University of Latvia
Faculty of Computing

Nikolajs Nahimovs

The Power and the Limits of Quantum Automata and Search Algorithms

Doctoral Thesis

Area: Computer Science
Sub-Area: Mathematical Foundations of Computer Science

Scientific Advisor:
Dr. Comp. Sci., Prof. Andris Ambainis

Riga 2012

EIROPAS SAVIENĪBA

IEGULDĪJUMS TAVĀ NĀKOTNĒ

LATVIJAS UNIVERSITĀTE
ANNO 1919

This work has been supported by the European Social Fund within the projects "Support for Doctoral Studies at University of Latvia" Nr. 2009/0138/1DP/ 1.1.2.1.2./09/IPIA/ VIAA/004 and "Applications of computer science and its links to quantum physics" Nr. 2009/0216/ 1DP/1.1.1.2.0/09/APIA/VIAA/044.

Abstract

Quantum computation is the field that investigates properties of models of computation based on the laws of the quantum mechanics. Quantum computation has many different sub-fields and research directions, starting from very physical ones and ending with purely algorithmic problems.

The thesis is dedicated to algorithmic aspects of quantum computation and provides results in three directions:

- **Quantum finite automata**

  We study space-efficiency of one-way quantum finite automata compared to one-way classical finite automata. We improve best known exponential separation [AF98] between quantum and classical finite automata.

- **Analysis of Grover's algorithm**

  We study fault-tolerance of Grover's algorithm to logical faults. We generalize the model of logical faults by [RS08] and present several new results.

- **Quantum walks**

  We study search by quantum walks on two-dimensional grid. We improve (speed-up) quantum walk search algorithm by [AKR05]. The comparable improvement has been already achieved by other groups using other methods. Nevertheless, we find our approach as interesting and promising.

Acknowledgements

First of all I want to thank my supervisor – professor Andris Ambainis – for many helpful ideas and suggestions, which was a great help during this research.

I thank my colleagues – Aleksandrs Rivošs, Dmitrijs Kravčenko, Agnis Skuškovniks and Abuzer Yakaryilmaz – for many pleasant, interesting, and useful scientific discussions.

Also I want to thank the co-authors of the various papers that make up this thesis: Andris Ambainis, Aleksandrs Rivošs, Dmitrijs Kravčenko, and Artūrs Bačkurs.

I thank Igor Shparlinski for pointing out the papers [AI+00] and [Bou05] that helped to complete the quantum automata part of this research.

Finally, I want to thank my parents, family, and friends for many things beyond the scope of this thesis.

CONTENTS

Part I Introduction and definitions 1

1. Introduction 2
1.1 Relevance of the thesis 2
1.2 Objectives of the research 3
1.3 Methods of the research 3
1.4 Synopsis 4
1.5 Theoretical and practical significance of the results 5
1.6 Approval of the results 6
1.7 Structure of the thesis 9

2. Quantum computation model 10
2.1 Preliminaries, terminology and notation 11
2.2 Quantum states 12
2.3 Operations on quantum states 14
2.4 General quantum states and operations 15

Part II Quantum finite automata 17

3. One-way quantum finite automata 18
3.1 One-way quantum finite automata models 19
3.1.1 Moore-Crutchfield (measure-once) model 19
3.1.2 Kondacs-Watrous (measure-many) model 20
3.1.3 General 1-way quantum finite automata 20

3.2 Space-efficiency of 1-way quantum finite automata 21

4. Space-efficient quantum automata 22
4.1 Summary of results 23
4.2 Used theorems 23
4.3 Probabilistic construction 25
4.4 Explicit constructions 28
4.4.1 The first construction: cyclic sequences 28
4.4.2 The second construction: AIKPS sequences 31
4.5 Conclusions 33

Part III Analysis of Grover's algorithm 34

5. Quantum query model and Grover's algorithm 35
5.1 Quantum query model 36
5.2 Grover's quantum search algorithm 39

6. Optimality of Grover's algorithm 42
6.1 Summary of results 43
6.2 Average number of steps of Grover's algorithm 43
6.3 Conclusions 46

7. Grover's algorithm with faulty oracle : omitted query model 47
7.1 Technical preliminaries 48
7.2 Model and results 48
7.3 Related work 49
7.4 Omitting a single query 50
7.5 Omitting multiple queries 51
7.6 Probability distribution of the median 56
7.7 Conclusions 59

8. Grover's algorithm with faulty oracle: independent error model . . . 60
8.1 Technical preliminaries . . . 61
8.2 Model and results . . . 62
8.3 Related work . . . 63
8.4 Limiting behaviour of Grover's algorithm with errors . . . 65
8.5 Convergence speed of Grover's algorithm with errors . . . 68
8.6 Conclusions . . . 72

Part IV Quantum walks 73

9. Search by quantum walks on two-dimensional grid . . . 74
9.1 [AKR05] quantum walk search algorithm . . . 75
9.2 Summary of results . . . 77
9.3 Related work . . . 78
9.4 Proofs . . . 81
9.4.1 Approximation of the final state of the quantum walk . . 81
9.4.2 Bounds on the probability of being close to the marked location . . . 86
9.5 Conclusions . . . 96

10. Conclusions . . . 97

LIST OF FIGURES

4.1 A cyclic sequence giving better results than most of random sequences 30

4.2 A cyclic sequence giving worse results than most of random sequences 31

5.1 Classical device implementing one-bit function f. 36

5.2 Quantum device implementing one-bit function f with an auxilary qubit. 37

6.1 Growth of success probability of Grover's algorithm compared to linear growth 44

8.1 Grover's algorithm with different error probabilities for different marked elements, $n = 1024$. 63

9.1 Probability by distance, one marked location, grid size 1024×1024, normal scale. 77

9.2 Probability by distance, one marked location, grid size 1024×1024, logarithmic scale. 78

9.3 Probability to be within $\sqrt{N}$ neibourghood from the marked location. 79

LIST OF TABLES

4.1 Comparison of error values for cyclic and random sequences . . 29
4.2 Error values for different generators 30
4.3 Minimal generators for different p 32

Part I

INTRODUCTION AND DEFINITIONS

1. INTRODUCTION

1.1 Relevance of the thesis

Today's computers – both theoretical models and practical implementations – are based on the laws of classical physics [Pen89]. Classical physics, however, does not capture all known physical effects, which (at least in theory) can lead to more powerful models of computation. *Quantum computation* is the field that investigates properties of models of computation based on the laws of the quantum mechanics – the generalization of classical physics, describing the nature at the elementary particle level.

Quantum computation as a separate field was born in 80s with a realization that it is not possible to efficiently simulate (model) quantum mechanics on classical computers [Fey82]. The above problem appears as a result of non-local nature of quantum mechanics; one needs exponentially many coefficients to describe an N-particle quantum system. Quantum computer, or computation using a quantum mechanical system, was proposed as a solution to the above problem. It has also been conjectured that quantum computers will allow to exponentially speed-up a classical computation and, thus, solve otherwise unsolvable problems (such as NP-complete problems). At the moment, the question if quantum computers can provide an exponential parallelism is still open.

During the 80s the so-called standard model of quantum computation has been developed and shown to be a generalization of a classical computation. That is a general-purpose quantum computer can solve any problem solved by a classical computer using the same amount of computational resources (time, space, etc.) [Wat06]. The opposite may not be true. At the moment, there are problems which can be efficiently (in polynomial time) solved on a quantum computer, but no polynomial-time classical algorithm is known [Sho97].

Since then many brilliant and important results were found, starting from Grover's algorithm [Gro96], which solves the unstructured search problem of size N in just $O(\sqrt{N})$ steps, and ending with recently developed quantum-

walk-based search algorithms [CC+03, Amb07, Sze04, BS06].

Grover's quantum search algorithm is known to be optimal in both amount of used memory and number of steps. Many other quantum algorithms are not optimal. Thus, an improvement of existing algorithms and development of new algorithms (especially if new algorithms are based on novel ideas) is one of major research directions.

Another important research direction is tolerance of existing quantum algorithms to physical and logical faults. It has been shown that in "faulty environment" some quantum algorithms may lose their superiority over classical algorithms [SBW03, RS08]. Therefore, study of fault-tolerance of existing algorithms and development of methods of protection of certain classes of algorithms from certain types of errors is of great interest [BN+05].

Quantum computer can solve many important computational problems faster than a classical computer. However, quantum computation also gives us a better understanding of potential and limits of classical computation. Development of quantum computation has provided a new tools (methods of proof and analysis) and insights which are useful in classical theory of computing [DW11].

1.2 Objectives of the research

The main objective of the research done within the thesis is the study of power and limits of quantum computation model, that is:

- To study known effects and properties of quantum computation model which make quantum algorithms superior over classical algorithms.
- To find new effects of such type.
- To use found effects to improve existing algorithms and construct new algorithms, with the main emphasis on the search algorithms.
- To analyse in which situations quantum algorithms lose their superiority over classical algorithms.

1.3 Methods of the research

The research done within the thesis is based on

- Construction of mathematical models of processes (problems) of interest.
- Numerical study of the models, which include implementation of simulation of the process being studied and analysis of results of simulation using tools from probability theory and mathematical statistics.
- Analytical study of the models using mathematical formalism of theory of quantum computation and quantum information, as well as results from computational complexity and probability theory.

1.4 Synopsis

The thesis summarizes research results in three directions:

Quantum finite automata

Quantum finite automata are a mathematical model for quantum computers with limited memory.

- We improve exponential separation between quantum and classical finite automata, for the same computational problem as in [AF98]. The construction in [AF98] requires $O(poly\left(\frac{1}{\epsilon}\right)\log p)$ states, where ϵ is a probability of error. Our construction requires $O(\frac{1}{\epsilon}\log p)$ states.

The results of this part are joint work with A. Ambainis. The author's contribution is 60%.

Analysis of Grover's algorithm

Grover's algorithm is a quantum search algorithm solving the unstructured search problem [Gro96].

- We show that despite the Grover's algorithm being optimal [Zal99] it is still possible to reduce the average number of steps required to find the marked element (by approximately 12.14%) by ending the computation earlier and repeating the algorithm if necessary.
- We study fault-tolerance of Grover's algorithm to logical faults in [RS08] model for a small number of errors. We show that $k \ll t$ (t is a number of steps of the algorithm) uniformly distributed independent errors change

the sequence of transformations of the algorithm from $(DQ)^t$ to $(DQ)^T$, where T is the random variable with expectation $O\left(\frac{t}{k}\right)$ and standard deviation $O\left(\frac{t}{\sqrt{k}}\right)$.

- We generalize the model of logical faults of [RS08]. We analyse the limiting behaviour of Grover's algorithm for a large number of steps and prove the existence of limiting state ρ_{lim}. If we measure ρ_{lim}, the probability of getting one of the marked states $i_1, \ldots, i_k$ is $\frac{k}{k+1}$. We show that convergence time is $O(N)$.

The results of this part are joint work with A. Ambainis, A. Bačkurs and A. Rivošs. The author's contribution is 70%.

Quantum walks

Quantum walks are quantum counterparts of random walks [Amb03].

- We study search by quantum walks on two-dimensional $\sqrt{N} \times \sqrt{N}$ grid. We speed-up quantum walk search algorithm by [AKR05] from $O(\sqrt{N}\log(N))$ to $O(\sqrt{N\log(N)}$ steps.

The results of this part are joint work with A. Ambainis, A. Bačkurs and A. Rivošs. The author's contribution is 50%.

1.5 Theoretical and practical significance of the results

Similarly to the summary of the results we give significance of the results for each research direction separately.

Quantum finite automata

At the moment no general purpose quantum computer exist. Even then built, quantum computers will probably consist of two parts: a classical part and a small but expensive quantum part. This motivates the study of systems with a smallest possible quantum mechanical part.

We study space-efficiency of one-way quantum finite automata compared to one-way classical finite automata. We improve best known exponential separation [AF98] between quantum and classical finite automata, that is we show

that quantum automata can be much more efficient than classical automata.

Analysis of Grover's algorithm

Grover's algorithm is one of most important and widely known quantum algorithms. It solves the unstructured search problem of size N in $O(\sqrt{N})$ queries, providing a significant speed-up over any deterministic or probabilistic algorithm solving the same problem. Many other quantum algorithms use Grover's algorithm as a subroutine.

The running time of the algorithm, however, is very sensitive to errors [RS08]. We study fault-tolerance of Grover's algorithm to logical faults. We generalize the model of logical faults by [RS08] and present several new results. Both the results and used methods can be applied to wide range of other quantum query algorithms (mentioned in summary of corresponding chapters) and serve as a basis for further research.

Quantum walks

Quantum walks have been useful to design quantum algorithms for a variety of problems. In many of those applications, quantum walks are used as a tool for search.

We study search by quantum walks on two-dimensional grid. We improve (speed-up) quantum walk search algorithm by [AKR05]. Our improvement is based on effect which has never been studied before. It opens several new questions and has potential to be extended to graphs of other types. A comparable improvement has been already achieved by other groups using other methods [Tul08, KM+10]. Nevertheless, we find our approach as interesting and promising.

Overall, the results are of theoretical nature and serve as a basis for further research.

1.6 Approval of the results

Author of the thesis studied quantum computation problems in the following research projects: University of Latvia project "Jaunas zinātniskās grupas izveide kvantu skaitļošanā un datorzintu teorijā" ("Creating a new research group in quantum computing and theory of computing") and ESF project "Datorzinātnes pielietojumi un tās saiknes ar kvantu fiziku" ("Applications of

computer science and its links to quantum physics").

The results of the research done within the thesis are reflected in the following publications. The author's contribution is 50-70%.

1. [AN08] A. Ambainis, N. Nahimovs.
   Improved constructions of quantum automata.
   Proceedings of TQC 2008, Lecture Notes in Computer Science, 5106:47-56, 2008.

2. [AN09] A. Ambainis, N. Nahimovs.
   Improved constructions of quantum automata.
   Theoretical Computer Science (special issue on probabilistic and quantum automata), 410:1916-1922, 2009.

3. [NR10] N. Nahimovs, A. Rivošs.
   A note on the optimality of the Grover's algorithm.
   Scientific Papers University of Latvia, 756:221-225, 2010.

4. [KNR12] D. Kravchenko, N. Nahimovs, A. Rivosh.
   On fault-tolerance of Grover's algorithm.
   Scientific Papers University of Latvia, 787:135-145, 2012.

5. [AB+12] A. Ambainis, A. Bačkurs, N. Nahimovs, R. Ozols, A. Rivosh.
   Search by quantum walks on two dimensional grid without amplitude amplification.
   Proceedings of TQC 2012, Lecture Notes in Computer Science, 7582:87-97, 2012.

6. [AB+13] A. Ambainis, A. Bačkurs, N. Nahimovs, A. Rivosh.
   Grover's algorithm with errors.
   Proceedings of MEMICS 2012, Lecture Notes in Computer Science, 7721:180-189, 2013.

The results of the thesis were presented at the following international conferences and workshops:

1. TQC 2008 (The 3rd Workshop on Theory of Quantum Computation, Communication, and Cryptography), Tokyo, Japan, 2008.
   Presentation: *Improved constructions of quantum automata.*

2. CEQIP 2008 (5th Central European Quantum Information Processing Workshop), Telč, Czech Republic, 2008.
   Presentation: *Space-efficient quantum automata.*

3. CEQIP 2009 (5th Central European Quantum Information Processing Workshop), Jindricuv Hradec, Czech Republic, 2009.
   Poster presentation: *Grover's algorithm with probabilistic solutions.*

4. 68. LU konference, Rīga, Latvia, 2010.
   Presentation: *Kvantu meklēšana ir ātrāka, ja to pārtrauc priekšlaikus.*

5. CEQIP 2010 (7th Central European Quantum Information Processing Workshop), Valtice, Czech Republic, 2010.
   Poster presentation: *On fault-tolerance of Grover's algorithm.*

6. Joint Estonian-Latvian Theory Days, Rakari, Latvia, 2011.
   Presentation: *Constant factor improvement of the Grover's algorithm.*

7. CEQIP 2011 (8th Central European Quantum Information Processing Workshop), Znojmo, Czech Republic, 2011.
   Poster presentation: *Search by quantum walks on two dimensional grid without amplitude amplification.*

8. QIP 2012 (Quantum Information Processing), Montréal, Québec, Canada, 2011.
   Poster presentation: *Search by quantum walks on two-dimensional grid without amplitude amplification.*

9. 70. LU konference, Rīga, Latvia, 2012.
   Presentation: *Kvantu klejošana uz divdimensiju režģa.*

10. TQC 2012 (The 7rd Workshop on Theory of Quantum Computation, Communication, and Cryptography), Tokyo, Japan, 2012.
    Presentation: *Search by quantum walks on two-dimensional grid without amplitude amplification.*

11. CEQIP 2012 (9th Central European Quantum Information Processing Workshop), Smolenice, Slovakia, 2012.
    Presentation: *Better algorithms for search by quantum walks on two-dimensional grid.*

12. MEMICS 2012 (Annual Doctoral Workshop on Mathematical and Engineering Methods in Computer Science), Znojmo, Czech Republic, 2012.
    Presentation: *Grover's algorithm with errors.*

1.7 Structure of the thesis

The thesis consists of an abstract, preface, acknowledgements, table of contents, list of figures, list of tables, 9 chapters organized into 4 parts and bibliography. The thesis is 113 pages long.

Part I provides an introduction. Chapter 1 gives an overview of the thesis. Chapter 2 provides the necessary background on quantum information and the standard model of quantum computation.

Part II is dedicated to quantum finite automata. Chapter 3 gives an overview of quantum automata models and their relation to classical automata. Chapter 4 contains new results on 1-way quantum finite automata. The results of this chapter were published in [AN08, AN09].

Part III is related to one of most popular quantum search algorithms – Grover's algorithm. Chapter 5 introduces quantum query model and Grover's algorithm. In chapter 6 we study the optimality of Grover's algorithm. The results of this chapter were published in [NR10]. Chapters 7 and 8 study fault-tolerance of Grover's algorithm for different models of logical faults. The results of these chapters were published in [KNR12, AB+13].

Part IV is dedicated to quantum walks in two dimensions. Chapter 9 introduces quantum walks on two-dimensional grid and describes an improved version of [AKR05] quantum walk search algorithm. The results of this chapter were published in [AB+12].

2. QUANTUM COMPUTATION MODEL

This chapter describes the *standard model* of quantum computation. The standard model can be seen as a generalization of classical computation [Wat06]. It replaces classical bits with two state quantum systems (called quantum bits or qubits) and enlarges the set of possible operations to include all operations allowed by quantum mechanics. This model is the most widely used model of quantum computation.

There are also more exotic models of quantum computation. In case of *measurement-only model* the computation is done by preparing a quantum system in a predefined state (independent of the problem) and then observing its particles in some specific order (dependent on the problem). It has been shown that this type of computation is equivalent to the standard model of quantum computation [Joz05].

In case of the *adiabatic model* of quantum computation, which is a continuous time model, the computation (the evolution of the quantum state) is done by a time-dependent Hamiltonian (physically implementable operation) that slowly changes between an initial Hamiltonian, whose lowest-energy state is easy to construct, and a final Hamiltonian, whose lowest-energy state describes a solution of a problem [FG+00]. Usually, the final Hamiltonian can be constructed based on a structure of the problem without knowing an exact solution. The laws of quantum mechanics guarantee that the system remains in the lowest-energy state, so that at the end of this process the state of the system describes the solution to the problem. The speed at which the Hamiltonians can be changed one into another depends on a problem and usually is hard to estimate. The adiabatic model is also equivalent to the standard model of quantum computation.

In the following sections we introduce the standard model by describing three of its components: a set of possible states of a quantum system, a set of transformations, which can be applied to the system, and the process of observation of the state of the system. A more detailed description of the standard model of quantum computation can be found in [Wat06] or [KLM07].

2.1 Preliminaries, terminology and notation

We assume familiarity with complex numbers, basics of matrix algebra (matrix addition, matrix multiplication, inverse matrix, etc.) and basic concepts of linear algebra (vector spaces, linear independence, linear span, etc.) [Lay11].

We use A_{ij} to denote the (i,j)-th entry of the matrix A and v_i to denote the i-th value of the vector v. We use A^* for the *conjugate transpose* of matrix A – the matrix obtained by transposing A and taking the complex conjugates of all entries.

We use $\mathbb{C}^d$ to denote the d–dimensional complex space. We use I_d to denote the $d \times d$ identity matrix, which has 1s on its diagonal and 0s elsewhere. We usually omit the subscript d when the dimension is clear from context.

The *inner product* of vectors v and w is a scalar $v^*w = \sum_i v_i^* w_i$. The *outer product* of vectors v and w is a matrix vw^*. The complex number λ is an *eigenvalue* of square matrix A with corresponding *eigenvector* v if $Av = \lambda v$.

The *tensor* or *Kronecker product* of $n \times m$ matrix A and $k \times l$ matrix B is $nk \times ml$ matrix

$$A \otimes B = \begin{pmatrix} A_{11}B & \dots & A_{1m}B \\ A_{21}B & \dots & A_{2m}B \\ \vdots & \ddots & \vdots \\ A_{n1}B & \dots & A_{nm}B \end{pmatrix}.$$

The tensor product satisfies following properties [Wat06]:

- $(\alpha A) \otimes B = A \otimes (\alpha B) = \alpha(A \otimes B)$ for any scalar α
- $(A \otimes B) \otimes C = A \otimes (B \otimes C)$ (associativity)
- $A \otimes (B + C) = A \otimes B + A \otimes C$ (distributivity)
- $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$

However, the tensor product is not commutative; in general

$$A \otimes B \neq B \otimes A.$$

2.2 Quantum states

In this section we describe possible states of a quantum system and introduce the corresponding notation.

Quantum bit

Quantum bit or *qubit* is a two-level quantum system (system with two possible states). We denote its states as $|0\rangle$ and $|1\rangle$ and refer to them as *basis* or *classical states*.

According to quantum mechanics a qubit can be not only in its basis states but also in any state

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

where α and β are complex numbers with the property $|\alpha|^2 + |\beta|^2 = 1$. Therefore, a state of a quantum bit is as a unit vector in $\mathbb{C}^2$. We call α and β *amplitudes* and $|\psi\rangle$ – a *superposition* of $|0\rangle$ and $|1\rangle$ [1].

We use column vectors to describe a state of a quantum system. We identify basis states with the vectors

$$|0\rangle \stackrel{def}{=} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad |1\rangle \stackrel{def}{=} \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Thus, the state $\alpha|0\rangle + \beta|1\rangle$ means

$$\alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}.$$

We use $\langle\psi|$ to denote the conjugate transpose of $|\psi\rangle$. That is $\langle\psi|$ is a row vector whose entries are complex conjugates of $|\psi\rangle$ entries. $\langle\psi|\phi\rangle$ denotes the inner product of $|\psi\rangle$ and $|\phi\rangle$, i.e. a scalar. $|\psi\rangle\langle\phi|$ denotes the outer product of $|\psi\rangle$ and $|\phi\rangle$ which is a matrix.

