle quantum walk. Technical report, arXiv:1205.3782, 2012.

[CK11] Andrew Childs and Robin Kothari. Quantum query complexity of minor-closed graph properties. In *Proceedings of the 28th International Symposium on Theoretical Aspects of Computer Science*, pages 661–672, 2011.

[CSL08] C. M. Chandrashekar, R. Srikanth, and Raymond Laflamme. Optimizing the Discrete Time Quantum Walk using a SU(2) Coin. *Physical Review A*, 77(032326), 2008.

[Deu85] David Deutsch. Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences (1934-1990)*, 400(1818):97–117, July 1985.

[DJ92] David Deutsch and Richard Jozsa. Rapid Solution of Problems by Quantum Computation. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 439(1907):553–558, December 1992.

[dLdMV00] Lieven de Lathauwer, Bart de Moore, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[DPV09] Matei David, Toniann Pitassi, and Emanuele Viola. Improved separations between nondeterministic and randomized multiparty communication. *ACM Transactions on Computation Theory*, 1(2):5, 2009.

[dW00] Ronald de Wolf. Characterization of non-deterministic quantum query and quantum communication complexity. In *Proceedings of the 15th Annual IEEE Conference on Computational Complexity*, pages 271–278, 2000.

[dW02] Ronald de Wolf. Quantum communication and complexity. *Theoretical Computer Science*, 287(1):337–353, September 2002.

[dW03] Ronald de Wolf. Nondeterministic quantum query and quantum communication complexities. *SIAM Journal on Computing*, 32(3):681–699, 2003.

[Fey82] Richard Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7):467–488, June 1982.

[FMRW85] Faith Ellen Fich, Friedhelm Meyer Auf Der Heide, Prabhakar Lakshman Ragde, and Avi Wigderson. One, two, three...infinity: Lower bounds for parallel computation. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pages 48–58, 1985.

[GJS04] Geoffrey Grimmett, Svante Janson, and Petra Scudo. Weak Limits for Quantum Random Walks. *Physical Review E*, 69(026119), 2004.

[GKP94] Ronald Graham, Donald Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley Professional, 2nd edition, 1994.

[Gro96] Lov Grover. A Fast Quantum Mechanical Algorithm for Database Search. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, pages 212–219, 1996.

[GS10] Dmitry Gavinsky and Alexander A. Sherstov. A Separation of NP and coNP in Multiparty Communication Complexity. *Theory of Computing*, 6(10):227–245, 2010.

[Ha90] Johan Håstad. Tensor rank is NP-complete. *Journal of Algorithms*, 11(4):644–654, 1990.

[Hog00] Tad Hogg. Quantum search heuristics. *Physical Review A*, 61(5), April 2000.

[Hv05] Peter Hoyer and Robert Špalek. Lower Bounds on Quantum Query Complexity. In *arXiv:quant-ph/0509153*, 2005.

[KB06] Hari Krovi and Todd Brun. Hitting Time for Quantum Walks on the Hypercube. *Physical Review A*, 73(032341), 2006.

[KB09] Tamara Kolda and Brett Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[Kem03] Julia Kempe. Quantum Random Walks: An Introductory Overview. *Contemporary Physics*, 44(4):307–327, 2003.

[Kem05] Julia Kempe. Discrete Quantum Walk Hit Exponentially Faster. *Probability Theory and Related Fields*, 133(2), 2005.

[Ker09] Iordanis Kerenidis. Quantum multiparty communication complexity and circuit lower bounds. *Mathematical Structures in Computer Science*, 19(1):119–132, 2009.

[Kla11] Hartmut Klauck. On arthur merlin games in communication complexity. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity*, pages 189–199, 2011.

[KLL+12] Iordanis Kerenidis, Sophie Laplante, Virginie Lerays, Jeremie Roland, and David Xiao. Lower bounds on information complexity via zero communication protocols and applications. Technical report, arXiv:1204.1505, 2012.

[KLM07] Phillip Kaye, Raymond Laflamme, and Michele Mosca. *An Introduction to Quantum Computing*. Oxford University Press, 2007.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[Kon03] Norio Konno. Quantum Random Walks in One Dimension. *Quantum Information Processing*, 1(5):345–354, 2003.

[Kon05] Norio Konno. A new type of limit theorems for the one-dimensional quantum random walk. *Journal of the Mathematical Society of Japan*, 57(4):1179–1195, 2005.