General case

Suppose we have a quantum system with k possible states. We denote the states as $|1\rangle, |2\rangle, \ldots, |k\rangle$. The state of the system is a unit vector in $\mathbb{C}^k$:

[1] The state introduced above is called *pure* state. The more general *mixed* states – probabilistic mixture of pure states – are introduced in section 2.4.

$$|\psi\rangle = \alpha_1|1\rangle + \alpha_2|2\rangle + \ldots + \alpha_k|k\rangle,$$

where $\sum_{i=1}^{k} |\alpha_k|^2 = 1$. Similarly to a single qubit case $\alpha_1, \ldots, \alpha_k$ are called amplitudes and $|\psi\rangle$ is called a superposition of $|1\rangle, \ldots, |k\rangle$.

Multiple qubits

Suppose we have n quantum bits. The state of the system is a unit vector in $\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \ldots \otimes \mathbb{C}^2 = \mathbb{C}^{2^n}$ space. This space is spanned by 2^n basis (classical) states that are tensor products of basis states of individual qubits:

$$|x_1\rangle \otimes |x_2\rangle \otimes \ldots \otimes |x_n\rangle,$$

where $x_1 \in \{0, 1\}, \ldots, x_n \in \{0, 1\}$. For simplicity we often omit $\otimes$ symbol and write $|\psi\rangle \otimes |\phi\rangle$ as $|\psi\rangle|\phi\rangle$, or $|\psi, \phi\rangle$, or even $|\psi\phi\rangle$. Thus, the general state of an n-qubit quantum system is

$$|\psi\rangle = \sum_{x_1,\ldots,x_n \in \{0,1\}} \alpha_{x_1,\ldots,x_n} |x_1, \ldots, x_n\rangle,$$

where $\sum_{x_1,\ldots,x_n \in \{0,1\}} |\alpha_{x_1,\ldots,x_n}|^2 = 1$.

2.3 Operations on quantum states

A quantum system can undergo two types of operations: a unitary evolution (a sequence of unitary transformations) and a measurement.

Unitary evolution

A unitary transformation is a linear transformation U on $\mathbb{C}^k$ that preserves l_2 norm – any $|\psi\rangle$ with $||\psi|| = 1$ is mapped to $|\psi'\rangle$ with $||\psi'|| = 1$. We use $U|\psi\rangle$ to denote a vector to which U maps $|\psi\rangle$.

Transformation U has a natural interpretation in terms of matrices. We identify U with the $k \times k$ matrix where i^{th} column is equal to $U|i\rangle$. Unitary transformations preserve an angle between vectors. Therefore, columns of matrix corresponding to U must be orthogonal (as are vectors $|i\rangle$).

Measurement

Measurement is the process of getting the information out of a quantum system.

Suppose we have a state:

$$|\psi\rangle = \alpha_1|1\rangle + \alpha_2|2\rangle + \ldots + \alpha_k|k\rangle.$$

The simplest type of measurement is *the measurement in the computational basis*. For the above state it gives $|i\rangle$ with a probability $|\alpha_i|^2$. This is why we require $\sum_{i=1}^{k} |\alpha_i|^2$ to be 1. After the measurement, the state of the system changes to $|j\rangle$ (the outcome of the measurement). In other words measurement "collapses" the superposition $|\psi\rangle$ into a classical state $|j\rangle$. Repeated measurements will give $|j\rangle$ with probability 1.

A more general type of measurement is a *projective* or *von Neuman measurement*. We decompose $\mathbb{C}^k$ into orthogonal subspaces $\mathcal{H}_1, \ldots, \mathcal{H}_m$ so that

$$\mathbb{C}^k = \mathcal{H}_1 \oplus \ldots \oplus \mathcal{H}_m.$$

A measurement of a pure state $|\psi\rangle$ gives the result i with a probability $||P_i|\psi\rangle||^2$, where $P_i|\psi\rangle$ denotes a projection of $|\psi\rangle$ to the subspace $\mathcal{H}_i$. The state after the measurement changes to $\frac{P_i|\psi\rangle}{||P_i|\psi\rangle||}$. Repeated measurements will give i with probability 1.

2.4 General quantum states and operations

In the previous sections we described the state of a quantum system by a definite state vector. Such states are commonly referred to as *pure* states. However, there are situations when all we know about a quantum system is that it is in a specific set of states with corresponding probabilities. In this case the system is said to be in *mixed* state.

Mixed states occur as a result of a stochastic process, such as interaction of a quantum system with the environment or decoherence. Also, if a quantum system consists of two or more subsystems that are entangled, then each individual subsystem must be treated as a mixed state even if the complete system is in a pure state.

The density matrix formalism was introduced by John von Neumann (and independently by Lev Landau and Felix Bloch) in 1927 to describe a statistical state of the quantum system. We introduce only a part of general quantum state formalism, sufficient to understand the following chapters. More profound overview of general quantum states and operators can be found in [KLM07].

Suppose we have a quantum system and we known it to be in the set of states $|\psi_1\rangle, \ldots, |\psi_k\rangle$ with probabilities $p_1, \ldots, p_k$ (the probabilities must sum to 1). The collection $\{(p_1, |\psi_1\rangle); (p_2, |\psi_2\rangle); \ldots; (p_k, |\psi_k\rangle)\}$, which describes possible states along with associated probabilities, is called the *mixture*. It is not convenient to use mixtures as a mathematical description of a state of a quantum system.

To describe a mixed state in a more convenient way, we use *density matrices*. For a pure state $|\psi\rangle = \sum_{i=1}^{n} \alpha_i |i\rangle$ the density matrix is given by

$$\rho = |\psi\rangle\langle\psi| = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix} \begin{pmatrix} \alpha_1^* & \alpha_2^* & \ldots & \alpha_n^* \end{pmatrix} = \begin{pmatrix} |\alpha_1|^2 & \alpha_1\alpha_2^* & \ldots & \alpha_1\alpha_n^* \\ \alpha_2\alpha_1^* & |\alpha_2|^2 & \ldots & \alpha_2\alpha_n^* \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_n\alpha_1^* & \alpha_n\alpha_2^* & \ldots & |\alpha_n|^2 \end{pmatrix}.$$

For a mixed state $(p_i, |\psi_i\rangle)$ the density matrix is the sum of density matrices of possible pure states with associated probabilities:

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|.$$

If two quantum systems $(p_i, |\psi_i\rangle)$ and $(q_i, |\phi_i\rangle)$ have the same density matrices they are physically indistinguishable [KLM07]. Any measurement will give same probability distribution of outcomes. Thus, density matrix contains all information about the quantum state.

The diagonal entries of a density matrix contain probabilities to find the system in the corresponding state, if we perform a measurement in the computational basis. Thus, their sum is equal to 1.

Consider an operation on mixed state. If we apply unitary operation U to the mixed state $(p_i, |\psi_i\rangle)$ we get mixed state $(p_i, U|\psi_i\rangle)$. The corresponding density matrix is

$$\rho' = \sum_i p_i U|\psi_i\rangle\langle\psi_i|U^\dagger = U\left(\sum_i p_i|\psi_i\rangle\langle\psi_i|\right)U^\dagger = U\rho U^\dagger.$$

Similarly, if we apply a stochastic operation (q_i, U_i) to the mixed state $(p_i, |\psi_i\rangle)$ we get

$$\rho' = \sum_j q_j U_j \rho U_j^\dagger.$$

Stochastic operations are just a special case of a general quantum operation. Any operation Φ that can be written as

$$\Phi(\rho) = \sum_i A_i \rho A_i^\dagger$$

for some matrices $A_1, \ldots, A_k$ satisfying

$$\sum_i A_j A_j^\dagger = I$$

can be physically implemented.

Part II

QUANTUM FINITE AUTOMATA

3. ONE-WAY QUANTUM FINITE AUTOMATA

Quantum finite automata are a mathematical model for quantum computers with limited memory. They are related to more general models of quantum computers (such as quantum Turing machines and quantum circuits) in a similar way classical finite automata are related to general models of classical computers (such as Turing machines). A quantum finite automaton has a finite state space and applies a sequence of transformations, corresponding to the letter of the input word to this state space. At the end, the state of the quantum automaton is measured and the input word is accepted or rejected, depending on the outcome of the measurement.

Similarly to a classical case several types of quantum finite automata has been defined. The introduced models differ by their properties, starting from classical ones, such as input head type (one-way, two-way), and ending with quantum properties, such as a way we perform the measurement (measure-once, measure-many).

Most commonly, finite automata (including quantum finite automata) are studied in 1-way model, where the transformations, corresponding to the letters of the input word, are applied in the order of the letters in the word, from the left to the right.

In this chapter we give an overview of 1-way quantum finite automata models and compare their computational power and space efficiency to classical (deterministic and probabilistic) 1-way finite automata.

3.1 One-way quantum finite automata models

First quantum automata models, such as [CM00] or [KW97], were defined in a restricted way (were naive "quantization" of classical automata models). These models recognize a subset of regular languages (for example, [KW97] demonstrates regular languages which can not be recognized by these models) and ,thus, are weaker then deterministic and probabilistic 1-way finite automata, which recognize all regular languages.

Later more general QFA models were introduced, which can recognize any regular language [BMP03, Cia01]. As QFA can be simulated by DFA with exponentially many states, QFA can not recognize languages not recognized by DFA. Thus, QFA and DFA recognize the same set of languages – regular languages – and have equivalent computational power.

One of the reasons why restricted quantum automata models are still being used is that they are simple but powerful enough to demonstrate many of QFA succinctness (space-efficiency) results.

3.1.1 Moore-Crutchfield (measure-once) model

We consider 1-way quantum finite automata (QFA) as defined in [CM00]. Namely, a 1-way QFA is a tuple $M = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej})$ where Q is a finite set of states, Σ is an input alphabet, δ is a transition function, $q_0 \in Q$ is a starting state, Q_{acc} and Q_{rej} are sets of accepting and rejecting states and $Q = Q_{acc} \cup Q_{rej}$. ¢ and \$ are symbols that do not belong to Σ. We use ¢ and \$ as the left and the right end-marker, respectively. The *working alphabet* of M is $\Gamma = \Sigma \cup \{¢, \$\}$.

For $q \in Q$, $|q\rangle$ denotes the unit vector with value 1 at q and 0 elsewhere. The state of an automaton is a superposition of $|q\rangle$.

The transition function δ maps $Q \times \Gamma \times Q$ to $\mathbb{C}$. The value $\delta(q_1, a, q_2)$ is the amplitude of $|q_2\rangle$ in the superposition of states to which M goes from $|q_1\rangle$ after reading a. For $a \in \Gamma$, V_a is a linear transformation on $l_2(Q)$ defined by

$$V_a|q_1\rangle = \sum_{q_2 \in Q} \delta(q_1, a, q_2)|q_2\rangle.$$

We require all V_a to be unitary.

The computation of a QFA starts in the superposition $|q_0\rangle$. Then transformations corresponding to the left end-marker ¢, the letters of the input word x

and the right end-marker \$ are applied. The transformation corresponding to $a \in \Gamma$ is V_a. If the superposition before reading a is $|\psi\rangle$, then the superposition after reading a is $V_a|\psi\rangle$.

After reading the right end-marker, the current state $|\psi\rangle$ is observed with respect to $E_{acc} \oplus E_{rej}$, where $E_{acc} = span\{|q\rangle : q \in Q_{acc}\}$, $E_{rej} = span\{|q\rangle : q \in Q_{rej}\}$ [1]. This observation gives $x \in E_i$ with the probability equal to the square of the projection of $|\psi\rangle$ to E_i. After that, the superposition collapses to this projection. If we get $|\psi\rangle \in E_{acc}$, the input is accepted. If $|\psi\rangle \in E_{rej}$, the input is rejected.

3.1.2 Kondacs-Watrous (measure-many) model

Independently of [CM00], quantum automata were introduced in [KW97]. The difference between these two definitions is that the measurement is performed after reading each letter (after each V_a). There are three types of states: Q_{acc} – accepting states, Q_{rej} – rejecting states and Q_{non} – nor accepting neither rejecting states.

After reading an input word letter, the current state $|\psi\rangle$ is observed with respect to $E_{acc} \oplus E_{rej} \oplus E_{non}$, where $E_{acc} = span\{|q\rangle : q \in Q_{acc}\}$, $E_{rej} = span\{|q\rangle : q \in Q_{rej}\}$ and $Q_{non} = span\{|q\rangle : q \in Q_{non}\}$. If we get $|\psi\rangle \in E_{acc}$, the input is accepted. If $|\psi\rangle \in E_{rej}$, the input is rejected. If we get $|\psi\rangle \in E_{rej}$, the computation process is continued.

It can be shown that this model is a generalization of the Moore-Crutchfield model [KW97].

3.1.3 General 1-way quantum finite automata

There exist several equivalent general quantum automata models, which were independently introduced by different authors [BMP03] [Cia01].

General quantum finite automata are similar to [CM00], but instead of unitary V_a (transformation corresponding to an input work letter a) we have general quantum operations Φ_a and state of an automaton is a mixed quantum state. For a formal definition of general 1-way QFA see [Hir11].

[1] $span(V)$ is the linear span of V, i.e. the set of all linear combinations of the elements of V.

3.2 Space-efficiency of 1-way quantum finite automata

This section overviews known results on space-efficiency of 1-way quantum finite automata (compared to 1-way classical finite automata). We start with a few definitions and then formulate the results.

Language is called *unary* is it is over alphabet consisting of a single letter. Unary language defines a function

$$f : x \to \{0, 1\}$$

which specifies if word a^x should be accepted or rejected. Language is called *periodic* if there exists k such that

$$f(x) = f(x + kn)$$

for $n \in \mathbb{N}$.

The main results on space-efficiency of 1-way QFA are:

Theorem 3.1 ([AF98]): There exists a set of unary periodic languages L_p of period p, for which any deterministic 1-way finite automaton requires at least p states, but there exists a 1-way QFA with $O(\log p)$ states.

Theorem 3.2 ([MP01]): For any periodic unary language L of size p there exists 1-way QFA with $O(\sqrt{p})$ states.

Note that quantum finite automata can not be super-exponentially more space-efficient than classical finite automata. One can always simulate a QFA (approximate its state) by a DFA with exponential number of states.

4. SPACE-EFFICIENT QUANTUM AUTOMATA

It is known that quantum finite automata can be exponentially more space-efficient than classical finite automata [AF98, Gal06]. We study a problem (defined in [AF98]) for which any classical 1-way finite automaton needs p states, but quantum 1-way finite automaton needs only $O(\log(p))$ states. Our first result is an improved exponential separation between quantum and classical finite automata. We provide a construction with less states and much simpler analysis.

Second, both constructions of QFAs (in [AF98] and this thesis) are probabilistic. They employ a sequence of parameters that are chosen at random. We present two non-probabilistic constructions of QFAs. The first of them is very simple but is supported by numerical experiments only. The second construction is more complex and has slightly larger number of states but it is provably correct.

4.1 Summary of results

Let p be a prime. We consider the language

$$L_p = \{\ a^i \mid i \text{ is divisible by } p\ \}.$$

It is easy to see that any deterministic or probabilistic 1-way finite automaton recognizing L_p has at least p states. Ambainis and Freivalds [AF98] have shown that L_p can be recognized by a QFA with $O(\log p)$ states.

The constant before $\log p$ in [AF98] depends on the required probability of correct answer. For $x \in L_p$, the answer is always correct with probability 1. For $x \notin L_p$, [AF98] give a QFA with $poly(\frac{1}{\epsilon}) \log p$ states, which is correct with probability at least $1 - \epsilon$ on inputs $x \notin L_p$.

We present a simpler construction of QFAs that achieves a better big-O constant.

Theorem 4.1: For any $\epsilon > 0$, there is a QFA with $4\frac{\log 2p}{\epsilon}$ states recognizing L_p with probability at least $1 - \epsilon$.

Similarly to [AF98] our construction is probabilistic. It employs a sequence of parameters that are chosen at random and hardwired into the QFA. We present two non-probabilistic constructions of QFAs. The first of them gives QFAs with $O(\log p)$ states but its correctness is shown by numerical experiments only. The second construction gives QFAs with $O(\log^{2+\epsilon} p)$ states but it is provably correct.

4.2 Used theorems

In the proof below we will use the following theorem from the linear algebra.

Theorem 4.2: Let $\alpha_1, \ldots, \alpha_m$ be complex numbers such that

$$|\alpha_1|^2 + \ldots + |\alpha_m|^2 = 1.$$

Then,

1. there is a unitary transformation U_1 such that $U_1|q_1\rangle = \alpha_1|q_1\rangle + \ldots + \alpha_m|q_m\rangle$.

2. there is a unitary transformation U_2 such that, for all $i \in \{1, \ldots, m\}$, $U_2|q_i\rangle$ is equal to $\alpha_i|q_1\rangle$ plus some combination of $|q_2\rangle, \ldots, |q_m\rangle$.

In the second case, we also have

$$U_2(\alpha_1|q_1\rangle + \ldots + \alpha_m|q_m\rangle) = |q_1\rangle.$$

We will also use the following theorem from the probability theory (variant of Azuma's theorem [MR94]):

Theorem 4.3: Let $X_1, \ldots, X_d$ be independent random variables such that $E[X_i] = 0$ and the value of X_i is always between -1 and 1. Then,

$$Pr[|\sum_{i=1}^{d} X_i| \geq \lambda] \leq 2e^{-\frac{\lambda^2}{2d}}.$$

4.3 Probabilistic construction

In this section we will describe the probabilistic construction of space-efficient QFA and will prove its correctness. We use QFA definition by [CM00] because it is simple and sufficient to describe our result.

Let U_k, for $k \in \{1, \ldots, p-1\}$, be a quantum automaton with a set of states $Q = \{q_0, q_1\}$, a starting state $|q_0\rangle$, $Q_{acc} = \{q_0\}$, $Q_{rej} = \{q_1\}$. The transition function is defined as follows. Transformation V_a that corresponds to symbol a maps $|q_0\rangle$ to $\cos\phi|q_0\rangle + \sin\phi|q_1\rangle$ and $|q_1\rangle$ to $-\sin\phi|q_0\rangle + \cos\phi|q_1\rangle$, where $\phi = \frac{2\pi k}{p}$ (it is easy to check that this transformation is unitary). Symbols ¢ and \$ leave $|q_0\rangle$ and $|q_1\rangle$ unchanged.

Lemma 4.1: After reading a^j, the state of U_k is

$$\cos\left(\frac{2\pi jk}{p}\right)|q_0\rangle + \sin\left(\frac{2\pi jk}{p}\right)|q_1\rangle.$$

Proof. By induction. ■

If j is divisible by p, then $\frac{2\pi jk}{p}$ is a multiple of 2π, $\cos(\frac{2\pi jk}{p}) = 1$, $\sin(\frac{2\pi jk}{p}) = 0$. Thus, reading a^j maps the starting state $|q_0\rangle$ to $|q_0\rangle$. Therefore, we get an accepting state with probability 1. This means that all automata U_k accept words in L with probability 1.

Let $k_1, \ldots, k_d$ be a sequence of $d = c \log p$ numbers. We construct an automaton U by combining $U_{k_1}, \ldots, U_{k_d}$. The set of states of U consists of $2d$ states $q_{1,0}$, $q_{1,1}$, $q_{2,0}$, $q_{2,1}$, …, $q_{d,0}$, $q_{d,1}$. The starting state is $q_{1,0}$.

The transformation for left end-marker ¢ is such that $V_{¢}(|q_{1,0}\rangle) = |\psi_0\rangle$ where

$$|\psi_0\rangle = \frac{1}{\sqrt{d}}(|q_{1,0}\rangle + |q_{2,0}\rangle + \ldots |q_{d,0}\rangle).$$

This transformation exists because of the first part of Theorem 4.2. The transformation for a is defined by

$$V_a(|q_{i,0}\rangle) = \cos\frac{2k_i\pi}{p}|q_{i,0}\rangle + \sin\frac{2k_i\pi}{p}|q_{i,1}\rangle,$$

$$V_a(|q_{i,1}\rangle) = -\sin\frac{2k_i\pi}{p}|q_{i,0}\rangle + \cos\frac{2k_i\pi}{p}|q_{i,1}\rangle.$$

The transformation for right end-marker \$ is as follows. The states $|q_{i,1}\rangle$ are left unchanged. The states $|q_{i,0}\rangle$ change to $\frac{1}{\sqrt{d}}|q_{1,0}\rangle$ plus a superposition of other

states (part 2 of Theorem 4.2, applied to $|q_{1,0}\rangle, \ldots, |q_{d,0}\rangle$). In particular,

$$V_{\$}|\psi_0\rangle = |q_{1,0}\rangle.$$

The set of accepting states Q_{acc} consists of one state $q_{1,0}$. All other states $q_{i,j}$ belong to Q_{rej}.

Claim 4.1: If the input word is a^j and j is divisible by p, then U accepts it with probability 1.

Proof. The left end-marker maps the starting state to $|\psi_0\rangle$. Reading j letters a maps each $|q_{i,0}\rangle$ to itself (see analysis of U_k). Therefore, the state $|\psi_0\rangle$, which consists of various $|q_{i,0}\rangle$, is also mapped to itself. The right end-marker maps $|\psi_0\rangle$ to $|q_{1,0}\rangle$, which is an accepting state.

■

Claim 4.2: If the input word is a^j, j not divisible by p, U accepts it with probability

$$\frac{1}{d^2}\left(\cos\frac{2\pi k_1 j}{p} + \cos\frac{2\pi k_2 j}{p} + \ldots + \cos\frac{2\pi k_d j}{p}\right)^2. \tag{4.1}$$

Proof. By Lemma 4.1, a^j maps $|q_{i,0}\rangle$ to $\cos\frac{2\pi k_i j}{p}|q_{i,0}\rangle + \sin\frac{2\pi k_i j}{p}|q_{i,1}\rangle$. Therefore, the state before reading the right end-marker \$ is

$$\frac{1}{\sqrt{d}}\sum_{i=1}^{d}(\cos\frac{2\pi k_i j}{p}|q_{i,0}\rangle + \sin\frac{2\pi k_i j}{p}|q_{i,1}\rangle).$$

The right end-marker maps each $|q_{i,0}\rangle$ to $\frac{1}{\sqrt{d}}|q_{1,0}\rangle$ plus a superposition of other basis states. Therefore, the state after reading the right end-marker \$ is

$$\frac{1}{d}\sum_{i=1}^{d}\cos\frac{2\pi k_i j}{p}|q_{1,0}\rangle$$

plus other states $|q_{i,j}\rangle$. Since $|q_{1,0}\rangle$ is the only accepting state, the probability of acceptance is the square of the coefficient of $|q_{1,0}\rangle$. This proves the lemma.