[Kon08] Norio Konno. Quantum Walks. *Lecture Notes in Mathematics*, 1954:309–452, 2008.

[Kre95] Ilan Kremer. Quantum communication. Master's thesis, The Hebrew University of Jerusalem, 1995.

[LG06] François Le Gall. Quantum weakly nondeterministic communication complexity. In *Proceedings of the 31st International Symposium on Mathematical Foundations of Computer Science*, volume 4162 of *Lecture Notes in Computer Science*, pages 658–669. Springer, 2006.

[LJL+10] Thaddeus Ladd, Fedor Jelezko, Raymond Laflamme, Yasunobu Nakamura, Christopher Monroe, and Jeremy O'Brien. Quantum computers. *Nature*, 464(7285):45–53, March 2010.

[LS09a] Troy Lee and Adi Shraibman. Disjointness is hard in the multiparty number-on-the-forehead model. *Computational Complexity*, 18(2):309–336, 2009.

[LS09b] Troy Lee and Adi Shraibman. Lower Bounds in Communication Complexity. *Foundations and Trends in Theoretical Computer Science*, 3(4):263–399, 2009.

[LS09c] Nati Linial and Adi Shraibman. Lower bounds in communication complexity based on factorization norms. *Random Structures and Algorithms*, 34(3):368–394, 2009.

[LSS09] Troy Lee, Gideon Schechtman, and Adi Shraibman. Lower bounds on quantum multiparty communication complexity. In *Proceedings of the 24th IEEE Conference on Computational Complexity*, 2009.

[MHS+12] Xiao-Song Ma, Thomas Herbst, Thomas Scheidl, Daqing Wang, Sebastian Kropatschek, William Naylor, Bernhard Wittman, Alexandra Mech, Johannes Kofler, Elena Anisimova, Vadim Makarov, Thomas Jennewein, Rupert Ursin, and Anton Zeilinger. Quantum teleportation over 143 kilometers using active feed-forward. *Nature*, 489:269–273, 2012.

[Mil06] Peter Miller. *Applied Asymptotic Analysis*. American Mathematical Society, 2006.

[MP88] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, second edition, 1988.

[MR02] Cristopher Moore and Alexander Russell. Quantum Walks on the Hypercube. In *Proceedings of the 6th International Workshop on Randomization and Approximation Techniques*, pages 164–178, London, UK, 2002. Springer-Verlag.

[MSS07] Frédéric Magniez, Miklos Santha, and Mario Szegedy. Quantum algorithms for the triangle problem. *SIAM Journal on Computing*, 37(2):413–424, 2007.

[NC00] Michael Nielsen and Isaac Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[Nik] Aleksandar Nikolov. Yao's minimax principle on monte carlo algorithms. Theoretical Computer Science. URL: http://cstheory.stackexchange.com/q/12933 (version: 2012-10-13).

[NS92] Noam Nisan and Mario Szegedy. On the degree of boolean functions as real polynomials. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 462–467, 1992.

[NV00] Ashwin Nayak and Ashvin Vishwanath. Quantum walk on the line. Technical report, arXiv:quant-ph/0010117, 2000.

[Pat92] Ramamohan Paturi. On the Degree of Polynomials that Approximate Symmetric Boolean Functions. In *24th Annual ACM Symposium on Theory of Computing*, pages 468–474, 1992.

[PGKJ09] Václav Potoček, Aurél Gábris, Tamás Kiss, and Igor Jex. Optimized Quantum Random-Walk Search Algorithms on the Hypercube. *Physical Review A*, 79(012325), 2009.

[Raz99] Ran Raz. Exponential separation of quantum and classical communication complexity. In *Proceedings of the 31st Annual ACM Symposium on the Theory of Computing*, pages 358–367. ACM, 1999.

[Raz03] Alexander A. Razborov. Quantum communication complexity and symmetric predicates. *Izvestiya: Mathematics*, 67(1), 2003.

[Raz10a] Ran Raz. Tensor-rank and lower bounds for arithmetical formulas. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, pages 659–666, 2010.

[Raz10b] Alexander Razborov. *An Invitation to Mathematics: From Competitions to Research*, chapter Communication Complexity, pages 95–117. Springer, 2010.

[Rei10] Ben Reichardt. Span programs and quantum query algorithms. Technical Report TR10-110, Electronic Colloqium in Computataionl Complexity, 2010.