■

The rest of the proof is based on Theorem 4.3. We apply the theorem as follows. Fix $j \in \{1, \ldots, p-1\}$. Pick each of $k_1, \ldots, k_d$ randomly from

$\{0, \ldots, p-1\}$. Define $X_i = \cos \frac{2\pi k_i j}{p}$. We claim that X_i satisfies the conditions of theorem. Obviously, the value of cos function is between -1 and 1. Since $k_i = k$ for each $k \in \{0, \ldots, p-1\}$ with probability $1/p$, the expectation of X_i is

$$E[X_i] = \frac{1}{p} \sum_{k=0}^{p-1} \cos \frac{2\pi kj}{p}.$$

We have $\cos \frac{2\pi kj}{p} = \cos \frac{2\pi (kj \bmod p)}{p}$ because $\cos(2\pi + x) = \cos x$. Consider the numbers 0, j, $2j \bmod p$, ..., $(p-1)j \bmod p$. They are all distinct. Since p is prime, $kj = k'j \bmod p$ implies $k = k'$. Therefore, the numbers 0, j, $2j \bmod p$, ..., $(p-1)j \bmod p$ are just $0, 1, \ldots, p-1$ in a different order. This means that the expectation of X_i is

$$E[X_i] = \frac{1}{p} \sum_{k=0}^{p-1} \cos \frac{2\pi k}{p}.$$

This is equal to 0.

By equation (4.1), the probability of accepting a^j is $\frac{1}{d^2}(X_1 + \ldots + X_d)^2$. To achieve

$$\frac{1}{d^2}(X_1 + \ldots + X_d)^2 \leq \epsilon,$$

we need $|X_1 + \ldots + X_d| \leq \sqrt{\epsilon} d$. By Theorem 4.3, the probability that this does not happen is at most $2e^{-\frac{\epsilon d}{2}}$.

There are $p-1$ possible inputs not in L: a^1, ..., a^{p-1}. The probability that one of them gets accepted with probability more than ϵ is at most $2(p-1)e^{-\frac{\epsilon d}{2}}$. If

$$2(p-1)e^{-\frac{\epsilon d}{2}} < 1, \tag{4.2}$$

then there is at least one choice of $k_1, \ldots, k_d$ for which U does not accept any of a^1, ..., a^{p-1} with probability more than ϵ. The equation (4.2) is true if we take $d = 2\frac{\log 2p}{\epsilon}$. Therefore, the number of states for U is $4\frac{\log 2p}{\epsilon}$.

■

4.4 Explicit constructions

In the previous section we proved that for every $\epsilon > 0$ and $p \in P$ there is a QFA with $4\frac{\log 2p}{\epsilon}$ states recognizing L_p with probability at least $1 - \epsilon$. The proposed QFA construction depends on $d = 2\frac{\log 2p}{\epsilon}$ parameters $k_1, \ldots, k_d$ and accepts input word $a^j \notin L_p$ with probability

$$\frac{1}{d^2}\left(\sum_{i=1}^{d} \cos\frac{2\pi k_i j}{p}\right)^2.$$

It is possible to choose $k_1, \ldots, k_d$ values to ensure

$$\frac{1}{d^2}\left(\sum_{i=1}^{d} \cos\frac{2\pi k_i j}{p}\right)^2 < \epsilon$$

or, equivalently,

$$\left|\sum_{i=1}^{d} \cos\frac{2\pi k_i j}{p}\right| < \sqrt{\epsilon} d \tag{4.3}$$

for every $a^j \notin L_p$.

However, our proof is by a probabilistic argument and does not give an explicit sequence $k_1, \ldots, k_d$. In this section we present two constructions of explicit sequences. The first construction works well in numerical experiments and gives a QFA with $O(\log p)$ states in all tested cases. The second construction uses a slightly larger number of states but has a rigorous proof of correctness.

4.4.1 The first construction: cyclic sequences

We conjecture

Hypothesis 4.1: If g is a primitive root modulo $p \in P$, then sequence $S_g = \{k_i \equiv g^i \bmod p\}_{i=1}^{d}$ for all d and all $j : a^j \notin L_p$ satisfies (4.3).

We will call g a *sequence generator*. The corresponding sequence will be referred to as cyclic sequence. We have checked all $p \in \{2, \ldots, 9973\}$, all generators g and all sequence lengths $d < p$ (choosing a corresponding ϵ value) and have not found any counterexample to our hypothesis.

Below we describe numerical experiments which compare two strategies: a random sequence $k_1, \ldots, k_d$ and a cyclic sequence.

We will use S_{rand} to denote a random sequence and S_g to denote a cyclic sequence with generator g. We will also use ϵ_{rand} and ϵ_g to denote the maximal probability with which corresponding automata accept input word $a^j \notin L_p$.

Table 4.1 shows ϵ_{rand} and ϵ_g for different p and g values. ϵ_{rand} is calculated as the average over 5000 randomly selected sequences. ϵ_g is for one specific generator. ϵ in the second column shows the theoretical upper bound given by Theorem 4.1.

p	ϵ	d	g	ϵ_{rand}	ϵ_g
1523	0,1	161	948	0,03635	0,01517
2689	0,1	172	656	0,03767	0,01950
3671	0,1	179	2134	0,03803	0,02122
4093	0,1	181	772	0,03822	0,01803
5861	0,1	188	2190	0,03898	0,01825
6247	0,1	189	406	0,03922	0,02006
7481	0,1	193	6978	0,03932	0,01691
8581	0,1	196	5567	0,03942	0,02057
9883	0,1	198	1260	0,04011	0,01905

Tab. 4.1: ϵ_{rand} and ϵ_g for different p and g

In 99.98% - 99.99% of our experiments random sequences achieve the bound of Theorem 4.1. Surprisingly, cyclic sequences substantially outperform random ones in almost all the cases.

More precisely, for randomly selected $p \in P$, $\epsilon > 0$ and generator g, a cyclic sequence S_g gives a better result than a random sequence S_{rand} in 98.29% of cases. A few random experiment instances are shown in Figure 4.1. For each instance we show the bound $\sqrt{\epsilon}d$ of (4.3) obtained by a probabilistic argument, the maximum of $f_{rand}(j)$ (which is defined as the value of (4.3) for the sequence S_{rand}) over all j, $a^j \notin L_p$ and the maximum of $f_g(j)$ (defined in a similar way using S_g instead of S_{rand}).

In 1.81% of cases, we get that $\sup |f_g(j)| > \sup |f_{rand}(j)|$, where $\sup |f_{rand}(j)|$ is calculated as the average over 5000 randomly selected sequences. Figure 4.2 shows one of these cases: $p = 9059$, $\epsilon = 0.09$ and $g = 2689$, comparing the cyclic sequence with 9 different randomly chosen sequences. The cyclic sequence gives a slightly worse result than most of the random ones, but still beats the probabilistic bound of (4.3) by a substantial amount.

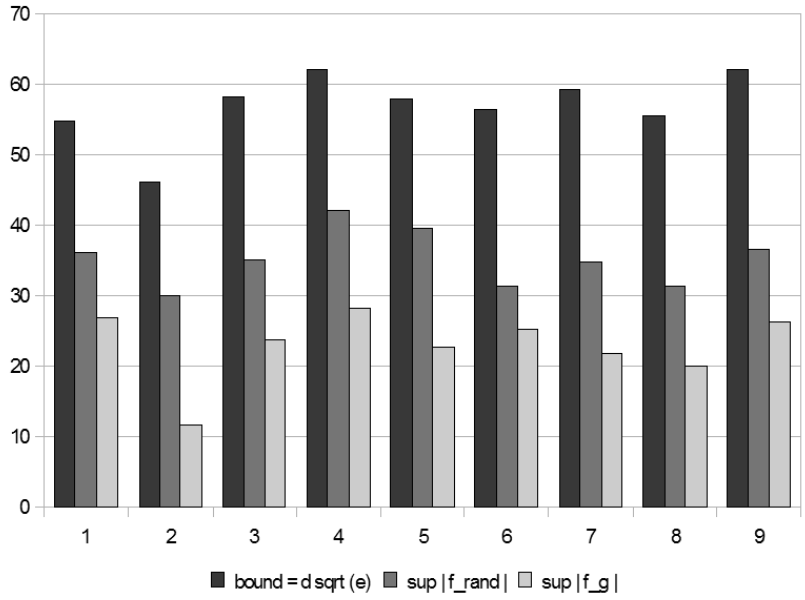

Fig. 4.1: $\sup|f_g(j)|$ and $\sup|f_{rand}(j)|$ for random p, ϵ and g

Comparing different generators

Every $p \in P$ might have multiple generators. Table 4.2 shows ϵ_g values for $p = 9059$ and $\epsilon = 0.1$ ($d = 197$, $\sqrt{\epsilon}d = 62.0101221453601$).

g	ϵ_g	g	ϵ_g	g	ϵ_g
102	0,02533	1545	0,01858	9023	0,01807
103	0,03758	1546	0,02235	9033	0,01413
105	0,01999	1549	0,02896	9034	0,01485
106	0,02852	1552	0,02873	9036	0,02509
110	0,01685	1553	0,02624	9039	0,02311

Tab. 4.2: ϵ_g values for different generators. $p = 9059$

Different generators have different ϵ_g values. We will use g_{min} to refer to a minimal generator, i.e. one having a minimal ϵ_g. Table 4.3 shows minimal generators for p values from Table 4.1.

Typically, the minimal generators give a QFA with substantially smaller probability of error. It is still an open question whether one could find a minimal generator without an exhaustive search of all generators.

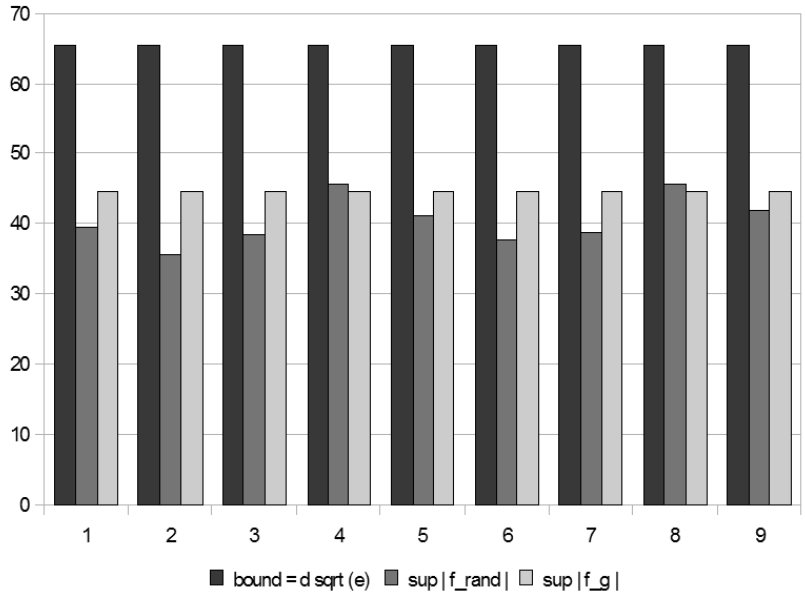

Fig. 4.2: $\sup|f_g(j)|$ and $\sup|f_{rand}(j)|$ for $p = 9059$, $\epsilon = 0.09$ and $g = 2689$

4.4.2 The second construction: AIKPS sequences

Fix $\epsilon > 0$. Let

$$R = \{r \mid r \text{ is prime, } (\log p)^{1+\epsilon}/2 < r \leq (\log p)^{1+\epsilon}\},$$
$$S = \{1, 2, \ldots, (\log p)^{1+2\epsilon}\},$$
$$T = \{s \cdot r^{-1} | r \in R, s \in S\},$$

with r^{-1} being the inverse modulo p. Ajtai et al. [AI+00] have shown

Theorem 4.4: [AI+00] For all $k \in \{1, \ldots, p-1\}$,

$$|\sum_{t \in T} e^{2tk\pi i/p}| \leq (\log p)^{-\epsilon}|T|.$$

Razborov et al. [RSW93] have shown that powers $e^{2tk\pi i/p}$ satisfy even stronger uniformity conditions. However, Theorem 4.4 is sufficient for our purposes.

By taking the real part of the left hand side, we get

$$\left|\sum_{t \in T} \cos\left(\frac{2tk\pi i}{p}\right)\right| \leq (\log p)^{-\epsilon}|T|.$$

p	ϵ	d	g	ϵ_g	g_{min}	$\epsilon_{g_{min}}$
1523	0,1	161	948	0,01517	624	0,00919
2689	0,1	172	656	0,01950	1088	0,01060
3671	0,1	179	2134	0,02122	1243	0,01121
4093	0,1	181	772	0,01803	1063	0,01154
5861	0,1	188	2190	0,01825	5732	0,01133
6247	0,1	189	406	0,02006	97	0,01182
7481	0,1	193	6978	0,01691	2865	0,01205
8581	0,1	196	5567	0,02057	4362	0,01335
9883	0,1	198	1260	0,01905	5675	0,01319

Tab. 4.3: Minimal generators for different p

Thus, the use of elements of T as $k_1, \ldots, k_d$ gives an explicit construction of a QFA with $O(\log^{2+3\epsilon})$ states.

For our first (cyclic) construction, the best provable result is a bound on exponential sums by Bourgain [Bou05]. This gives a QFA with $O(p^{c/\log\log p})$ states which is weaker than both the numerical results and the rigorous construction in this section.

4.5 Conclusions

We have considered a class of languages $L_p = \{ a^i \mid p \in \mathbb{P}, i \text{ is divisible by } p \}$ and studied exponential separation in number of states between quantum and classical finite automata. We have improved previously known result of [AF98], providing a construction of QFA with a better constant in front of $\log p$ and a much simpler analysis.

Both constructions of QFAs, presented in [AF98] and the thesis, are probabilistic. That is, they employ a sequence of parameters that are chosen at random and hardwired into the QFA. We have presented two non-probabilistic constructions of QFAs for the same class of languages. The first of them gives QFAs with $O(\log p)$ states but its correctness is only shown by numerical experiments. The second construction gives QFAs with $O(\log^{2+\epsilon} p)$ states but it is provably correct.

The language we have studied is an unary periodic language. Consider the class of unary periodic languages of period p. It is known, that some languages (such as language L_p considered above) can be recognized by QFA with logarithmic number of states, while others can not [BMP03a]. The interesting open question is to understand the necessary and sufficient properties of a language to be log-state recognizable.

Some of log-state recognizable languages, such as $\{a, a^3, a^5, \dots\}$, can be also recognized by DFA with logarithmic number of states, while others, such as [AF98], can not. It would also be interesting to understand the class of hard-for-classical-easy-for-quantum automata languages.

Part III

ANALYSIS OF GROVER'S ALGORITHM

5. QUANTUM QUERY MODEL AND GROVER'S ALGORITHM

Grover's algorithm is a quantum search algorithm solving the unstructured search problem. The algorithm is formulated within a query model where data is accessed through an oracle and query count is used as a measure of complexity of an algorithm. Grover's algorithm allows to solve the unstructured search problem in about $\frac{\pi}{4}\sqrt{N}$ queries. It is known that any deterministic or randomized algorithm requires linear time (number of queries) to solve the above problem. Thus, Grover's algorithm provides a significant speed-up over any classical algorithm.

In the following sections we overview the query model and give a description and an analysis of Grover's algorithm.

5.1 Quantum query model

Suppose we have a function

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

It is often useful to think of the function as implemented by some device, that is look at the function as a *black box*. This means that we cannot look inside the device to see how it works. The only way to gain information about the function f is to give the device some input $a \in \{0,1\}^n$ and allow the device to output $f(a) \in \{0,1\}^m$.

We will start with a simple example $f : \{0,1\} \to \{0,1\}$. In the classical case function f can be implemented by a device taking one input bit and producing one output bit (figure 5.1).

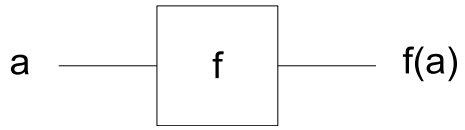

Fig. 5.1: Classical device implementing one-bit function f.

In quantum case a device needs to perform a valid quantum operation. More specifically, the action of the device must corresponds to a unitary transformation. Therefore, often it is not sufficient to consider the black box as a one-qubit operation $U_f|a\rangle \to |f(a)\rangle$. For example, if $f = 0$ (f is identically 0), then the operation would correspond to the matrix

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

which is not unitary.

To overcome the above limitation we add an auxiliary "input/output" qubit. For a one-qubit function $f : \{0,1\} \to \{0,1\}$ we define a 2-qubit operation

$$U_f|a\rangle|b\rangle \to |a\rangle|b \oplus f(a)\rangle$$

where $\oplus$ denotes the bitwise exclusive OR (figure 5.2). Usually the auxiliary

qubit is initialized to $|0\rangle$, thus, we get $|a\rangle|f(a)\rangle$ as U_f output. It can be verified that the matrix corresponding to the above operation will always be a permutation matrix, meaning that all of the entries are 0 or 1 and every row and every column has exactly one 1 in it. Permutation matrices are always unitary.

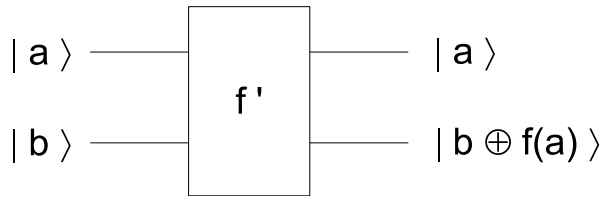

Fig. 5.2: Quantum device implementing one-bit function f with an auxilary qubit.

In the previous example (f is identically 0) the corresponding 2-bit operation will be

$$\begin{array}{lcl} U_f|0,0\rangle & \rightarrow & |0,0\rangle \\ U_f|0,1\rangle & \rightarrow & |0,1\rangle \\ U_f|1,0\rangle & \rightarrow & |1,1\rangle \\ U_f|1,1\rangle & \rightarrow & |1,0\rangle \end{array}$$

or, written in a matrix form,

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

which is unitary.

In general, for any function

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

the corresponding quantum transformation U_f will be defined by

$$U_f|x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle.$$

The associated matrix will be a permutation matrix and, therefore, will be unitary.

Speaking about quantum query model, we need to mention an interesting effect. Suppose we have a function $f : \{0,1\}^n \to \{0,1\}$ implemented as a black box U_f. We want to calculate f value on some input a. This time, however, the initial state of the auxiliary qubit will be

$$\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle.$$

Performing the U_f on the above mentioned state, we will get

$$\begin{aligned} U_f|a\rangle \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{\sqrt{2}}U_f|a\rangle|0\rangle - \frac{1}{\sqrt{2}}U_f|1\rangle|0\rangle = \\ \frac{1}{\sqrt{2}}|a\rangle|0 \oplus f(a)\rangle - \frac{1}{\sqrt{2}}|a\rangle|1 \oplus f(a)\rangle = \\ (-1)^{f(a)}|a\rangle \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right). \end{aligned}$$

We have used the fact that

$$|0 \oplus x\rangle - |1 \oplus x\rangle = (-1)^x(|0\rangle - |1\rangle)$$

for $x \in \{0,1\}$.

Notice that the U_f transformation has not changed the state of the auxiliary qubit; it has remained in the state

$$\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle.$$

At this point the auxiliary qubit can be discarded, for its state is completely independent from the state of the other qubits.

This phenomenon is usually referred to as *phase kick-back* and is a commonly used trick in quantum algorithms. Many quantum query algorithms use queries of form

$$U_f|a\rangle = (-1)^{f(a)}|a\rangle$$

with the implicitly assumed auxiliary qubit.

5.2 Grover's quantum search algorithm

Grover's algorithm is a quantum search algorithm solving the unstructured search problem. The algorithm works in the following model. We have an unstructured search space of N elements in which some elements have a certain property. We call these elements *marked*. We are given a procedure which checks whether the element is marked. The task of the algorithm is to find one of marked elements.

Grover's algorithm solves the unstructured search problem of size N in about $\frac{\pi}{4}\sqrt{N}$ queries. It is known that any deterministic or randomized algorithm requires linear number of queries to solve the above problem. Thus, Grover's algorithm provides a significant speed-up over any classical algorithm.

In this section we give a description and a brief analysis of the algorithm. The rigorous analysis of the algorithm can be found in [Gro96] or [Wat06].

Grover's algorithm

The algorithm starts in the state $|\psi_0\rangle = \frac{1}{\sqrt{N}}\sum_{x=1}^{N}|x\rangle$ (uniform superposition of all elements of the search space). Each step of the algorithm consists of two transformations: Q and D. Here Q is a query to a black box defined as

$$Q|x\rangle = (-1)^{f(x)}|x\rangle$$

and D is the an inversion about average (often called a *diffusion transformation*) defined as

$$D = 2|\psi_0\rangle\langle\psi_0| - I = \begin{bmatrix} -1+\frac{2}{n} & \frac{2}{n} & \dots & \frac{2}{n} \\ \frac{2}{n} & -1+\frac{2}{n} & \dots & \frac{2}{n} \\ \dots & \dots & \dots & \dots \\ \frac{2}{n} & \frac{2}{n} & \dots & -1+\frac{2}{n} \end{bmatrix}.$$

We refer to $|\psi_t\rangle = (DQ)^t|\psi_{start}\rangle$ as the state of Grover's algorithm after t steps.

If there is one marked element i, the probability of finding it by measuring $|\psi_t\rangle$ reaches $1 - o(1)$ for $t = O(\sqrt{N})$. If there are k marked elements, the probability of finding one of them by measuring $|\psi_t\rangle$ reaches $1 - o(1)$ for $t = O(\sqrt{N/k})$.

Analysis of Grover's algorithm

To analyse the algorithm, we define two sets:

$$A = \{x : f(x) = 1\}$$

$$B = \{x : f(x) = 0\}.$$

We will think of the set A as the set of elements that satisfy the search criterion. The set B contains all elements that do not satisfy the search criterion. The goal of the algorithm is to find one of strings from the A set.

Let $a = |A|$ and $b = |B|$. We define states

$$|A\rangle = \frac{1}{\sqrt{a}} \sum_{x \in A} |x\rangle \quad \text{and} \quad |B\rangle = \frac{1}{\sqrt{b}} \sum_{x \in B} |x\rangle,$$

which are both unit vectors and are orthogonal to each other.

The initial state of the algorithm is

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{x=1}^{N} |x\rangle = \sqrt{\frac{a}{N}}|A\rangle + \sqrt{\frac{b}{N}}|B\rangle.$$

Calculations show [Wat06] that the transformation $G = DQ$ changes states $|A\rangle$ and $|B\rangle$ as follows:

$$G|A\rangle = \left(1 - \frac{2a}{N}\right)|A\rangle - \frac{2\sqrt{ab}}{N}|B\rangle$$

$$G|B\rangle = \frac{2\sqrt{ab}}{N}|A\rangle - \left(1 - \frac{2b}{N}\right)|B\rangle.$$

As $\sqrt{\frac{a}{N}}^2 + \sqrt{\frac{b}{N}}^2 = 1$, there exists an angle θ that satisfies

$$\sin\theta = \sqrt{\frac{a}{N}} \quad \text{and} \quad \cos\theta = \sqrt{\frac{b}{N}}\ .$$

Using this notation, we can write the initial state of the register X as

$$|\psi_0\rangle = \sin\theta\,|A\rangle + \cos\theta\,|B\rangle$$

and the transformation G as

$$G|A\rangle = \cos 2\theta\,|A\rangle - \sin 2\theta\,|B\rangle$$

$$G|B\rangle = \sin 2\theta\,|A\rangle + \cos 2\theta\,|B\rangle$$

which is simply a rotation by angle 2θ in the space spanned by $|A\rangle$ and $|B\rangle$. This implies that after t iterations of G the state of the algorithm is

$$|\psi_t\rangle = \sin((2t+1)\theta)\,|A\rangle + \cos((2t+1)\theta)\,|B\rangle.$$

The goal of the algorithm is to measure some element $x \in A$, so we need the state of the algorithm to be as close to $|A\rangle$ as possible, that is

$$\sin((2t+1)\theta) \approx 1.$$

This implies

$$(2t+1)\theta \approx \frac{\pi}{2}$$

and it is sufficient to choose

$$t \approx \frac{\pi}{4\theta} - \frac{1}{2}.$$

Suppose $a = 1$. Then

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

so

$$t = \lfloor \frac{\pi}{4}\sqrt{N} \rfloor$$

is a reasonable choice for the algorithm.

In the general case the situation is more challenging. However, it can be shown that $O\left(\sqrt{\frac{N}{a}}\right)$ queries are still enough to find an $x \in A$ [Wat06].

6. OPTIMALITY OF GROVER'S ALGORITHM

Grover's quantum search algorithm is known to be optimal: no quantum algorithm can solve the unstructured search problem in less than a number of steps proportional to $\sqrt{N}$ [BB+97]. Moreover, for any number of queries up to about $\frac{\pi}{4}\sqrt{N}$, Grover's algorithm gives the maximal possible probability of finding the marked element [Zal99].

However, it is still possible to reduce the *average* number of steps required to find the marked element by ending the computation earlier and repeating the algorithm if necessary. This fact was mentioned by Christof Zalka as a short remark to the analysis of Grover's algorithm [Zal99]. Unfortunately, the remark went unnoticed by the most of scientific community. We have rediscovered this fact while analysing Grover's algorithm.

6.1 Summary of results

Theorem 6.1: Let T be a running time of Grover's algorithm. If the algorithm is stopped at moment $t \approx 0.74202\,T$ and rerun if necessary the average running time to find the marked item is $\approx 0.87857\,T$. This value is optimal.

Thus, the average number of steps can be reduced by approximately 12.14%.

6.2 Average number of steps of Grover's algorithm

Suppose we have an algorithm which gives a correct answer with some probability p. To obtain the correct answer (with probability $\Theta(1)$) we need to repeat it $\frac{1}{p}$ times on the average [MR94]. If the running time of the algorithm is t, the average running time will be $\frac{t}{p}$.

In the previous chapter we showed that the state of the Grover's algorithm after t steps is

$$|\psi_t\rangle = \sin((2t+1)\theta)\,|A\rangle + \cos((2t+1)\theta)\,|B\rangle.$$

The amplitude of the correct answer grows proportionally to $\sin(2t\theta) \approx \sin(\frac{2t}{\sqrt{N}})$, therefore, the probability to get the correct answer grows proportionally to $\sin^2(\frac{2t}{\sqrt{N}})$. To get rid of N, we scale t from $[0, \frac{\pi}{4}\sqrt{N}]$ to $[0,1]$, letting the running time of the original algorithm be 1 and t represent the fraction of steps completed by the algorithm. The probability to get the correct answer becomes $p(t) = \sin^2(\frac{\pi t}{2})$.

If we stop the computation at the moment t, the average running time of the algorithm will be

$$\frac{t}{p(t)} = \frac{t}{\sin^2\left(\frac{\pi t}{2}\right)}.$$

If $t \in [0, 0.5)$, then

$$\sin^2\left(\frac{\pi t}{2}\right) < t$$

and

$$\frac{t}{p(t)} = \frac{t}{\sin^2\left(\frac{\pi t}{2}\right)} > 1.$$

Therefore, the average running time is greater than in the original algorithm. If $t = 0.5$, then

$$\sin^2\left(\frac{\pi t}{2}\right) = 0.5$$

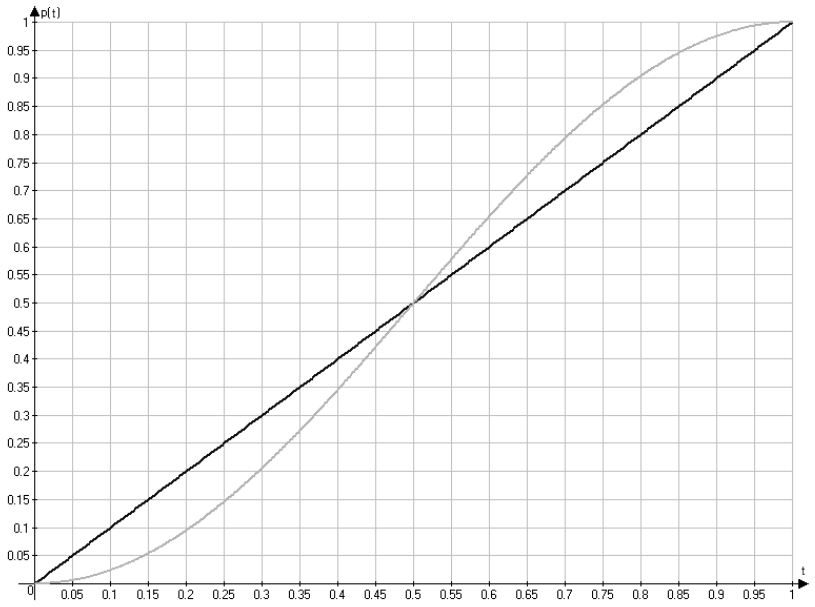

Fig. 6.1: $p(t) = \sin^2(\frac{\pi t}{2})$ and $p(t) = t$

and

$$\frac{t}{p(t)} = \frac{t}{\sin^2\left(\frac{\pi t}{2}\right)} = 1.$$

The average running time is the same as in the original algorithm. If $t \in (0.5, 1]$, then

$$\sin^2\left(\frac{\pi t}{2}\right) > t$$

and

$$\frac{t}{p(t)} = \frac{t}{\sin^2\left(\frac{\pi t}{2}\right)} < 1.$$

Therefore, the average running time is less than in the original algorithm.

The optimal moment to end the computation is the minimum of the $\frac{t}{p(t)}$ function, which can be found by solving

$$\left(\frac{t}{p(t)}\right)' = \left(\frac{t}{\sin^2\left(\frac{\pi t}{2}\right)}\right)' = \frac{\sin^2(\frac{\pi t}{2}) - t \cdot 2 \cdot \sin(\frac{\pi t}{2}) \cdot \cos(\frac{\pi t}{2}) \cdot \frac{\pi}{2}}{\sin^4(\frac{\pi t}{2})} = 0.$$

As $\sin(\frac{\pi t}{2}) \neq 0$, we have

$$\sin^2\left(\frac{\pi t}{2}\right) = 2 \cdot \sin\left(\frac{\pi t}{2}\right) \cdot \cos\left(\frac{\pi t}{2}\right) \cdot \frac{\pi t}{2}$$

or

$$\pi t = \tan\left(\frac{\pi t}{2}\right).$$

This equation has infinitely many solutions. We are interested in the one for which $t \in (0.5, 1)$. Numeric calculation gives $t \approx 0.74202$ and the average running time $\frac{t}{p(t)} \approx 0.87857$. Thus, the average number of steps can be reduced by approximately 12.14%.

6.3 Conclusions

We have shown how to reduce the average number of Grover's algorithm steps by approximately 12.14%. In case of multiple search operations this can significantly increase the performance of the algorithm.

The same argument can be used for a wide range of other quantum query algorithms, such as amplitude amplification [BH+00], some variants of quantum walks and NAND formula evaluation [AKR05, Amb07a], etc. In general, it applies to any algorithm having the same rotation-from-bad-to-good-state analysis.

7. GROVER'S ALGORITHM WITH FAULTY ORACLE : OMITTED QUERY MODEL

Grover's algorithm is a quantum search algorithm solving the unstructured search problem. The algorithm is formulated within a query model – data is accessed through an oracle and query count is used as a measure of complexity of an algorithm. Grover's algorithm solves the unstructured search problem in about $\frac{\pi}{4}\sqrt{N}$ queries, while any deterministic or randomized algorithm needs a linear number of queries. Thus, Grover's algorithm provides a significant speed-up over any classical algorithm.

The running time of the algorithm (number of queries), however, is very sensitive to errors. Regev and Schiff have shown [RS08] that if that if query transformation has some small probability of failing (reporting that none of the elements are marked), then quantum speed-up disappears: no quantum algorithm can be faster than a classical exhaustive search by more than a constant factor.

We find it interesting to understand what happens if only a constant number of failed queries is allowed. We show that even a single failed query can stop the algorithm from finding *any* of marked elements. Remarkably, this property does not depend on a number of marked elements. This makes the quantum case completely different from the classical case.

A failure of a single or multiple query transformations results in a number of steps not being executed. We show that k failed queries with a high probability change the number of actually executed steps of Grover's algorithm from l to $O\left(\frac{l}{\sqrt{k}}\right)$.

7.1 Technical preliminaries

Grover's algorithm

Suppose we have an unstructured search space of size N. Grover's algorithm finds a marked element in the search space in $O(\sqrt{N})$ steps (queries to the black-box). Each step of the algorithm consists of two transformations: D – inversion above average and Q – query transformation. Thus, the sequence of transformations of Grover's algorithm is

$$DQ\,DQ\,\ldots\,DQ = (DQ)^l.$$

Our analysis will not depend on a particular value of l and how it is related to N. We will simply treat l as the number of steps of the algorithm.

We will also use the following fact:

$$DD = QQ = I,$$

which follows from the definitions of D and Q transformations.

7.2 Model and results

Error model

In their paper [RS08], Regev and Schiff introduce the following error model: on each step, instead of the correct query Q, a faulty query Q', defined as follows, is applied:

- $Q' = I$ with probability p (error);
- $Q' = Q$ with probability $1 - p$ (no error);

[RS08] proves that in this model we need $O(N)$ steps to find any of marked elements.

We use the same definition of error (replacement of Q with I), but instead of fixing the probability of error we fix a number of errors. We assume that positions of errors are uniformly distributed independent random variables.

Summary of results

For the model above we show:

Theorem 7.1: Let l be a number of steps of the algorithm. Then $k \ll l$ uniformly distributed independent errors change the sequence of transformations of the algorithm from $(DQ)^l$ to $(DQ)^L$, where L is the random variable with expectation $O\left(\frac{l}{k}\right)$ and standard deviation $O\left(\frac{l}{\sqrt{k}}\right)$.

Therefore, with a high probability the number of actually executed steps of Grover's algorithm changes from l to $O\left(\frac{l}{\sqrt{k}}\right)$.

7.3 Related work

The work of Regev and Schiff [RS08] mentioned above is the paper that is most closely related to our work.

Several authors [LL+00, SMB03, SBW03] have studied the effect of random imperfections in either diffusion transformation or black box query on the performance of Grover's algorithm, showing that such type of noise can completely destroy the advantage of Grover's algorithm over classical exhaustive search. The difference between their work and ours is that they consider small random imperfections that occur on every step of the algorithm, while we consider the case there query is performed correctly for some marked elements and not performed at all for others.

Buhrman et al. [BN+05] have looked at a *coherent noise* model in which the algorithm has access to procedures A_i that check whether an element is marked and have some error probability. The algorithm is allowed to run both A_i and A_i^{-1} multiple times. This model is sufficiently general to enable a fault-tolerant computation and allows to simulate any noise-free quantum algorithm that makes T queries by a noisy algorithm that makes $O(T \log T)$ queries. In some cases, a constant overhead instead of a logarithmic one is sufficient. The difference between coherent noise and our models is that in coherent noise model the state after the query is still a pure state, while in our model query leads to a mixes state.

7.4 Omitting a single query

The sequence of transformations of Grover's algorithm is

$$DQ\,DQ\,\ldots\,DQ = (DQ)^l.$$

If we omit a single query transformation, the sequence changes to

$$(DQ)^{l_1} D (DQ)^{l_2},$$

where $l_1 + l_2 + 1 = l$, or

$$D(QD)^{l_1}(DQ)^{l_2}.$$

As $DD = QQ = I$, the shortest subsequence will cancel a part of the longest subsequence. More precisely

$$l_1 \geq l_2 : \quad D(QD)^{l_1}(DQ)^{l_2} = D(QD)^{l_1 - l_2}$$

$$l_1 < l_2 : \quad D(QD)^{l_1}(DQ)^{l_2} = D(DQ)^{l_2 - l_1}.$$

Thus, a single omitted query transformation changes the sequence of transformations of the algorithm from $(DQ)^l$ to $(DQ)^{O(|l_1 - l_2|)}$, decreasing the number of actually executed steps.

Suppose the query transformation can be omitted on a random step if the algorithm, that is l_1 is a uniformly distributed random variable. The length of the resulting sequence of transformations will also be a random variable. Simple calculations show that it has mean $\frac{l}{2} + O(1)$ and variance $\frac{l^2}{12} + O(l)$.

Corollary

A single omitted query transformation on the average will twice decrease the number of actually executed steps of the algorithm. If the query transformation will be omitted right in the middle of the sequence of transformations ($l_1 = l_2$), the number of actually executed steps will be 0. That is the algorithm will leave the initial state unchanged.

7.5 Omitting multiple queries

The sequence of transformations of the algorithm is

$$DQ\,DQ\,\ldots\,DQ = (DQ)^l.$$

If we omit $k-1$ query transformations, the sequence changes to

$$(DQ)^{l_1} D (DQ)^{l_2} D \ldots (DQ)^{l_{k-1}} D (DQ)^{l_k},$$

where $l_1 + l_2 + \cdots + l_k + (k-1) = l$. By regrouping the brackets we will get

$$(DQ)^{l_1} DD (QD)^{l_2} (DQ)^{l_3} DD (QD)^{l_4} \cdots =$$

$$(DQ)^{l_1} (QD)^{l_2} (DQ)^{l_3} (QD)^{l_4} \ldots$$

Transformations Q and D have the following commutativity property:

$$(QD)^i (DQ)^j = (DQ)^j (QD)^i.$$

Thus, the sequence can be rewritten as

$$(DQ)^{l_1+l_3+\ldots}\ (QD)^{l_2+l_4+\ldots}\ .$$

Therefore, k omitted query transformations change the sequence of transformations of the algorithm from $(DQ)^l$ to $(DQ)^{O(|l_1-l_2+l_3-l_4+\cdots\pm l_k|)}$.

Positions of errors, and, therefore, also $l_1, \ldots, l_k$, are random variables. Thus, the length of resulting sequence, i.e. number of actually executed steps, is also a random variable.

Next we examine the continuous approximation case, where positions of errors have continuous uniform distributions and $l_1 + l_2 + \cdots + l_k = l$. This is completely valid as $k \ll l$. We show that the length of the resulting sequence of transformations is a random variable with mean 0 (even k) or $\frac{l}{k}$ (odd k) and variance $O\left(\frac{l^2}{k}\right)$. These values perfectly agree with numerical experiment results for discrete case.

Proof of the main result

Suppose we have $k-1$ independent random variables $X_1, X_2, \ldots, X_{k-1}$. Each X_i is uniformly distributed between 0 and l. That is the probability density function of X_i is

$$f_{X_i}(x) = \begin{cases} \frac{1}{l} & x \in [0, l] \\ \\ 0 & x \notin [0, l] \end{cases}$$

and the cumulative distribution function is

$$F_{X_i}(x) = \begin{cases} 0 & x < 0 \\ \frac{x}{l} & x \in [0, l] \\ 1 & x > l \end{cases} .$$

The above random variables split the segment $[0, l]$ into k subsegments $l_1, l_2, \ldots, l_k$. The length of each subsegment is also a random variable.

Let us focus on the subsegment l_1. Probability that $l_1 \leq x$ is the probability that at least one of $X_i \leq x$. Thus, the cumulative distribution function of l_1 is

$$F_{l_1} = 1 - (1 - F_{X_1})(1 - F_{X_2}) \ldots (1 - F_{X_{k-1}})$$

or

$$F_{l_1}(x) = \begin{cases} 0 & x < 0 \\ 1 - (1 - \frac{x}{l})^{k-1} & x \in [0, l] \\ 1 & x > l \end{cases} .$$

The probability density function of l_1 is

$$f_{l_1}(x) = \begin{cases} \frac{k-1}{l}(1 - \frac{x}{l})^{k-2} & x \in [0, l] \\ 0 & x \notin [0, l] \end{cases} .$$

Knowing the probability density function of l_1, we can calculate its mean and variance by using the following formulae:

$$E[X] = \int_{-\infty}^{\infty} x \cdot f_X(x) dx$$

$$E[X^2] = \int_{-\infty}^{\infty} x^2 \cdot f_X(x) dx$$

$$Var[X] = E[X^2] - E[X]^2.$$

We leave out the details of calculation of integrals and give the results.

$$E[l_1] = \int_{-\infty}^{\infty} x \cdot f_{l_1}(x) dx = \int_0^l x \frac{k-1}{l}(1 - \frac{x}{l})^{k-2} dx = \frac{l}{k}$$

$$E[(l_1)^2] = \int_{-\infty}^{\infty} x^2 \cdot f_{l_1}(x)dx = \int_0^l x^2 \frac{k-1}{l}(1-\frac{x}{l})^{k-2}dx = \frac{2l^2}{k(k+1)}$$

$$Var[l_1] = \frac{2l^2}{k(k+1)} - \left(\frac{l}{k}\right)^2 = \frac{k-1}{k+1} \cdot \left(\frac{l}{k}\right)^2.$$

It is easy to see that all l_i subsegments have the same mean and variance. This follows from the fact that all X_i are independent and are uniformly distributed. We should also note that, although X_i are independent random variables, l_i are not independent (the length of one subsegment increases as other decreases and vice versa) .

Now let us focus on $L = l_1 - l_2 + l_3 - ... \pm l_k$. First we will calculate the mean of L. We will use the following well known formulae:

$$E[-X] = -E[X]$$

$$E[X_1 + \cdots + X_k] = E[X_1] + \cdots + E[X_k].$$

As all l_i have the same mean, then

$$E[L] = E[l_1] - E[l_2] + \cdots \pm E[l_k] = \begin{cases} 0 & k = 2m \\ \\ \frac{l}{k} & k = 2m+1 \end{cases}.$$

Now we will calculate the variance of L. As l_i subsegments are correlated, we have to use the following formula:

$$Var[X_1 + \cdots + X_k] = \sum_{i=1}^{k} Var[X_i] + \sum_{i \neq j} Cov[X_i, X_j]$$

The subsegment covariance can be easily calculated from the following trivial fact:

$$Var(l_1 + \cdots + l_k) = 0.$$

This is so because $l_1 + \cdots + l_k$ is always equal to l. Using the above formula, we will get:

$$Var[l_1 + \cdots + l_k] = \sum_{i=1}^{k} Var[l_i] + \sum_{i \neq j} Cov[l_i, l_j] = 0$$

or

$$\sum_{i=1}^{k} Var[l_i] = -\sum_{i \neq j} Cov[l_i, l_j].$$

As all l_i have the same mean and variance, they will also have the same covariances $Cov[l_i, l_j]$. Using this fact, we will get

$$k \cdot Var[l_i] = -k(k-1) \cdot Cov[l_i, l_j]$$

or

$$Cov[l_i, l_j] = -\frac{1}{k-1} \cdot Var[l_i] = -\frac{1}{k+1} \cdot \left(\frac{l}{k}\right)^2.$$

Now let us return to the variance of L:

$$Var[L] = \sum_{i=1}^{k} Var[l_i] \pm \sum_{i \neq j} Cov[l_i, l_j].$$

Covariance sign will depend on l_i and l_j signs (whether they are the same or not). More precisely:

$$Cov[-X, Y] = Cov[X, -Y] = -Cov[X, Y]$$

$$Cov[-X, -Y] = Cov[X, Y].$$

If $k = 2m$, then m subsegments have plus sign and m subsegments have minus sign. There are $2m(m-1)$ subsegment pairs with the same signs and $2m^2$ subsegment pairs with opposite signs (we should count both (l_i, l_j) and (l_j, l_i) pairs). Thus, we can rewrite the formula as:

$$Var[L] = k \cdot Var[l_i] + Cov[l_i, l_j] \cdot \big(2m(m-1) - 2m^2\big) =$$

$$= k \cdot Var[l_i] - k \cdot Cov[l_i, l_j] =$$

$$= k \cdot Var[l_i] + \frac{k}{k-1} \cdot Var[l_i] =$$

$$= k \cdot Var[l_i] \cdot \frac{k}{k-1}.$$

If $k = 2m+1$, then $m+1$ subsegments have plus sign and m subsegments have minus sign. There are $(m+1)m + m(m-1) = 2m^2$ subsegment pairs with the same signs and $2(m+1)m$ subsegment pairs with opposite signs. Thus, we can rewrite the formula as:

$$Var[L] = k \cdot Var[l_i] + Cov[l_i, l_j] \cdot \big(2m^2 - 2m(m-1)\big) =$$

$$= k \cdot Var[l_i] + (k-1) \cdot Cov[l_i, l_j] =$$
$$= k \cdot Var[l_i] - Var[l_i] =$$
$$= k \cdot Var[l_i] \cdot \frac{k-1}{k}.$$

Using O notation, we can rewrite both cases as $O(k) \cdot Var[l_i] = O\left(\frac{l^2}{k}\right)$.

■

Corollary

We have shown that $k-1$ omitted query transformations change the length of the resulting sequence of transformations from l to a random variable with mean 0 (even k) or $\frac{l}{k}$ (odd k) and variance $O\left(\frac{l^2}{k}\right)$.

From Chebyshev's inequality we have that with 96% probability L lies within five standard deviations from its mean [MR94]. For large k (but still $k \ll l$) even a tighter bound applies. In the next section we will show that the probability distribution of L for large k is close to the normal distribution. Thus, with 99.7% probability L lies within three standard deviations from the mean.

Therefore, with a very high probability the length of the resulting sequence of transformations changes from l to $O\left(\frac{l}{\sqrt{k}}\right)$. In other words k failed query transformations decrease the length of the resulting sequence of transformations $O(\sqrt{k})$ times.

7.6 Probability distribution of the median

In the previous sections we have studied the following model. We have independent random variables $X_1, X_2, \ldots, X_{k-1}$. Each X_i is uniformly distributed between 0 and l. The random variables split the segment $[0, l]$ into k subsegments $l_1, l_2, \ldots, l_k$. Our task was to estimate $L = l_1 - l_2 + l_3 - l_4 + \cdots \pm l_k$. Due to symmetry of l_i, L is equal to $\frac{l}{2} - X_m$, where X_m is the median of $X_1, X_2, \ldots, X_{k-1}$, that is the point separating the higher half of the points from the lower half of the points.

In this section we will show that for a large number of uniformly distributed random variables (points) the probability distribution of the median is close to the normal distribution.

2k + 1 points

Consider a real number interval $[-N; N]$ and $2k+1$ random points, each having a uniform distribution. Median is the point number $k + 1$.