[RS04] Ran Raz and Amir Shpilka. On the power of quantum proofs. In *Proceedings of the 19th IEEE Annual Conference on Computational Complexity*, 2004.

[She08] Alexander A. Sherstov. The pattern matrix method for lower bounds on quantum communication. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 85–94, 2008.

[Sho94] Peter Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.

[SKJ08] Martin Stefanak, Tamas Kiss, and Igor Jex. Recurrence properties of unbiased coined quantum walks on infinite d-dimensional lattices. *Physical Review A*, 78(032306), 2008.

[SKW03] Neil Shenvi, Julia Kempe, and Birgitta Whaley. Quantum Random-Walk Search Algorithm. *Physical Review A*, 67(052307), 2003.

[Str05] Daniel Stroock. *An Introduction to Markov Processes*. Springer, 2005.

[Sze04] Mario Szegedy. Quantum Speed-Up of Markov Chain Based Algorithms. In *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science*, pages 23–41, 2004.

[Vil] Marcos Villagra. Yao's minimax principle on monte carlo algorithms. Theoretical Computer Science. URL: http://cstheory.stackexchange.com/q/12881 (version: 2012-10-12).

[vS06] Robert Špalek and Mario Szegedy. All Quantum Adversary Methods are Equivalent. *Theory of Computing*, 2:1–18, 2006.

[Won01] Roderick Wong. *Asymptotic Approximation of Integrals*. SIAM: Society for Industrial and Applied Mathematics, 2001.

[Yao77] Andrew Chi-Chih Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, pages 222–227, 1977.

[Yao79] Andrew Chi-Chih Yao. Some complexity questions related to distribuitive computing. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing*, 1979.

[Yao93] Andrew Chi-Chih Yao. Quantum circuit complexity. In *Proceedings of the 34th Annual Symposium on Foundations of Computer Science*, 1993.

Publications

Journal Papers

1. **Tensor Rank and Strong Quantum Nondeterminism in Multiparty Communication**.
   Marcos Villagra, Masaki Nakanishi, Shigeru Yamashita, Yasuhiko Nakashima. *IEICE Transactions on Information and Systems*, E96-D(1), pp. 1-8, January 2013.
2. **Quantum Walks on the Line with Phase Parameters**.
   Marcos Villagra, Masaki Nakanishi, Shigeru Yamashita, Yasuhiko Nakashima. *IEICE Transactions on Information and Systems*, E95-D(3), pp. 722-730, March 2012.

International Conferences (peer-reviewed)

1. **Tensor Rank and Strong Quantum Nondeterminism in Multiparty Communication**.
   Marcos Villagra, Masaki Nakanishi, Shigeru Yamashita, Yasuhiko Nakashima. In *Lecture Notes in Computer Science (LNCS)*, volume 7287, *Proceedings of the 9th Annual Conference on Theory and Applications of Models of Computation (TAMC)*, pp. 400-411, Springer-Verlag. Beijing, China, May 16-21, 2012.
2. **Quantum Query Complexity of Hamming Distance Estimation**.
   Marcos Villagra, Masaki Nakanishi, Shigeru Yamashita, Yasuhiko Nakashima. In *Proceedings of the 11th Asian Conference on Quantum Information Science (AQIS)*, pp. 103-104. Busan, Korea. August 23-27, 2011.

3. **Asymptotics of Quantum Walks on the Line with Phase Parameters**.
   Marcos Villagra, Masaki Nakanishi, Shigeru Yamashita, Yasuhiko Nakashima. In *Proceedings of the 10th Asian Conference on Quantum Information Science (AQIS)*, pp. 163-164. Tokyo, Japan. August 27-31, 2010.

International Conferences (non-peer-reviewed)

1. **Hamming Distance Estimation with Quantum Queries**.
   Marcos Villagra, Masaki Nakanishi, Shigeru Yamashita, Yasuhiko Nakashima. *8th Canadian Student Conference on Quantum Information*. Québec, Canada, June 17-18, 2011.
2. **Discrete Quantum Walks on the Line with Phase Parameters**.
   Marcos Villagra, Masaki Nakanishi, Shigeru Yamashita, Yasuhiko Nakashima. *International Conference on Quantum Information and Technology (ICQIT)*. Tokyo, Japan, December 2-5, 2009.

Index