Probability density of the median at position x, which is at the distance $|x|$ from 0, can be expressed by the formula

$$pdf(x) = \frac{(N-x)^k(N+x)^k}{(2N)^{2k+1}} \times \frac{(2k+1)!}{k!k!} = \frac{(N^2-x^2)^k(2k)!(2k+1)}{(2N)^{2k+1}k!k!}. \quad (7.1)$$

Using the Stirling approximation, we can rewrite (7.1):

$$pdf(x) \approx \frac{(N^2-x^2)^k\sqrt{4\pi k}\left(\frac{2k}{e}\right)^{2k}(2k+1)}{(2N)^{2k+1}\sqrt{2\pi k}\left(\frac{k}{e}\right)^k\sqrt{2\pi k}\left(\frac{k}{e}\right)^k} = \frac{(N^2-x^2)^k(2k+1)}{2N^{2k+1}\sqrt{\pi k}}$$

$$= \frac{\left(1-\frac{x^2}{N^2}\right)^k(2k+1)}{2N\sqrt{\pi k}}.$$

For large k we can approximate $2k + 1$ with $2k$:

$$pdf(x) \approx \frac{\left(1-\frac{x^2}{N^2}\right)^k\sqrt{k}}{N\sqrt{\pi}}. \quad (7.2)$$

For small $\frac{x}{N}$ values (7.2) can be approximated (applying $1 - z \approx e^{-z}$) by

$$pdf(x) \approx \frac{\left(e^{-\frac{x^2}{N^2}}\right)^k \sqrt{k}}{N\sqrt{\pi}} = \frac{\sqrt{k}}{N\sqrt{\pi}} e^{-k\frac{x^2}{N^2}},$$

which corresponds to the normal distribution with mean 0 and variance $\frac{N^2}{2k}$.

2k points

Consider a real number interval $[-N; N]$ and $2k$ random points, each having a uniform distribution. Median is the point number k.

Probability density of the median at position x, which is at the distance $|x|$ from 0, can be expressed by the formula

$$pdf(x) = \frac{(N-x)^{k-1}(N+x)^k}{(2N)^{2k}} \times \frac{(2k)!}{(k-1)!k!} = \frac{(N^2-x^2)^k k(2k)!}{(2N)^{2k}(N-X)k!k!}. \quad (7.3)$$

Using the Stirling approximation, we can rewrite (7.3):

$$pdf(x) \approx \frac{(N^2-x^2)^k k\sqrt{4\pi k}\left(\frac{2k}{e}\right)^{2k}}{(2N)^{2k}(N-x)\sqrt{2\pi k}\left(\frac{k}{e}\right)^k\sqrt{2\pi k}\left(\frac{k}{e}\right)^k} = \frac{(N^2-X^2)^k\sqrt{k}}{N^{2k}(N-x)\sqrt{\pi}}$$

$$= \frac{\left(1-\frac{x^2}{N^2}\right)^k\sqrt{k}}{\left(1-\frac{x}{N}\right)N\sqrt{\pi}}. \quad (7.4)$$

For small $\frac{x}{N}$ values (7.4) can be approximated (applying $1-z \approx e^{-z}$) by

$$pdf(x) \approx \frac{\left(e^{-\frac{x^2}{N^2}}\right)^k\sqrt{k}}{\left(e^{-\frac{X}{N}}\right)N\sqrt{\pi}} = \frac{\sqrt{k}}{N\sqrt{\pi}}e^{-k\frac{x^2}{N^2}+\frac{x}{N}}. \quad (7.5)$$

By multiplying (7.5) with $e^{-\frac{1}{4k}}$, which for large k is close to 1, we will get

$$pdf(x) \approx \frac{\sqrt{k}}{N\sqrt{\pi}}e^{-k\frac{x^2}{N^2}+\frac{x}{N}-\frac{1}{4k}} = \frac{\sqrt{k}}{N\sqrt{\pi}}e^{-k\frac{(x-\frac{N}{2k})^2}{N^2}},$$

which corresponds to the normal distribution with mean $\frac{N}{2k}$ and variance $\frac{N^2}{2k}$.

7.7 Conclusions

We show that k failed queries change the number of actually executed steps from l to a random variable L with expectation $O\left(\frac{l}{k}\right)$ and standard deviation $O\left(\frac{l}{\sqrt{k}}\right)$. Chebyshev's inequality guarantees that with 96% probability L lies within five standard deviations from its mean [MR94]. For large k (but still $k \ll l$) even a tighter bound applies. We show that the probability distribution of L for large k is close to the normal distribution. Thus, with 99.7% probability L lies within three standard deviations from the mean. That is with high probability the number of actually executed steps of Grover's algorithm is $O\left(\frac{l}{\sqrt{k}}\right)$. In other words k failed query transformations decrease the length of the resulting sequence of transformations $O(\sqrt{k})$ times.

However, even a single error can be very destructive. If the error occurs right in the middle of the sequence of transformations ($l_1 = l_2$), the number of actually executed steps will be 0. That is the algorithm will leave the initial state unchanged. Moreover, this behaviour is independent of number of marked elements. This makes the quantum case completely different from the classical case.

Our analysis is very generic – the same argument can be used for a wide range of other quantum query algorithms, such as amplitude amplification, some variants of quantum walks and NAND formula evaluation, etc. In general, it applies to any quantum query algorithm for which a transformation X used between queries has the property $X^2 = I$.

8. GROVER'S ALGORITHM WITH FAULTY ORACLE: INDEPENDENT ERROR MODEL

In this chapter we continue to study the behaviour of Grover's quantum search algorithm in presence of logical faults. This time, however, we use a slightly different model. Instead of omitting the query transformation, we allow it to report *some* marked elements as unmarked. Each marked element has its own probability of failing, independent of other marked elements. We assume that faults are one-sided. That is, if the i^{th} element is not marked, the black box always answers that it is not marked. If the i^{th} element is marked, the black box may give the correct answer (with probability $1-p_i$) or mistakenly answer that the element is not marked (with probability p_i).

We analyse the limiting behaviour of Grover's algorithm for a large number of steps and prove the existence of limiting state ρ_{lim}. Interestingly, the limiting state is independent of error probabilities of individual marked elements. If we measure ρ_{lim}, the probability of getting one of the marked states $i_1, \ldots, i_k$ is $\frac{k}{k+1}$. We show that convergence time is $O(n)$.

8.1 Technical preliminaries

Grover's algorithm

Suppose we have an unstructured search space of size n. Grover's algorithm starts with a state $|\psi_{start}\rangle = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle$. Each step of the algorithm consists of two transformations: Q and D. Here, Q is a query to the black box defined by

- $Q|i\rangle = -|i\rangle$ if i is a marked element;
- $Q|i\rangle = |i\rangle$ if i is not a marked element.

D is the diffusion transformation described by the following $n \times n$ matrix:

$$D = \begin{bmatrix} -1+\frac{2}{n} & \frac{2}{n} & \dots & \frac{2}{n} \\ \frac{2}{n} & -1+\frac{2}{n} & \dots & \frac{2}{n} \\ \dots & \dots & \dots & \dots \\ \frac{2}{n} & \frac{2}{n} & \dots & -1+\frac{2}{n} \end{bmatrix}.$$

We refer to $|\psi_t\rangle = (DQ)^t |\psi_{start}\rangle$ as the state of Grover's algorithm after t steps.

If there is one marked element i, the probability of finding it by measuring $|\psi_t\rangle$ reaches $1 - o(1)$ for $t = O(\sqrt{n})$. If there are k marked elements, the probability of finding one of them by measuring $|\psi_t\rangle$ reaches $1 - o(1)$ for $t = O(\sqrt{n/k})$.

Frobenius norm

Let $\rho = (\rho_{ij})$ be an $n \times n$ matrix. The *Frobenius norm* (also called *Euclidean norm* and *l_2-norm*) of ρ is defined as

$$\|\rho\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} |\rho_{ij}|^2}.$$

Frobenius norm is unitary invariant: if U is unitary, then $\|U\rho\|_F = \|\rho\|_F = \|\rho U\|_F$ [HJ06, chapter 5.6]. Also, $\|\rho\|_F \geq 0$ and $\|\rho_1 + \rho_2\| \leq \|\rho_1\| + \|\rho_2\|$, as for any matrix or vector norm.

8.2 Model and results

Error model

Suppose that a search space of size n contains k marked elements $i_1, i_2, \ldots, i_k$. In each step, instead of the correct query Q, we apply a faulty query (faulty oracle) Q' defined as follows:

- $Q'|i_j\rangle = |i_j\rangle$ with probability p_j;
- $Q'|i_j\rangle = -|i_j\rangle$ with probability $1 - p_j$;
- $Q'|i\rangle = |i\rangle$ if i is not a marked element.

For different elements i_j, faults occur independently one from another. Also, for different steps faults are independent.

Summary of results

For the model above we show

Theorem 8.1: Let ρ_t be the density matrix of the state of Grover's algorithm with a faulty oracle after t queries. Then, the sequence $\rho_1, \rho_2, \ldots$ converges to

$$\rho_{lim} = \frac{1}{k+1} \sum_{j=1}^{k} |i_j\rangle\langle i_j| + \frac{1}{k+1} |\phi\rangle\langle\phi|,$$

where $|\phi\rangle = \frac{1}{\sqrt{n-k}} \sum_{i \neq i_j} |i\rangle$ is the uniform superposition over all non-marked elements.

If we measure ρ_{lim}, the probability of getting one of the marked states $i_1, \ldots, i_k$ is $\frac{k}{k+1}$. Interestingly, the final state is independent of the error probabilities $p_1, \ldots, p_k$. Initially the probabilities of finding the elements with higher probabilities of correct answer grow faster but, in the limit for a large number of steps, the probabilities of finding all elements i_j converge to the same value $\frac{1}{k+1}$. Figure 8.1 illustrates this behaviour.

The following result quantifies the speed of convergence to the limiting state ρ_{lim}.

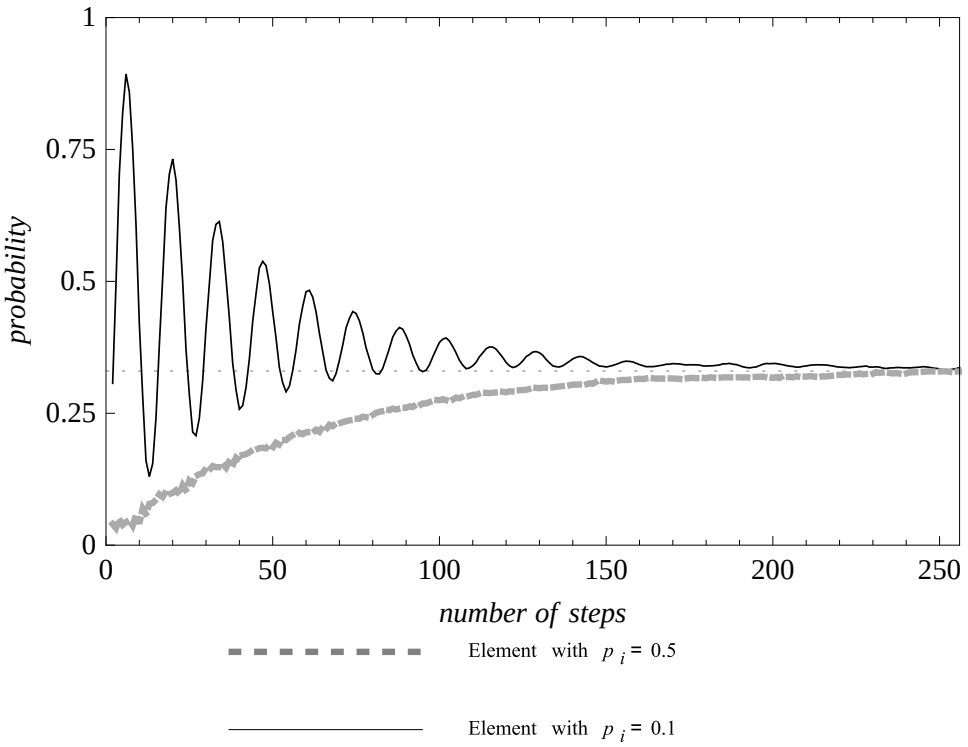

Fig. 8.1: Grover's algorithm with different error probabilities for different marked elements, $n = 1024$.

Theorem 8.2: Assume that errors occur with the same probability $p_1 = \ldots = p_k = p$ for all marked elements. Then, for every $\epsilon > 0$ there exists a number of steps of the algorithm $t = O(n)$, for which the probability to find one of the marked elements is in $[\frac{k}{k+1} - \epsilon, \frac{k}{k+1} + \epsilon]$.

8.3 Related work

The work of Regev and Schiff [RS08], mentioned in the previous chapter, is the paper that is most closely related to our research. The difference between the two approaches is that [RS08] assume that a query either outputs the correct answer for all elements (with probability $1 - p$) or answers that there is no marked element (with probability p). p_i). Whereas, we consider a model in which each marked element has its own probability of failing, independent of other marked elements. We assume that faults are one-sided. That is, if the i^{th} element is not marked, the black box always answers that it is not marked. If the i^{th} element is marked, the black box may give the correct answer (with

probability $1 - p_i$) or mistakenly answer that the element is not marked (with probability p_i).

8.4 Limiting behaviour of Grover's algorithm with errors

In this section we will study limiting behaviour of Grover's algorithm with errors and will prove the Theorem 8.1.

The state of Grover's algorithm after t steps is a pure state

$$DQ\,DQ\,\ldots\,DQ|\psi_0\rangle = (DQ)^t|\psi_0\rangle.$$

We have replaced unitary query transformation Q with stochastic faulty query transformation Q'. Thus, the state of the algorithm is no longer a pure, but a mixed state. Therefore, we should consider the density matrix ρ_t of the state of Grover's algorithm after t steps. Due to symmetry, we can assume that the first k basis states correspond to the marked elements. Note that Grover's algorithm acts in the same way on all unmarked elements. Therefore, the state of the algorithm is a probabilistic mixture of pure states of the form

$$\alpha_1|1\rangle + \ldots + \alpha_k|k\rangle + \sum_{i=k+1}^{n} \beta|i\rangle, \tag{8.1}$$

with the amplitudes of all unmarked states being equal. The density matrix ρ_t, then, takes the form

$$\rho_t = \begin{bmatrix} a_1 & b_{1,2} & \ldots & b_{1,k} & c_1 & \ldots & c_1 \\ b_{1,2} & a_2 & \ldots & b_{2,k} & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ldots & \vdots & \ddots & \vdots \\ b_{1,k} & b_{2,k} & \ldots & a_k & c_k & \ldots & c_k \\ c_1 & \ldots & \ldots & c_k & d & \ldots & d \\ \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_1 & \ldots & \ldots & c_k & d & \ldots & d \end{bmatrix}$$

because the density matrix for every pure state (8.1) in the mixture ρ_t is of this form.

Let p_i be the error probability for the i^{th} marked element. The effect of the faulty query transformation Q' on the density matrix ρ_t is:

$$\begin{array}{rcl} a_i & \mapsto & a_i \\ b_{i,j} & \mapsto & (2p_i-1)(2p_j-1)b_{i,j} \\ c_i & \mapsto & (2p_i-1)c_i \\ d & \mapsto & d \end{array}. \tag{8.2}$$

Let us prove $b_{i,j} \mapsto (2p_i - 1)(2p_j - 1)b_{i,j}$. Consider the corresponding entry $(Q'\rho_t Q')_{ij}$ of the density matrix, after the faulty oracle Q' is applied. If Q' changes the sign of either $|i\rangle$ or $|j\rangle$, the entry is equal to $-b_{ij}$. This happens with probability $p_i(1-p_j) + p_j(1-p_i)$. If Q' changes the sign of both $|i\rangle$ and $|j\rangle$ or none of them, the entry is equal to b_{ij}. This happens with probability $p_i p_j + (1-p_i)(1-p_j)$. Hence,

$$(Q'\rho_t Q')_{ij} = -b_{ij}(p_i(1-p_j) + p_j(1-p_i)) + b_{ij}(p_i p_j + (1-p_i)(1-p_j)) =$$

$$= (1-2p_i)(1-2p_j)b_{ij}.$$

Similarly, we can prove that $c_i \mapsto (2p_i - 1)c_i$, $a_i \mapsto a_i$ and $d \mapsto d$.

Consider the Frobenius norm of the density matrix. If we multiply the density matrix by the unitary diffusion matrix, its Frobenius norm does not change. Since the faulty query transformation decreases the Frobenius norm (if $0 < p_i < 1$) and the Frobenius norm takes non-negative values, the $\lim_{t\to\infty} \|\rho_t\| = C$ exists.

If $\lim_{t\to\infty} b_{i,j} \neq 0$, we obtain a contradiction, because the Frobenius norm decreases infinitely. Analogously, we can prove $\lim_{t\to\infty} c_i = 0$.

Let us prove $\lim_{t\to\infty}(a_i - a_j) = 0$ for each $i \neq j$. Assume it is not true, i.e. there exist $i \neq j$ and $\delta > 0$ so that $|a_i - a_j| > \delta$ for infinitely many t. Consider t' so that for all $t > t'$ and all m, l inequalities $b_{m,l} < \epsilon$ and $c_m < \epsilon$ hold. After right multiplying the density matrix by the diffusion matrix

$$\rho_t D = \begin{bmatrix} a_1 & \dots & O(\epsilon) & O(\epsilon) & \dots & O(\epsilon) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ O(\epsilon) & \dots & a_k & O(\epsilon) & \dots & O(\epsilon) \\ O(\epsilon) & \dots & O(\epsilon) & d & \dots & d \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ O(\epsilon) & \dots & O(\epsilon) & d & \dots & d \end{bmatrix} \begin{bmatrix} -1+\frac{2}{n} & \frac{2}{n} & \dots & \frac{2}{n} \\ \frac{2}{n} & -1+\frac{2}{n} & \dots & \frac{2}{n} \\ \dots & \dots & \dots & \dots \\ \frac{2}{n} & \frac{2}{n} & \dots & -1+\frac{2}{n} \end{bmatrix},$$

the last column contains values $\frac{2a_1}{n} + O(\epsilon)$, $\dots$, $\frac{2a_k}{n} + O(\epsilon)$ and $\frac{d(n-2k)}{n} + O(\epsilon)$ ($n-k$ times). After left multiplying this matrix by the diffusion matrix, each of the first k elements in the last column takes the value $2v - \frac{2a_i}{n} + O(\epsilon)$, where v is the arithmetic mean of the last column of $\rho_t D$. We obtain a contradiction by choosing a sufficiently small ϵ, because at least two of these values differ by at least $\frac{2\delta}{n} + O(\epsilon)$.

For an arbitrary ϵ we can choose t' so that for every $t > t'$ the inequalities $b_{m,l} < \epsilon$, $c_m < \epsilon$ and $|a_m - a_l| < \epsilon$ hold for all m and l. Since $a_1 + \ldots + a_k + d(n-$

$k) = 1$ (a property of the density matrix), it follows that $a_i = \frac{1-d(n-k)}{k} + O(\epsilon)$. So, the arithmetic mean of the last column of $\rho_t D$ is

$$v = \frac{2(a_1 + \ldots + a_k) + d(n-2k)(n-k)}{n^2} + O(\epsilon) =$$

$$= \frac{2 + d(n-2k-2)(n-k)}{n^2} + O(\epsilon).$$

After left and right multiplying the density matrix by the diffusion matrix, the i-th value in the last column is

$$2v - \frac{2a_i}{n} + O(\epsilon) = 2v - \frac{2 - 2d(n-k)}{nk} + O(\epsilon) =$$

$$= \frac{4 + 2d(n-2k-2)(n-k)}{n^2} - \frac{2 - 2d(n-k)}{nk} + O(\epsilon) =$$

$$= \frac{2(n-2k)(d(k+1)(n-k) - 1)}{kn^2} + O(\epsilon).$$

Since this sum must be $O(\epsilon)$, it follows that $d(k+1)(n-k) - 1 = O(\epsilon)$, assuming $n \neq 2k$. Choosing ϵ arbitrarily small, we obtain $\lim_{t\to\infty} d = \frac{1}{(k+1)(n-k)}$ and $\lim_{t\to\infty} a_i = \frac{1}{k+1}$.

■

8.5 Convergence speed of Grover's algorithm with errors

In this section we will study how fast Grover's algorithm with errors converges to its limiting state and will prove the Theorem 8.2.

We describe the state of Grover's algorithm after t queries by the density matrix

$$\rho_t = \begin{bmatrix} a_1 & b_{1,2} & \dots & b_{1,k} & c_1 & \dots & c_1 \\ b_{1,2} & a_2 & \dots & b_{2,k} & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \dots & \vdots & \ddots & \vdots \\ b_{1,k} & b_{2,k} & \dots & a_k & c_k & \dots & c_k \\ c_1 & \dots & \dots & c_k & d & \dots & d \\ \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_1 & \dots & \dots & c_k & d & \dots & d \end{bmatrix}.$$

In this section we assume that errors occur with the same probability $p_1 = \ldots = p_k = p$ for all marked elements. Thus, the density matrix takes the much simpler form

$$\rho_t = \begin{bmatrix} a & b & \dots & b & c & \dots & c \\ b & a & \dots & b & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ b & b & \dots & a & c & \dots & c \\ c & \dots & \dots & c & d & \dots & d \\ \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c & \dots & \dots & c & d & \dots & d \end{bmatrix}.$$

In the further analysis we use the square of the Frobenius norm of the density matrix:

$$\|\rho\|_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} |\rho_{ij}|^2.$$

We will also need the function

$$S(\rho) = k(k-1)b^2 + 2k(n-k)c^2, \tag{8.3}$$

which gives the sum of squares of all b and c elements of the density matrix.

According to (8.2), the faulty query transformation Q' decreases the square of the Frobenius norm of the density matrix by

$$k(k-1)b^2 + 2k(n-k)c^2 - k(k-1)(b(2p-1)^2)^2 - 2k(n-k)(c(2p-1))^2 =$$

$$= k(k-1)b^2(1-(2p-1)^4) + 2k(n-k)c^2(1-(2p-1)^2) >$$
$$> (k(k-1)b^2 + 2k(n-k)c^2)(1-(2p-1)^2) = S(\rho)(4p-4p^2). \qquad (8.4)$$

Before the first application of the query transformation, the Frobenius norm is 1. Each further application of the query transformation decreases the Frobenius norm. We have proved that the Frobenius norm has a limit of $\frac{1}{\sqrt{k+1}}$ (Frobenius norm of the limiting state ρ_{lim}). Thus, total decrease of the Frobenius norm is $1 - \frac{1}{\sqrt{k+1}}$. Similarly, the square of the Frobenius norm decreases from 1 to $\frac{1}{k+1}$ and has the total decrease of $\frac{k}{k+1}$.

Among first $2m$ applications of the query transformation, there exist two sequential applications which decrease the square of the Frobenius norm by less than $\frac{1}{m}$. Let ρ_1 and ρ_2 be density matrices before these applications. Let a_1, b_1, c_1, d_1 and a_2, b_2, c_2, d_2 be a, b, c, d values of ρ_1 and ρ_2 respectively.

From (8.4) we have

$$S(\rho_1) < \frac{1}{m(4p-4p^2)} \quad \text{and} \quad S(\rho_2) < \frac{1}{m(4p-4p^2)}. \qquad (8.5)$$

In the further proof we use the following straightforward-to-prove lemma:

Lemma 8.1: If $S = k(k-1)b^2 + 2k(n-k)c^2 < R$ and $k \geq 2$ hold then $|c| < \sqrt{\frac{R}{n}}$ and $|b| < \sqrt{R}$ also hold.

We also use the notation $\delta(a,b) = \{x | a - b < x < a + b\}$.

Lemma 8.1 and the equation (8.5) implies

$$c_1 \in \delta\left(0, \sqrt{\frac{R}{n}}\right),$$
$$b_1 \in \delta\left(0, \sqrt{R}\right),$$
$$c_2 \in \delta\left(0, \sqrt{\frac{R}{n}}\right),$$
$$b_2 \in \delta\left(0, \sqrt{R}\right),$$

where $R = \frac{1}{m(4p-4p^2)}$.

The diffusion matrix changes each element a of a vector to $2v - a$, where v is the arithmetic mean of all elements. We will call this the diffusion matrix property.

The arithmetic mean of each of the first k columns of the matrix ρ_1' (after the first application of the query transformation) is

$$v \in \delta\left(\frac{a_1}{n}, \sqrt{R}\frac{k-1}{n} + \sqrt{\frac{R}{n}}\frac{n-k}{n}\right) \subseteq \delta\left(\frac{a_1}{n}, \frac{k}{n}\sqrt{R} + \sqrt{\frac{R}{n}}\right).$$

Because of the diffusion matrix property, the value of the last elements of the first k columns of the matrix $D\rho_1'$ is

$$c_1' = 2v - c_1 \in \delta\left(2\frac{a_1}{n}, \frac{2k}{n}\sqrt{R} + 3\sqrt{\frac{R}{n}}\right).$$

The arithmetic mean of each of the last $n-k$ columns of the matrix ρ_1' is

$$v \in \delta\left(d_1\frac{n-k}{n}, \frac{k}{n}\sqrt{\frac{R}{n}}\right).$$

Hence, the value of the last elements of the last $n-k$ columns of the matrix $D\rho_1'$ is

$$d_1' = 2v - d_1 \in \delta\left(d_1\frac{n-2k}{n}, \frac{2k}{n}\sqrt{\frac{R}{n}}\right).$$

The arithmetic mean of the last row of the matrix $D\rho_1'$ is

$$v \in \delta\left(a_1\frac{2k}{n^2} + d_1\frac{(n-k)(n-2k)}{n^2}, \frac{2k^2}{n^2}\sqrt{R} + \frac{5nk-2k^2}{n^2}\sqrt{\frac{R}{n}}\right).$$

Assuming $n > 2k$ and using the definition of the diffusion matrix, we obtain

$$c_2 = 2v - c_1' \in$$

$$\in \delta\left(-2a_1\frac{n-2k}{n^2} + 2d_1\frac{(n-k)(n-2k)}{n^2}, \frac{4k^2}{n^2}\sqrt{R} + \frac{10nk-4k^2}{n^2}\sqrt{\frac{R}{n}} + \frac{2k}{n}\sqrt{R} + 3\sqrt{\frac{R}{n}}\right) \subseteq$$

$$\subseteq \delta\left(-2a_1\frac{n-2k}{n^2} + 2d_1\frac{(n-k)(n-2k)}{n^2}, \frac{4k}{n}\sqrt{R} + 13\sqrt{\frac{R}{n}}\right) =$$

$$= \delta \left(\frac{2(d_1(n-k)-a_1)(n-2k)}{n^2}, \frac{4k}{n}\sqrt{R} + 13\sqrt{\frac{R}{n}} \right).$$

As $c_2 \in \delta\left(0, \sqrt{\frac{R}{n}}\right)$, $\left|\frac{2(d_1(n-k)-a_1)(n-2k)}{n^2}\right| < \frac{4k}{n}\sqrt{R} + 14\sqrt{\frac{R}{n}}$ holds.

As $ka_1 + d_1(n-k) = 1$, it follows that $d_1(n-k) - a_1 = 1 - (k+1)a_1$. Using the inequality

$$\left| \frac{k}{k+1} - ka_1 \right| < |1 - (k+1)a_1|,$$

we obtain

$$\left| \frac{k}{k+1} - ka_1 \right| < \left(\frac{2k}{n} + \frac{7}{\sqrt{n}} \right) \frac{n^2}{n-2k}\sqrt{R}.$$

The left side of this inequality is the absolute value of the difference between the probability of finding any of the marked elements and $\frac{k}{k+1}$.

For an arbitrary ϵ the inequality

$$\left(\frac{2k}{n} + \frac{7}{\sqrt{n}} \right) \frac{n^2}{n-2k}\sqrt{R} < \epsilon$$

holds if

$$m > \frac{1}{4p(1-p)\epsilon^2} \left(\frac{2k}{n} + \frac{7}{\sqrt{n}} \right)^2 \frac{n^4}{(n-2k)^2} = O(n)$$

(substituting $R = \frac{1}{4mp(1-p)}$).

■

8.6 Conclusions

We have analysed the behaviour of Grover's algorithm in the model of logical errors where the query transformation is allowed to report some marked elements as unmarked. We have shown existence of the limiting state ρ_{lim} to which the state of the algorithm converges after the large number of steps. If we measure ρ_{lim}, the probability of getting one of the marked states $i_1, \ldots, i_k$ is $\frac{k}{k+1}$. We have analysed the speed of convergence to the limiting state and shown that this happens in $O(n)$ steps. This matches the lower bound of [RS08] [1].

Our analysis uses the density matrix formalism, which is the standard tool for analysing the effect of stochastic operations on a quantum state. Although, our results can not be directly applied to other query algorithms (as different algorithms have different transformations applied between subsequent queries), our approach (structure of the proof and used techniques) can be adapted to analyse the behaviour of query algorithms in the described and similar error models.

For example, applying our approach to the error model of [RS08] (described in chapter 7) we can prove existing of limiting state

$$\rho_{lim} = \frac{1}{2k} \sum_{j=1}^{k} |i_j\rangle\langle i_j| + \frac{1}{2(N-k)} |\phi\rangle\langle\phi|$$

and $O(N)$ convergence time. The corresponding proofs are just a minor modification of proofs of theorems 8.1 and 8.2.

Our proofs provide useful insights into fault-tolerance of quantum query algorithms. For example, zero limits of coefficients of the density matrix affected by a faulty query (in our case $b_{i,j}$ and c_i coefficients) hold for *any* quantum query algorithm, i.e. does not depend on transformation used between subsequent queries.

It would be interesting to generalize our results to a wider class of quantum query algorithms (and, if possible, other models of errors) and understand limits of our approach. It would also be interesting to connect our results with results from quantum Markov chain theory, which studies limiting states of a quantum system which undergoes a sequence of stochastic quantum operations [Gud08, LP11].

[1] Technically, the lower bound of [RS08] is for a slightly different model. However, the difference between the models is not important in this case.

Part IV

QUANTUM WALKS

9. SEARCH BY QUANTUM WALKS ON TWO-DIMENSIONAL GRID

Quantum walks are quantum counterparts of random walks [Amb03, Kem03]. They have been useful to design quantum algorithms for a variety of problems [CC+03, Amb07, Sze04, AKR05, MSS05, BS06]. In many of those applications, quantum walks are used as a tool for search.

We study a search by quantum walks on a finite two-dimensional grid according to [AKR05]. For grid of size $\sqrt{N} \times \sqrt{N}$ the original [AKR05] algorithm takes $O(\sqrt{N \log N})$ steps and finds a marked location with probability $O(1/\log N)$. This probability is small, thus, the algorithm needs amplitude amplification to get $\Theta(1)$ probability. The amplitude amplification adds an additional $O(\sqrt{\log N})$ factor to the number of steps, making it $O(\sqrt{N} \log N)$.

We show that despite small probability to find marked location, the probability to be within $O(\sqrt{N})$ neighbourhood, i.e. at $O(\sqrt[4]{N})$ distance from the marked location, is $\Theta(1)$. This allows us to replace amplitude amplification with classical post processing which does not increase time complexity of the algorithm and leads to $O(\sqrt{\log N})$ speed-up.

The same speed-up has been already achieved by other research groups. However, their approaches to this problem are based on modification of the original algorithm [Tul08] or both the algorithm and the structure of the graph [KM+10]. Therefore, we find our approach as deserving an interest.

9.1 [AKR05] quantum walk search algorithm

This section describes a quantum walk model of [AKR05] for two-dimensional grid. The model, however, is very generic and can be used for other types of graphs.

Search problem

Suppose we have N items arranged on a two dimensional lattice of size $\sqrt{N} \times \sqrt{N}$. The locations on the lattice are labelled by their x and y coordinate as (x, y) for $x, y \in \{0, \ldots, \sqrt{N} - 1\}$. We assume that the grid has periodic boundary conditions. For example, going right from a location $(\sqrt{N} - 1, y)$ on the right edge of the grid leads to the location $(0, y)$ on the left edge of the grid. Similarly to Grover's algorithm some of locations have a certain property. We call these locations marked. We are given a procedure which checks whether the location is marked. The algorithm is allowed to check its current location or to move to an adjacent location. The task of the algorithm is to find one of marked locations.

[AKR05] algorithm

To introduce quantum version of random walk, we define a "location" register with basis states $|i, j\rangle$, $i, j \in \{0, \ldots, \sqrt{N} - 1\}$. We also define an additional "coin" register with four states, one for each direction: $|\Uparrow\rangle$, $|\Downarrow\rangle$, $|\Leftarrow\rangle$ and $|\Rightarrow\rangle$ [1]. Thus, basis states of quantum walk are $|i, j, d\rangle$ for $i, j \in \{0, \ldots, \sqrt{N} - 1\}$, $d \in \{\Uparrow, \Downarrow, \Leftarrow, \Rightarrow\}$ and the state of quantum walk is given by:

$$|\psi(t)\rangle = \sum_{i,j} (\alpha_{i,j,\Uparrow}|i, j, \Uparrow\rangle + \alpha_{i,j,\Downarrow}|i, j, \Downarrow\rangle + \alpha_{i,j,\Leftarrow}|i, j, \Leftarrow\rangle + \alpha_{i,j,\Rightarrow}|i, j, \Rightarrow\rangle).$$

The [AKR05] quantum walk algorithm starts in the state

$$|\psi(0)\rangle = \frac{1}{2\sqrt{N}} \sum_{i,j} \big(|i, j, \Uparrow\rangle + |i, j, \Downarrow\rangle + |i, j, \Leftarrow\rangle + |i, j, \Rightarrow\rangle\big).$$

Each step of the algorithm consists of three transformations: Q, C and S. Here, Q is a query to the black box defined by

[1] There are also quantum walk models which does not have coin register (e.g. [Sze04]).

- $Q|i,j,d\rangle = -|i,j,d\rangle$ if location (i,j) is marked;
- $Q|i,j,d\rangle = |i,j,d\rangle$ if location (i,j) is not marked.

C is the transform on the coin register, called *coin flip transformation*. The [AKR05] algorithm uses Grover's diffusion transformation

$$D = \frac{1}{2}\begin{pmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{pmatrix}$$

as C. The transformation S is called *shift transformation* and is defined as:

$$\begin{array}{lcl} |i,j,\Uparrow\rangle & \rightarrow & |i,j-1,\Downarrow\rangle \\ |i,j,\Downarrow\rangle & \rightarrow & |i,j+1,\Uparrow\rangle \\ |i,j,\Leftarrow\rangle & \rightarrow & |i-1,j,\Rightarrow\rangle \\ |i,j,\Rightarrow\rangle & \rightarrow & |i+1,j,\Leftarrow\rangle \end{array}.$$

Notice that after moving to an adjacent location S changes the value of the direction register to the opposite. This is necessary for the quantum walk algorithm of [AKR05] to work. The state of the algorithm after t steps is refered as $|\psi(t)\rangle$.

If there are marked locations, the state of the algorithm starts to deviate from $|\psi(0)\rangle$. It has been shown [AKR05] that after $O(\sqrt{N \log N})$ steps the inner product $\langle\psi(t)|\psi(0)\rangle$ becomes close to 0. If the state of the algorithm is measured at this moment then for one or two marked locations we find a marked location with $O(1/\log N)$ probability. For multiple marked locations this is not always the case. There exist marked location configurations for which quantum walk fails to find any of marked locations [AR08].

The probability to find a marked location is small, thus, the algorithm uses amplitude amplification [BH+00] to get $\Theta(1)$ probability. The amplitude amplification adds an additional $O(\sqrt{\log N})$ factor to the number of steps, making it $O(\sqrt{N}\log N)$.

9.2 Summary of results

Suppose we have an $\sqrt{N} \times \sqrt{N}$ grid with one marked location [2]. The [AKR05] algorithm takes $O(\sqrt{N \log N})$ steps and finds the marked location with $O(1/\log N)$ probability. The algorithm then uses amplitude amplification to get $\Theta(1)$ probability. The amplitude amplification adds an additional $O(\sqrt{\log N})$ factor to the number of steps, making it $O(\sqrt{N} \log N)$.

Performing numerical experiments with [AKR05] algorithm, we have noticed that probability to be close to the marked location is much higher than probability to be far from the marked location. Figure 9.1 shows probability distribution by distance from the marked location for 1024×1024 grid. Figure 9.2 shows the same probability distribution on logarithmic scale.

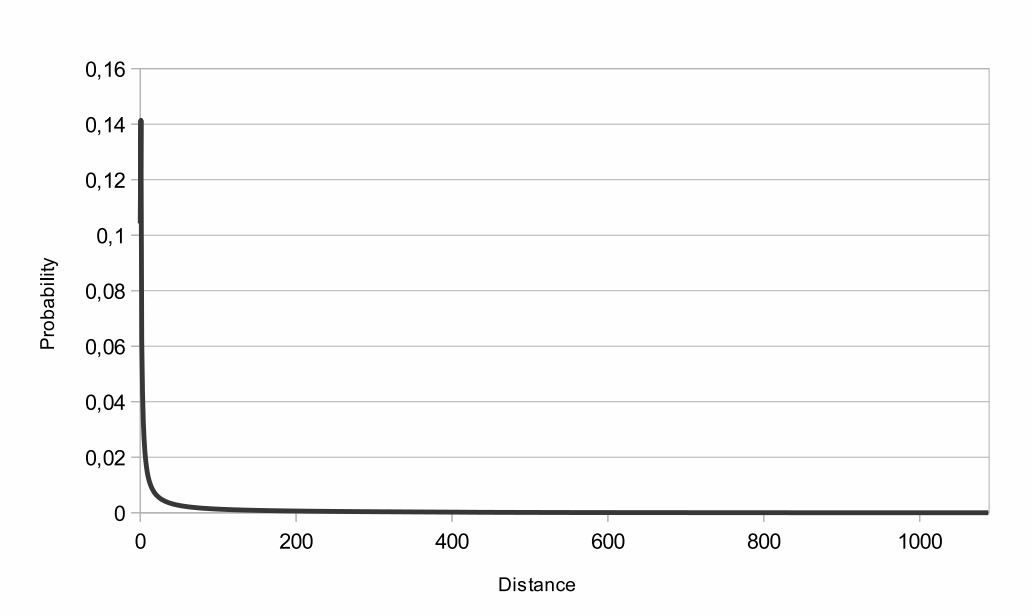

Fig. 9.1: Probability by distance, one marked location, grid size 1024×1024, normal scale.

In this chapter we show:

Theorem 9.1: We can choose $t = O(\sqrt{N \log N})$ so that, if we run [AKR05] algorithm with one marked location (i, j) for t steps and perform the measurement, the probability of obtaining a location (i', j') with $|i - i'| \leq N^{\epsilon}$ and $|j - j'| \leq N^{\epsilon}$ is $\Omega(\epsilon)$.

[2] Numerical experiments give very similar results for multiple marked locations.

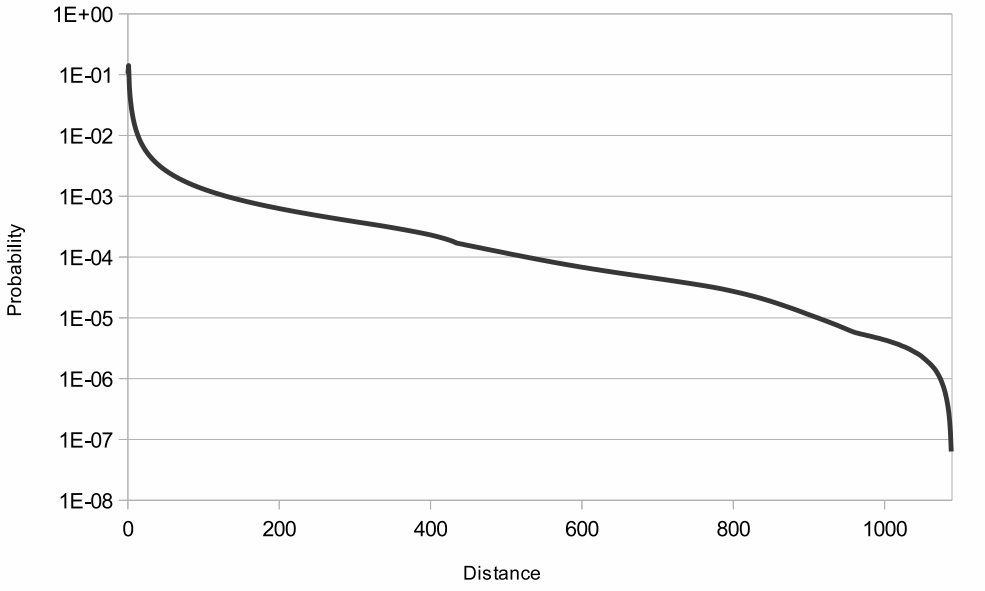

Fig. 9.2: Probability by distance, one marked location, grid size 1024×1024, logarithmic scale.

The theorem allows us to replace amplitude amplification with a classical post processing step. We run the algorithm for $O(\sqrt{N \log N})$ steps and perform a measurement. Then we classically check $O(\sqrt{N})$ neighbourhood of the outcome of the measurement. According to the theorem the probability to find the marked location is $\Omega(1/2)$ (figure 9.3 shows this probability for different grid sizes). Thus, we do not need to perform the amplitude amplification and, therefore, the running time of the algorithm stays $O(\sqrt{N \log N})$.

9.3 Related work

The problem of search on a two-dimensional grid was stated in 2002 by Paul Benioff [Ben02], who conjectured that search on two-dimensional $\sqrt{N} \times \sqrt{N}$ grid needs $\Omega(N)$ time, i.e. no quantum speed-up is possible in this setting. One year later Ambainis and Aaronson proposed an algorithm [AA03] which finds a marked location in $O(\sqrt{N} \log^2 N)$ steps. In 2005 Ambainis, Kempe and Rivosh [AKR05] proposed a quantum walk based algorithm. The [AKR05] algorithm requires $O(\sqrt{N} \log N)$ steps. The basic building block consists in $O(\sqrt{N \log N})$ steps of a quantum walk, which succeeds in finding the marked

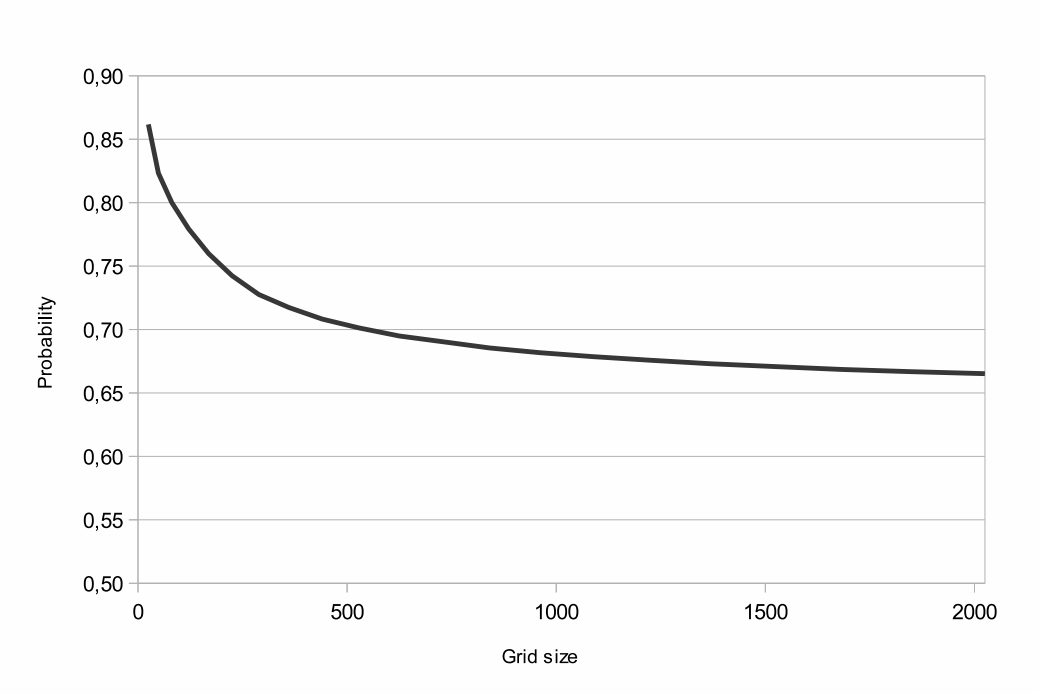

Fig. 9.3: Probability to be within $\sqrt{N}$ neibourghood from the marked location.

location with $O(1/\log N)$ probability. This success probability can in turn be amplified to $\Theta(1)$ by using amplitude amplification, which adds an additional $O(\sqrt{\log N})$ factor to the number of steps.

Following the [AKR05] algorithm, it had been conjectured that this cost could be reduced to $O(\sqrt{N \log N})$, hence providing a full quadratic speed-up over the corresponding random walk based approach. This conjecture has been confirmed a few years later, first by Tulsi, who showed in 2008 how the basic quantum walk could be modified so that a constant success probability could be achieved in $O(\sqrt{N \log N})$ steps [Tul08]. Tulsi's technique has later been extended by Magniez et al., who showed in 2009 that a full quadratic speed-up could be obtained over any state-transitive random walk [MN+09]. In 2010, Krovi et al. gave another technique leading to a similar result, but for an extended class of random walks, namely any reversible random walk [KM+10].

We propose a third technique to reduce the cost to $O(\sqrt{N \log N})$ for the search on the two dimensional grid. Our idea is that while the basic quantum walk can only find the marked location with probability $O(1/\log N)$, it actually returns a location close to the marked one with high probability. Therefore, the marked location can in turn be found by running a classical search over

the neighbourhood of the location returned by the quantum walk.

We find our result (localization of probability around marked elements) interesting and hope to extend it to other types of graphs.

9.4 Proofs

The proof of Theorem 9.1 consists of two steps. First, in Lemma 9.1, we derive an approximation for the state of quantum walk at time $t = O(\sqrt{N \log N})$ when the state of quantum walk has the biggest difference from the starting state. Then, in section 9.4.2, we use this approximation to derive our main result.

Informally, the idea of the proof is as follows. Denote $Pr[0]$ the probability to find a marked location and $Pr[R]$ the probability to be at distance R from the marked location (we use Manhattan or L_1 distance). For small R values ($R \ll \sqrt{N}$) we have:

$$Pr[R] \approx \frac{Pr[0]}{R^2}.$$

There are $4R$ points at the distance R from the marked location. Thus, the total probability to be within $\sqrt{N}$ neighbourhood of the marked location is:

$$Pr[\leq \sqrt[4]{N}] = \sum_{R=1}^{\sqrt[4]{N}} 4R \times O\left(\frac{Pr[0]}{R^2}\right) = Pr[0] \times \sum_{R=1}^{\sqrt[4]{N}} O\left(\frac{1}{R}\right) = Pr[0] \times O(\log N).$$

The probability to find the marked location is $O(1/\log N)$, thus, we have

$$Pr[\leq \sqrt[4]{N}] = O\left(\frac{1}{\log N}\right) \times O(\log N) = const.$$

9.4.1 Approximation of the final state of the quantum walk

Let

$$|\psi\rangle = \sum_{j=0}^{\sqrt{N}-1} \sum_{j'=0}^{\sqrt{N}-1} \sum_{d} \alpha_{j,j',d}^{t} |j, j', d\rangle$$

be the state of the quantum walk after t steps.

Lemma 9.1: We can choose $t = O(\sqrt{N \log N})$ so that for any set

$$S \subseteq \{0, ..., \sqrt{N} - 1\}^2$$

we have

$$\sum_{(j,j')\in S} |\alpha^t_{j,j',\Uparrow}|^2 \geq C^2 \sum_{(j,j')\in S} (f(j,j') - f(j-1,j'))^2 + o(1),$$

where

$$f(j,j') = \sum_{(k,l)\neq(0,0)} \frac{1}{2 - \cos\frac{2k\pi}{\sqrt{N}} - \cos\frac{2l\pi}{\sqrt{N}}} \omega^{kj+lj'},$$

$\omega = e^{\frac{2\pi i}{\sqrt{N}}}$ and $C = \Theta(\frac{1}{\sqrt{N\log N}})$.

Proof: We will repeatedly use the following lemma.

Lemma 9.2: [BV97] Let $|\psi\rangle = \sum_{i=1}^m \alpha_i|i\rangle$ and $|\psi'\rangle = \sum_{i=1}^m \beta_i|i\rangle$. Then, for any set $S \subseteq \{1,2,\ldots,m\}$,

$$\sum_{i\in S} \left||\alpha_i|^2 - |\beta_i|^2\right| \leq 2\|\psi - \psi'\|.$$

We recast the algorithm for search on the grid as an instance of an *abstract search algorithm* (generalization of Grover's search algorithm) [AKR05]. An abstract search algorithm consists of two unitary transformations U_1 and U_2 and two states $|\psi_{start}\rangle$ and $|\psi_{good}\rangle$. We require the following properties:

1. $U_1 = I - 2|\psi_{good}\rangle\langle\psi_{good}|$. In other words, $U_1|\psi_{good}\rangle = -|\psi_{good}\rangle$ and, if $|\psi\rangle$ is orthogonal to $|\psi_{good}\rangle$, then $U_1|\psi\rangle = |\psi\rangle$;
2. $U_2|\psi_{start}\rangle = |\psi_{start}\rangle$ for some state $|\psi_{start}\rangle$ with real amplitudes and there is no other eigenvector with eigenvalue 1;
3. U_2 is described by a real unitary matrix.

The abstract search algorithm applies the unitary transformation $(U_2U_1)^T$ to the starting state $|\psi_{start}\rangle$. We claim that under certain constraints its final state $(U_2U_1)^T|\psi_{start}\rangle$ has a sufficiently large inner product with $|\psi_{good}\rangle$.

For the quantum walk on $\sqrt{N}\times\sqrt{N}$ grid

$$|\psi_{good}\rangle = \frac{1}{2}|i,j,\Uparrow\rangle + \frac{1}{2}|i,j,\Downarrow\rangle + \frac{1}{2}|i,j,\Leftarrow\rangle + \frac{1}{2}|i,j,\Rightarrow\rangle,$$

where i,j is the marked location, and

$$|\psi_{start}\rangle = \frac{1}{2\sqrt{N}} \sum_{i,j=0}^{\sqrt{N}-1} \left(|i,j,\Uparrow\rangle + |i,j,\Downarrow\rangle + |i,j,\Leftarrow\rangle + |i,j,\Rightarrow\rangle\right).$$

Since U_2 is described by a real-value unitary matrix, its eigenvectors (with eigenvalues that are not 1 or -1) can be divided into pairs: $|\Phi_j^+\rangle$ and $|\Phi_j^-\rangle$, with eigenvalues $e^{i\theta_j}$ and $e^{-i\theta_j}$, respectively. In the case of the walk on the 2-dimensional grid we have:

Claim 9.1: [AKR05, Claim 6] Quantum walk on the 2-dimensional grid with no marked locations has $N-1$ pairs of eigenvalues $e^{\pm i\theta_j}$ that are not equal to 1 or -1. These values can be indexed by pairs $(k,l) \in \{0,1,\ldots,\sqrt{N}-1\}^2 \setminus (0,0)$. The corresponding eigenvalues are equal to $e^{\pm i\theta_{k,l}}$, where $\theta_{k,l}$ satisfies $\cos\theta_{k,l} = \frac{1}{2}(\cos\frac{2\pi k}{\sqrt{N}} + \cos\frac{2\pi l}{\sqrt{N}})$.

We use $|\Phi_{k,l}^+\rangle$ and $|\Phi_{k,l}^-\rangle$ to denote the corresponding eigenvectors. According to [MPA10, pages 3-4] these eigenvectors are equal to

$$|\Phi_{k,l}^+\rangle = |\xi_k\rangle \otimes |\xi_l\rangle \otimes |v_{k,l}^+\rangle \quad \text{and} \quad |\Phi_{k,l}^-\rangle = |\xi_k\rangle \otimes |\xi_l\rangle \otimes |v_{k,l}^-\rangle,$$

where $|\xi_k\rangle = \sum_{i=0}^{\sqrt{N}-1} \omega^{ki} \frac{1}{\sqrt[4]{N}} |i\rangle$,

$$|v_{k,l}^+\rangle = \frac{i}{2\sqrt{2}\sin\theta_{k,l}} \begin{bmatrix} e^{-i\theta_{k,l}} - \omega^k \\ e^{-i\theta_{k,l}} - \omega^{-k} \\ e^{-i\theta_{k,l}} - \omega^l \\ e^{-i\theta_{k,l}} - \omega^{-l} \end{bmatrix}, \quad |v_{k,l}^-\rangle = \frac{i}{2\sqrt{2}\sin\theta_{k,l}} \begin{bmatrix} \omega^k - e^{i\theta_{k,l}} \\ \omega^{-k} - e^{i\theta_{k,l}} \\ \omega^l - e^{i\theta_{k,l}} \\ \omega^{-l} - e^{i\theta_{k,l}} \end{bmatrix}.$$

The order of directions for the coin register is: $|\Downarrow\rangle$, $|\Uparrow\rangle$, $|\Rightarrow\rangle$, $|\Leftarrow\rangle$. The sign of $|v_{k,l}^-\rangle$ has been adjusted so that

$$\frac{1}{\sqrt{2}}|\Phi_{k,l}^+\rangle + \frac{1}{\sqrt{2}}|\Phi_{k,l}^-\rangle = |\xi_k\rangle \otimes |\xi_l\rangle \otimes |\psi_0\rangle, \tag{9.1}$$

where $|\psi_0\rangle = \frac{1}{2}|\Downarrow\rangle + \frac{1}{2}|\Uparrow\rangle + \frac{1}{2}|\Rightarrow\rangle + \frac{1}{2}|\Leftarrow\rangle$.

Due to symmetry, we can assume that $|\psi_{good}\rangle = |0\rangle \otimes |0\rangle \otimes |\psi_0\rangle$. This gives us an expression of $|\psi_{good}\rangle$ in terms of the eigenvectors of U_2:

$$|\psi_{good}\rangle = \frac{1}{\sqrt{N}} \sum_{k,l} |\xi_k\rangle \otimes |\xi_l\rangle \otimes |\psi_0\rangle =$$

$$\frac{1}{\sqrt{N}}|\psi_{start}\rangle + \frac{1}{\sqrt{2N}} \sum_{(k,l)\neq(0,0)} |\Phi_{k,l}^+\rangle + |\Phi_{k,l}^-\rangle.$$

Using the results from [AKR05], we can transform this into an expression for the final state of our quantum search algorithm. According to the first big equation in the proof of Lemma 5 in [AKR05], after $t = O(\sqrt{N \log N})$ steps we get a final state $|\psi\rangle$ such that $\||\psi\rangle - |\phi_{final}\rangle\| = o(1)$, where $|\phi_{final}\rangle = \frac{|\phi'_{final}\rangle}{\|\phi'_{final}\|}$ and

$$|\phi'_{final}\rangle = \frac{1}{\sqrt{N}}|\psi_{start}\rangle + \frac{1}{\sqrt{2N}} \sum_{(k,l)\neq(0,0)} a_{k,l}|\Phi^+_{k,l}\rangle + b_{k,l}|\Phi^-_{k,l}\rangle \tag{9.2}$$

and

$$a_{k,l} = 1 + \frac{i}{2}\cot\frac{\alpha+\theta_{k,l}}{2} + \frac{i}{2}\cot\frac{-\alpha+\theta_{k,l}}{2},$$
$$b_{k,l} = 1 + \frac{i}{2}\cot\frac{\alpha-\theta_{k,l}}{2} + \frac{i}{2}\cot\frac{-\alpha-\theta_{k,l}}{2}.$$

We now replace $\sum_{(j,j')\in S} |\alpha^t_{j,j',d}|^2$ by the corresponding sum of squares of amplitudes for the state $|\phi_{final}\rangle$. By Lemma 9.2, this changes the sum by an amount that is $o(1)$.

From [AKR05] we have $\alpha = \Theta(\frac{1}{\sqrt{N \log N}})$, $\min \theta_{k,l} = \Theta(\frac{1}{\sqrt{N}})$ and $\max \theta_{k,l} = \pi - \Theta(\frac{1}{\sqrt{N}})$. Hence, we have $\pm\alpha + \theta_{k,l} = (1 + o(1))\theta_{k,l}$ and we get

$$|\phi'_{final}\rangle = \frac{1}{\sqrt{N}}|\psi_{start}\rangle + \frac{1}{\sqrt{2N}} \sum_{(k,l)\neq(0,0)} \left(1 + i(1+o(1))\cot\frac{\theta_{k,l}}{2}\right)|\Phi^+_{k,l}\rangle +$$
$$\left(1 - i(1+o(1))\cot\frac{\theta_{k,l}}{2}\right)|\Phi^-_{k,l}\rangle. \tag{9.3}$$

This means that $\||\psi_{final}\rangle - |\phi_{final}\rangle\| = o(1)$ where $|\psi_{final}\rangle = \frac{|\psi'_{final}\rangle}{\|\psi'_{final}\|}$ and

$$|\psi'_{final}\rangle = |\psi_{good}\rangle + \frac{1}{\sqrt{2N}} \sum_{(k,l)\neq(0,0)} i\cot\frac{\theta_{k,l}}{2}\left(|\Phi^+_{k,l}\rangle - |\Phi^-_{k,l}\rangle\right). \tag{9.4}$$

Again, we can replace a sum of squares of amplitudes for the state $|\phi_{final}\rangle$ by the corresponding sum for $|\psi_{final}\rangle$ and, by Lemma 9.2, the sum changes by an amount that is $o(1)$.

We now estimate the amplitude of $|j, j', \Uparrow\rangle$ in $|\psi_{final}\rangle$. We assume that $(j, j') \neq (0, 0)$. Then, the amplitude of $|j, j', \Uparrow\rangle$ in $|\psi_{good}\rangle$ is 0. Hence, we can evaluate the amplitude of $|j, j', \Uparrow\rangle$ in

$$\frac{1}{\sqrt{2N}} \sum_{(k,l)\neq(0,0)} i\cot\frac{\theta_{k,l}}{2}(|\Phi^+_{k,l}\rangle - |\Phi^-_{k,l}\rangle) \tag{9.5}$$

and then divide the result by $\Theta(\sqrt{\log N})$, because $\|\psi'_{final}\| = \Theta(\sqrt{\log N})$.

From the definitions of $|v^{\pm}_{k,l}\rangle$ we have

$$\frac{1}{\sqrt{2}}|v^{+}_{k,l}\rangle - \frac{1}{\sqrt{2}}|v^{-}_{k,l}\rangle = \frac{i}{4\sin\theta_{k,l}} \begin{bmatrix} 2\cos\theta_{k,l} - 2\omega^{k} \\ 2\cos\theta_{k,l} - 2\omega^{-k} \\ 2\cos\theta_{k,l} - 2\omega^{l} \\ 2\cos\theta_{k,l} - 2\omega^{-l} \end{bmatrix}.$$

The amplitude of $|\Uparrow\rangle$ in this state is $\frac{i}{2\sin\theta_{k,l}}(\cos\theta_{k,l} - \omega^{-k})$. The amplitude of $|j\rangle$ in $|\xi_k\rangle$ is $\frac{1}{\sqrt[4]{N}}\omega^{kj}$. The amplitude of $|j'\rangle$ in $|\xi_l\rangle$ is $\frac{1}{\sqrt[4]{N}}\omega^{lj'}$. Therefore, the amplitude of $|j, j', \Uparrow\rangle$ in $\frac{1}{\sqrt{2}}|\Phi^{+}_{k,l}\rangle - \frac{1}{\sqrt{2}}|\Phi^{-}_{k,l}\rangle$ is

$$\frac{1}{\sqrt{N}}\omega^{kj+lj'}\frac{i}{2\sin\theta_{k,l}}(\cos\theta_{k,l} - \omega^{-k}).$$

The amplitude of $|j, j', \Uparrow\rangle$ in (9.5) is

$$\frac{1}{\sqrt{2}N}\sum_{(k,l)\neq(0,0)} i\cot\frac{\theta_j}{2}\cdot\frac{i}{2\sin\theta_{k,l}}(\cos\theta_{k,l} - \omega^{-k})\omega^{kj+lj'}.$$

By using $\sin\theta_{k,l} = 2\sin\frac{\theta_{k,l}}{2}\cos\frac{\theta_{k,l}}{2}$, we get that the amplitude of $|j, j', \Uparrow\rangle$ is

$$\frac{1}{\sqrt{2}}\sum_{(k,l)\neq(0,0)}\frac{1}{4N}\left(-\frac{\cos\theta_{k,l}}{\sin^2\frac{\theta_{k,l}}{2}}\omega^{kj+lj'} + \frac{1}{\sin^2\frac{\theta_{k,l}}{2}}\omega^{k(j-1)+lj'}\right) =$$

$$\frac{1}{\sqrt{2}}\sum_{(k,l)\neq(0,0)}\frac{1}{4N}\left(2\omega^{kj+lj'} - \frac{1}{\sin^2\frac{\theta_{k,l}}{2}}(\omega^{kj+lj'} - \omega^{k(j-1)+lj'})\right), \tag{9.6}$$

with the equality following from $\cos 2x = 1 - 2\sin^2 x$.

We can decompose the sum into two sums – one over all the first components and one over all the second components. The first component of the sum in (9.6) is close to 0 and, therefore, can be omitted. Hence, we get that the amplitude of $|j, j', \Uparrow\rangle$ in the unnormalized state $|\psi'_{final}\rangle$ can be approximated by

$$\frac{1}{\sqrt{2}}\sum_{(k,l)\neq(0,0)}\frac{1}{4N}\frac{1}{\sin^2\frac{\theta_{k,l}}{2}}(-\omega^{kj+lj'} + \omega^{k(j-1)+lj'}) =$$

$$\Theta\left(\frac{1}{N}\right)\cdot(f(j-1, j') - f(j, j')).$$

To obtain the amplitude of $|j, j', \Uparrow\rangle$ in $|\psi_{final}\rangle$, this should be divided by $\|\psi'_{final}\|$ which is of the order $\Theta(\sqrt{\log N})$. This implies Lemma 9.1.

■

9.4.2 Bounds on the probability of being close to the marked location

We start by performing some rearrangements in the expression $f(j, j')$.

Let $n = \sqrt{N}$ and S be the set of all pairs (k, l) such as $k, l \in \{0, 1, \ldots, n-1\}$, except for $(0, 0)$. We consider

$$f(j,j') = \sum_{(k,l)\in S} \frac{1}{2 - \cos\frac{2k\pi}{n} - \cos\frac{2l\pi}{n}} \omega^{kj+lj'} = \sum_{(k,l)\in S} \frac{\cos\frac{2(kj+lj')\pi}{n} + i\sin\frac{2(kj+lj')\pi}{n}}{2 - \cos\frac{2k\pi}{n} - \cos\frac{2l\pi}{n}}. \tag{9.7}$$

Since the cosine function is periodic with period 2π, we have $\cos\frac{2l\pi}{n} = \cos\frac{2(l-N)\pi}{n}$. Hence, we can replace the summation over S by the summation over

$$S' = \left\{(k,l) | k, l \in \left\{-\left\lfloor\frac{n}{2}\right\rfloor, 1, \ldots, \left\lfloor\frac{n}{2} - 1\right\rfloor\right\}\right\} \setminus \{(0,0)\}.$$

This implies that the imaginary part of (9.7) cancels out because terms in the sum can be paired up so that, in each pair, the imaginary part in both terms has the same absolute value but opposite sign. Namely:

- If none of $k, l, -k$ and $-l$ is equal to $\frac{n}{2}$, we pair up (k, l) with $(-k, -l)$.
- If none of k and $-k$ is equal to 0 or $\frac{n}{2}$, we pair up $(-\frac{n}{2}, k)$ with $(-\frac{n}{2}, -k)$ and $(k, -\frac{n}{2})$ with $(-k, -\frac{n}{2})$.
- The terms $(-\frac{n}{2}, 0)$, $(0, -\frac{n}{2})$ and $(-\frac{n}{2}, -\frac{n}{2})$ are left without a pair. This does not affect the argument because the imaginary part is equal to 0 in those terms.

Thus, we have

$$f(j,j') = \sum_{(k,l)\in S'} \frac{\cos\frac{2(kj+lj')\pi}{n}}{2 - \cos\frac{2k\pi}{n} - \cos\frac{2l\pi}{n}}.$$

We define a function $g(j, j') = f(j, j') - f(j-1, j')$. By Lemma 9.1, $Cg(j, j')$ is a good approximation for the amplitude of $|j, j', \Uparrow\rangle$ in the state of the quantum walk after $t = O(\sqrt{N \log N})$ steps.

Lemma 9.3:

$$\sum_{0<j',j<M} g^2(j,j') = \Omega(n^2 \ln M)$$

where $M = n^\epsilon$ and $\epsilon = \Omega(1)$, and $\epsilon = 1 - \Omega(1)$.

Together with Lemma 9.1, this implies that the sum of amplitudes of $|j, j', \Uparrow\rangle$ for $0 < j', j < M$ is $\Omega\left(\frac{\log M}{\log N}\right) - o(1)$. Since $\frac{\log M}{\log n} = \epsilon$ this implies Theorem 9.1.

Proof: [of Lemma 9.3] We introduce a function

$$R(M', M'', k) = \sum_{l=M'+1}^{M''} g^2(l, k)$$

where $M'' > M' > k$ and $M'' = \alpha M'$ for some α.

Claim 9.2: $|f(j, j') - \frac{n^2}{2\pi^2} f'(j, j')| = O(n^2)$ where

$$f'(j, j') = \sum_{(k,l)\in S'} \frac{\cos\frac{2(kj+lj')\pi}{n}}{k^2 + l^2}.$$

Claim 9.3: Let $j' = j\beta$ where $0 < \beta \leq 1$ and $j = n^\epsilon$, and $\epsilon = \Omega(1)$, and $\epsilon = 1 - \Omega(1)$. Then the following equality holds:

$$f'(j, j') = \frac{\pi}{2} \ln \frac{n}{j} + O(1).$$

Given these two claims, we now complete the proof of Lemma 9.3. From the inequality of quadratic and arithmetic means, we get

$$\begin{aligned} R(M', M'', k) &\geq \frac{(f(M'', k) - f(M', k))^2}{M'' - M'} \\ &= \frac{\left(\frac{n^2}{4\pi} \ln \frac{n}{M''} - \frac{n^2}{4\pi} \ln \frac{n}{M'} + O(n^2)\right)^2}{M'' - M'} \\ &= \frac{\left(\frac{n^2}{4\pi} \ln \alpha + O(n^2)\right)^2}{(\alpha - 1)M'} \\ &= \frac{\Omega(n^2)}{M'}, \end{aligned}$$

where the first equality follows from $M'', M' > k$ and Claims 9.2 and 9.3. The last equality holds if we choose an α large enough that $\frac{n^2}{4\pi} \ln \alpha + O(n^2) = \Omega(n^2)$.

We introduce a notation

$$P(M') = \sum_{l=0}^{M'-1} R(M', \alpha M', l).$$

From $R(M', M'', k) = \frac{\Omega(n^2)}{M'}$ we get $P(M') = \Omega(n^2)$. We obtain the following lower bound:

$$\sum_{0<j',j<M} g^2(j,j') > \sum_{0<j'<j<M} g^2(j,j')$$

$$> \sum_{l=1}^{\log_\alpha \sqrt{M}} P\left(\frac{M}{\alpha^l}\right) = \Omega\left(n^2 \log_\alpha \sqrt{M}\right) = \Omega(n^2 \ln M).$$

■

Proof: [of Claim 9.2]

We have

$$|f(j,j') - \frac{n^2}{2\pi^2} f'(j,j')|$$

$$\leq \sum_{(k,l)\in S'} \left|\cos\frac{2(kj+lj')\pi}{n}\right| \cdot \left|\frac{1}{2-\cos\frac{2k\pi}{n}-\cos\frac{2l\pi}{n}} - \frac{n^2}{2\pi^2(k^2+l^2)}\right|.$$

The claim now follows from $|S'| = n^2 - 1$, $|\cos x| \leq 1$ and

$$\left|\frac{1}{2-\cos\frac{2k\pi}{n}-\cos\frac{2l\pi}{n}} - \frac{n^2}{2\pi^2(k^2+l^2)}\right| \leq \frac{1}{2}.$$

To prove the last inequality, we first rewrite

$$\frac{1}{2-\cos\frac{2k\pi}{n}-\cos\frac{2l\pi}{n}} = \frac{1}{2(\sin^2\frac{k\pi}{n}+\sin^2\frac{l\pi}{n})}.$$

We have $x - \frac{x^3}{6} \leq \sin x \leq x$ for all $x \in [0,\pi]$. This implies $x^2 - \frac{x^4}{3} \leq \sin^2 x \leq x^2$. Hence, we have

$$\left|\frac{1}{2(\sin^2\frac{k\pi}{n}+\sin^2\frac{l\pi}{n})} - \frac{1}{2((\frac{k\pi}{n})^2+(\frac{l\pi}{n})^2)}\right| = \frac{(\frac{k\pi}{n})^2+(\frac{l\pi}{n})^2-(\sin^2\frac{k\pi}{n}+\sin^2\frac{l\pi}{n})}{2((\frac{k\pi}{n})^2+(\frac{l\pi}{n})^2)(\sin^2\frac{k\pi}{n}+\sin^2\frac{l\pi}{n})}$$

$$\leq \frac{(\frac{k\pi}{n})^4+(\frac{l\pi}{n})^4}{6((\frac{k\pi}{n})^2+(\frac{l\pi}{n})^2)\left((\frac{k\pi}{n})^2+(\frac{l\pi}{n})^2-\frac{(\frac{k\pi}{n})^4+(\frac{l\pi}{n})^4}{3}\right)} \leq \frac{1}{2}$$

where the last inequality follows from

$$\frac{a^2+b^2}{(a+b)\left(a+b-\frac{a^2+b^2}{3}\right)} \leq 3$$

which holds for $0 \leq a, b \leq (\frac{\pi}{2})^2$. ■

Proof: [of Claim 9.3]

We will use the notation $\alpha = \frac{2\pi}{n}$.

The following equalities hold

$$\sum_{(k,l)\in S'} \frac{\cos\alpha(kj+lj')}{k^2+l^2} = \sum_{(k,l)\in S'} \frac{\cos\alpha j(k+l\beta)}{k^2+l^2}$$

$$= \sum_{\substack{k+l\beta\le n \\ k,l\in Z^{0+} \\ (k,l)\neq(0,0)}} \frac{\cos\alpha j(k+l\beta)}{k^2+l^2} + O(1). \tag{9.8}$$

The last equality holds because it lacks some summands, with absolute value of their sum bounded above by

$$\sum_{\substack{(k,l)\in S' \\ k+l>n}} \frac{1}{k^2+l^2} = O(1).$$

It also has some new summands, with absolute value of their sum bounded above by

$$\sum_{\substack{l>n \\ 0<k<n \\ k,l\in Z^{0+}}} \frac{1}{k^2+l^2} = O(1).$$

We will use the notation $k' = k + \lceil l\beta\rceil - l\beta$. We replace the sum (9.8) (without the asymptotic) with

$$\sum_{\substack{k+l\beta\le n \\ k,l\in Z^{0+} \\ (k,l)\neq(0,0)}} \frac{\cos\alpha j(k'+l\beta)}{k^2+l^2}. \tag{9.9}$$

The error because of the replacement is

$$2\pi n^{\epsilon-1} \sum_{\substack{k+l\beta\le n \\ k,l\in Z^{0+} \\ (k,l)\neq(0,0)}} \frac{1}{k^2+l^2} \le 2\pi n^{\epsilon-1} \sum_{\substack{k,l\in Z^{0+} \\ (k,l)\neq(0,0) \\ 0\le k\le n \\ l\ge 0}} \frac{1}{k^2+l^2}$$

$$= 2\pi n^{\epsilon-1} O(\ln n) = o(1)$$

where we used the fact that $|\cos \alpha j(k' + l\beta) - \cos \alpha j(k + l\beta)| \leq 2\pi n^{\epsilon-1}$.

We replace the sum (9.9) with

$$\sum_{\substack{k+l\beta \leq n \\ k,l \in Z^{0+} \\ (k,l) \neq (0,0)}} \frac{\cos \alpha j(k' + l\beta)}{(k')^2 + l^2}. \tag{9.10}$$

The error of the last replacement is

$$\left| \sum_{\substack{k+l\beta \leq n \\ k,l \in Z^{0+} \\ (k,l) \neq (0,0)}} \left(\frac{\cos \alpha j(k' + l\beta)}{(k')^2 + l^2} - \frac{\cos \alpha j(k' + l\beta)}{k^2 + l^2} \right) \right|$$

$$\leq \sum_{\substack{k,l \in Z^{0+} \\ (k,l) \neq (0,0)}} \frac{(k')^2 - k^2}{(k^2 + l^2)^2} \leq \sum_{\substack{k,l \in Z^{0+} \\ (k,l) \neq (0,0)}} \frac{2k+1}{(k^2 + l^2)^2}$$

$$\leq 3 \sum_{\substack{k,l \in Z^{0+} \\ (k,l) \neq (0,0)}} \frac{k+l}{(k^2 + l^2)^2} \leq 12 \sum_{\substack{k,l \in Z^{0+} \\ (k,l) \neq (0,0)}} \frac{1}{(k+l)^3} = O(1).$$

We replace the sum (9.10) with

$$\sum_{\substack{k+l\beta \leq n \\ k,l \in Z^{0+} \\ (k,l) \neq (0,0)}} \cos(\alpha j(k' + l\beta)) \frac{1}{\beta} \int_{-\frac{\beta}{2}}^{\frac{\beta}{2}} \frac{dt}{(k' - t)^2 + \left(l + \frac{t}{\beta}\right)^2}. \tag{9.11}$$

Because of the last replacement the error in a fixed summand is

$$\left| \frac{1}{(k')^2 + l^2} - \frac{1}{\beta} \int_{-\frac{\beta}{2}}^{\frac{\beta}{2}} \frac{dt}{(k' - t)^2 + \left(l + \frac{t}{\beta}\right)^2} \right| =$$

$$\left| \frac{1}{(k')^2 + l^2} - \frac{\arctan\left(\frac{k'+l\beta}{(k')^2+l^2-\frac{1+\beta^2}{4}}\right)}{k'+l\beta} \right|.$$

By using $x - \frac{x^3}{3} < \arctan x < x$ that holds for all $x > 0$ we bound the error from above by

$$\left| \frac{-\frac{1+\beta^2}{4}}{((k')^2+l^2)((k')^2+l^2-\frac{1+\beta^2}{4})} \right| + \left| \frac{\left(\frac{k'+l\beta}{(k')^2+l^2-\frac{1+\beta^2}{4}}\right)^3}{3(k'+l\beta)} \right|.$$

By using the inequalities $(k')^2+l^2 \geq \frac{1}{2}(k'+l)^2$ and $(k')^2+l^2-\frac{1}{2} \geq \frac{1}{4}(k'+l)^2$ which hold if $k+l \geq 1$ and $k, l \in Z^{0+}$, and $0 < \beta \leq 1$, we obtain the following upper bound of the error:

$$\frac{4}{(k'+l)^4} + \frac{64}{3(k'+l)^4} = \frac{76}{3(k'+l)^4} \leq \frac{76}{3(k+l)^4}.$$

Thus, the error made in (9.11) can be bounded from above by

$$\sum_{\substack{k+l\beta \leq n \\ k,l \in Z^{0+} \\ (k,l) \neq (0,0)}} \frac{76}{3(k+l)^4} = O(1).$$

We replace (9.11) with

$$\frac{1}{\beta} \sum_{s=1}^{n} \cos \alpha j s \int_0^s \frac{dk}{k^2 + \left(\frac{s-k}{\beta}\right)^2}.$$

We grouped summands with equal cosine arguments. We also altered integration limits to obtain an integral on the interval $[0, s]$. The error made in this step can be bounded from above by

$$\sum_{s=1}^{n} \frac{1}{s^2} = O(1).$$

By using $\int_0^s \frac{dk}{k^2+\left(\frac{s-k}{\beta}\right)^2} = \frac{\beta\pi}{2s}$ we obtain the following sum:

$$\frac{\pi}{2}\sum_{s=1}^{n}\frac{\cos\alpha js}{s}. \tag{9.12}$$

Proposition 9.1: Let $j = n^\epsilon$ and $\epsilon = \Omega(1)$, and $\epsilon = 1 - \Omega(1)$.

The following equality holds:

$$\sum_{k=1}^{n}\frac{\cos\left(\frac{2\pi}{n}jk\right)}{k} = (1-\epsilon)\ln n + O(1).$$

Now Proposition 9.1 gives us that (9.12) is equal to

$$\frac{\pi}{2}\ln\frac{n}{j} + O(1).$$

■

Proof: [of Proposition 9.1]

We can rewrite the sum $\sum_{k=1}^{n}\frac{\cos\left(\frac{2\pi}{n}n^\epsilon k\right)}{k}$ in the following way:

$$\sum_{k=1}^{\lfloor n^{1-\epsilon}\rfloor}\frac{\cos\left(2\pi n^{\epsilon-1}k\right)}{k} + \sum_{l=1}^{\lfloor n^\epsilon\rfloor-1}\sum_{t=\lfloor n^{1-\epsilon}l\rfloor+1}^{\lfloor n^{1-\epsilon}(l+1)\rfloor}\frac{\cos\left(2\pi n^{\epsilon-1}t\right)}{t}$$

$$+\sum_{k=\lfloor n^{1-\epsilon}\lfloor n^\epsilon\rfloor\rfloor+1}^{n}\frac{\cos\left(2\pi n^{\epsilon-1}k\right)}{k} \tag{9.13}$$

Proposition 9.2:

$$\sum_{k=1}^{n}\frac{\cos\left(\frac{2\pi}{n}k\right)}{k} = \ln n + O(1).$$

Proof:

The proposition follows from

$$\sum_{k=1}^{n}\frac{\cos\left(\frac{2\pi}{n}k\right)}{k} \leq \ln n + 1.$$

and

$$\sum_{k=1}^{n} \frac{\cos\left(\frac{2\pi}{n}k\right)}{k} \geq \sum_{k=1}^{n} \frac{1}{k} - \sum_{k=1}^{n} \frac{2\pi}{n} \geq \ln n - 2\pi$$

where the first inequality in the last expression follows from $\cos x \geq 1 - x$ which holds if $x \geq 0$. ■

From proposition 9.2 we get the following equality for the first big summand of (9.13):

$$\sum_{k=1}^{\lfloor n^{1-\epsilon} \rfloor} \frac{\cos\left(2\pi n^{\epsilon-1}k\right)}{k} = (1-\epsilon)\ln n + O(1).$$

We can also obtain the following bound for the third big summand of (9.13):

$$\left| \sum_{k=\lfloor n^{1-\epsilon} \lfloor n^{\epsilon} \rfloor \rfloor + 1}^{n} \frac{\cos\left(2\pi n^{\epsilon-1}k\right)}{k} \right| < \frac{n^{1-\epsilon}+1}{n-n^{1-\epsilon}} = o(1).$$

We replace the second big summand of (9.13) with

$$\sum_{l=1}^{\lfloor n^{\epsilon} \rfloor - 1} \sum_{t=\lfloor n^{1-\epsilon} l \rfloor + 1}^{\lfloor n^{1-\epsilon} l \rfloor + \lfloor n^{1-\epsilon} \rfloor} \frac{\cos\left(2\pi n^{\epsilon-1}t\right)}{t}. \tag{9.14}$$

The error because of the replacement is

$$\left| \sum_{l=1}^{\lfloor n^{\epsilon} \rfloor - 1} \frac{\cos\left(2\pi n^{\epsilon-1} \lfloor n^{1-\epsilon}(l+1) \rfloor\right)}{\lfloor n^{1-\epsilon}(l+1) \rfloor} \right| < \sum_{l=1}^{\lfloor n^{\epsilon} \rfloor - 1} \frac{1}{n^{1-\epsilon}l} = o(1)$$

which follows from the fact that the inequality $|\lfloor x \rfloor + \lfloor y \rfloor - \lfloor x+y \rfloor| \leq 1$ holds for all x and y.

We rewrite (9.14) as

$$\sum_{t=1}^{\lfloor n^{1-\epsilon} \rfloor} \sum_{l=1}^{\lfloor n^{\epsilon} \rfloor - 1} \frac{\cos\left(2\pi n^{\epsilon-1}\left(\lfloor n^{1-\epsilon}l \rfloor + t\right)\right)}{\lfloor n^{1-\epsilon}l \rfloor + t}.$$

We get rid of the floor function in the numerator of the last expression, thus, obtaining the following sum:

$$\sum_{t=1}^{\lfloor n^{1-\epsilon} \rfloor} \cos\left(2\pi n^{\epsilon-1}t\right) p(t) \tag{9.15}$$

where

$$p(t) = \sum_{l=1}^{\lfloor n^\epsilon \rfloor - 1} \frac{1}{\lfloor n^{1-\epsilon} l \rfloor + t}.$$

Using the fact that $|\cos x - \cos y| \leq |x - y|$ holds for all x and y, we obtain that the cosine value because of the replacement changed at most by

$$\left|2\pi n^{\epsilon-1} \left(\lfloor n^{1-\epsilon} l \rfloor - n^{1-\epsilon} l\right)\right| \leq 2\pi n^{\epsilon-1}.$$

Thus, we obtain the following bound of the error of the replacement:

$$2\pi n^{\epsilon-1} \sum_{t=1}^{\lfloor n^{1-\epsilon} \rfloor} p(t) \leq 2\pi n^{\epsilon-1} n^{1-\epsilon} p(t) = o(1)$$

where we used

$$p(t) \leq \sum_{l=1}^{\lfloor n^\epsilon \rfloor - 1} \frac{1}{n^{1-\epsilon} l} < \frac{\epsilon \ln n + 1}{n^{1-\epsilon}} = o(1).$$

To prove that the expression (9.15) is $O(1)$, first we will pair almost all of it's summands so that the sum of cosine values in each pair is very close to 0.

Let $k(t) = \lfloor \frac{n^{1-\epsilon}}{2} \rfloor - t$ and $r(t) = \lfloor \frac{3n^{1-\epsilon}}{2} \rfloor - t$. We replace (9.15) with

$$\sum_{t=3}^{\lfloor \frac{n^{1-\epsilon}}{4} \rfloor - 3} \left(\cos\left(2\pi n^{\epsilon-1} t\right) p(t) + \cos\left(2\pi n^{\epsilon-1} k(t)\right) p(k(t))\right)$$

$$+ \sum_{\lfloor \frac{3n^{1-\epsilon}}{4} \rfloor + 3}^{\lfloor n^{1-\epsilon} \rfloor - 3} \left(\cos\left(2\pi n^{\epsilon-1} t\right) p(t) + \cos\left(2\pi n^{\epsilon-1} r(t)\right) p(r(t))\right) \tag{9.16}$$

where we removed some of the summands of (9.15). Let the number of the removed summands be $C = O(1)$. From $p(t) = o(1)$ we get that the error of the last replacement is $o(1)$.

Now we replace (9.16) with

$$\sum_{t=3}^{\lfloor \frac{n^{1-\epsilon}}{4} \rfloor - 3} \left(\cos\left(2\pi n^{\epsilon-1} t\right) p(t) + \cos\left(\pi - 2\pi n^{\epsilon-1} t\right) p(k(t))\right)$$

$$+ \sum_{\lfloor \frac{3n^{1-\epsilon}}{4} \rfloor + 3}^{\lfloor n^{1-\epsilon} \rfloor - 3} \left(\cos\left(2\pi n^{\epsilon-1} t\right) p(t) + \cos\left(3\pi - 2\pi n^{\epsilon-1} t\right) p(r(t))\right). \tag{9.17}$$

The error of the last replacement is $n^{1-\epsilon} \cdot 2\pi n^{\epsilon-1} \cdot o(1) = o(1)$ where the first factor is larger than the number of summands of the last sum; the second factor is the maximum change in the value of the cosine function; the third factor is $p(t) = o(1)$.

Now we can bound the maximum value of (9.17) with

$$\frac{\lfloor n^{1-\epsilon} \rfloor - C}{2} \sum_{l=1}^{\lfloor n^{\epsilon} \rfloor - 1} \frac{1}{n^{1-\epsilon} l} - \frac{\lfloor n^{1-\epsilon} \rfloor - C}{2} \sum_{l=1}^{\lfloor n^{\epsilon} \rfloor - 1} \frac{1}{n^{1-\epsilon}(l+1)}$$

$$= \frac{\lfloor n^{1-\epsilon} \rfloor - C}{2} \left(\frac{1}{n^{1-\epsilon}} - \frac{1}{n^{1-\epsilon} \lfloor n^{\epsilon} \rfloor} \right) = O(1).$$

■

9.5 Conclusions

We have studied a search by quantum walks on a finite two-dimensional grid with one marked location according to [AKR05]. We have shown that while the original quantum walk can only find the marked location with probability $O(1/\log N)$, it actually returns a location close to the marked one with high probability. This allows us to replace amplitude amplification with classical post processing which does not increase time complexity of the algorithm and, thus, leads to $O(\sqrt{\log N})$ speed-up.

We have shown the effect of localization of probability around the marked location for one marked location case. However, our numerical simulations show the very same effect for multiple marked locations. Moreover, as the model of quantum walk is very generic, we expect the same effect for other types of graphs. This, if proven, can be used to improve various quantum walk based algorithms.

We were unable to prove our conjecture as our analysis technique is very dependent on the structure of a graph as well as number and positions of marked locations. This is due to expressing the initial state of the algorithm in terms of eigenvectors of a single step of the walk. Every change in structure of the graph or positions of marked locations change the eigenvectors. Therefore, the analysis should be started from the very beginning. Thus, the important question is to find more appropriate techniques for approximation of final state of general graph.

Our result opens a very natural question: if the quantum search on a graph is stopped after a certain number of steps and outputs an element, how far is the output from a marked location. This question has never been studied before. We have solved the problem for a special case of the two-dimensional grid. The more general solution is still to be found.

10. CONCLUSIONS

In this thesis we studied the power and the limits of quantum computation. We examined two models: quantum finite automata and quantum search algorithms in the query model.

First, we examined one-way quantum finite automata. This is the case where quantum computation demonstrates a clear advantage over classical counterparts, namely deterministic and probabilistic one-way finite automata. We studied a space-efficiency of quantum finite automata and have improved best known exponential separation between quantum and classical finite automata.

Next, we examined quantum search algorithms in the query model. We studied Grover's quantum search algorithm, which is one of most important and widely known quantum query algorithms. Similarly to quantum automata case Grover's algorithm provides a significant speed-up over any deterministic and probabilistic algorithms. The algorithm, however, is very sensitive to errors in queries and may completely lose its superiority over classical algorithms [RS08]. This makes Grover's algorithm a good candidate for study of the limits of the quantum query model. We analysed a behaviour of Grover's algorithm in two different models of query errors – model of [RS08] and it's generalization – and have shown that in both cases the algorithm loses its quantum speed-up. Our analysis provides useful insights into fault-tolerance of quantum query algorithms. However, many questions are still open. The main open question (stated in [RS08]) is if there exist a search problem for which a quantum speed-up is achievable in the faulty query model?

Lastly, we examined a problem of search on a two-dimensional grid. The problem is also formulated in the query model. We show that despite of quantum search problem being well studied there still exists unnoticed effects, which can be used to build efficient quantum search algorithms. We demonstrate one such effect – localization of probability around the marked location – and use it to improve (speed-up) quantum walk search algorithm by [AKR05]. We expect the effect to be applicable to other search problems (with other structure of the search space). However, the limit of its applicability is still to be understood.

The thesis does not provide a complete theory of effects and models being studied. There are still many questions to be answered. Nevertheless, it provides a number of notable results and new approaches and is a step towards understanding the power and the limits of quantum computation.

BIBLIOGRAPHY

Part I references

[Aar04] S. Aaronson. Limits on Efficient Computation in the Physical World, *arXiv:quant-ph/0412143*, 2004.

[BB+97] C. H. Bennett, E. Bernstein, G. Brassard, U. Vazirani. Strengths and Weaknesses of Quantum Computing, *SIAM Journal on Computing (special issue on quantum computing)*, 26(5):1510-1523, 1997.

[DW11] A. Drucker, R. de Wolf. Quantum Proofs for Classical Theorems, *arXiv:0910.3376 [quant-ph]*, 2011.

[Fey82] R. Feynman. Simulating Physics with Computers, *International Journal of Theoretical Physics*, 21(6/7), 1982.

[FG+00] E. Farhi, J. Goldstone, S. Gutmann, M. Sipser. Quantum Computation by Adiabatic Evolution, *arXiv:quant-ph/0001106*, 2000.

[Joz05] R. Jozsa. An introduction to measurement based quantum computation, *arXiv:quant-ph/0508124*, 2005.

[Gro96] L. K. Grover. A fast quantum mechanical algorithm for database search, *Proceedings of the 28th ACM STOC*, 212-219, 1996.

[Lad75] R. Ladner. On the Structure of Polynomial Time Reducibility, *Journal of the ACM*, 22(1):155-171, 1975.

[KLM07] P. Kaye, R. Laflamme, M. Mosca. An Introduction to Quantum Computing, *Oxford University Press*, 2007.

[Lay11] D. C. Lay. Linear Algebra and Its Applications, 4th edition, *Pearson*, 2011.

[Pen89] R. Penrose. The Emperor's New Mind: Concerning Computers, Minds and The Laws of Physics, *Oxford University Press*, 1989.

[Sho97] P. W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, *SIAM Journal on Computing*, 26(5):1484-1509, 1997.

[Wat06] J. Watrous. Quantum Computation, *Lecture course "CPSC 519/619"*, University of Calgary, 2006. Available at `http://www.cs.uwaterloo.ca/~watrous/lecture-notes.html`

Part II references

[AI+00] M. Ajtai, H. Iwaniec, J. Komlos, J. Pintz, E. Szemeredi. Construction of a thin set with small Fourier coefficients, *Bulletin of the London Mathematical Society*, 22:583-590, 1990.

[AF98] A. Ambainis, R. Freivalds. 1-way quantum finite automata: strengths, weaknesses and generalizations, *Proceedings of FOCS'98*, 332-341, 1998.

[AN08] A. Ambainis, N. Nahimovs. Improved constructions of quantum automata, *Proceedings of TQC 2008*, 47-56, 2008.

[AN09] A. Ambainis, N. Nahimovs. Improved constructions of quantum automata, *Theoretical Computer Science (special issue on probabilistic and quantum automata)*, 410:1916-1922, 2009.

[BMP03] A. Bertoni, C. Mereghetti, B. Palano. Quantum Computing: 1-Way Quantum Automata, *Developments in Language Theory*, Lecture Notes in Computer Science, 2710:1-20, 2003.

[BMP03a] A. Bertoni, C. Mereghetti, B. Palano. Lower Bounds on the Size of Quantum Automata Accepting Unary Languages, *Theoretical Computer Science*, Lecture Notes in Computer Science, 2841:86-96, 2003.

[Bou05] J. Bourgain. Estimates on exponential sums related to Diffie-Hellman distributions, *Geometric and Functional Analysis*, 15:1-34, 2005.

[Cia01] M. Ciamarra. Quantum Reversibility and a New Model of Quantum Automaton, *Proceedings of FCT'01*, 376-379, 2001.

[CM00] C. Moore, J. Crutchfield. Quantum automata and quantum grammars, *Theoretical Computer Science*, 237(1-2):275-306, 2000.

[Gal06] F. Le Gall. Exponential separation of quantum and classical online space complexity, *Proceedings of SPAA'06*, 67-73, 2006.

[Hir11] M. Hirvensalo. Quantum Automata with Open Time Evolution, *International Journal of Natural Computing Research*, 1(1):70-85, 2010.

[KW97] A. Kondacs, J. Watrous. On the power of quantum finite state automata, *Proceedings of FOCS'97*, 66-75, 1997.

[MP01] C. Mereghetti, B. Palano. Upper Bounds on the Size of One-Way Quantum Finite Automata, *Proceedings of ICTCS'2001*, 123-135, 2001.

[MR94] R. Motwani, P. Raghavan. Randomized Algorithms, *Cambridge University Press*, 1994.

[RSW93] A. Razborov, E. Szemeredi, A. Wigderson. Constructing small sets that are uniform in arithmetic progressions, *Combinatorics, Probability and Computing*, 2:513-518, 1993.

Part III references

[Amb07a] A. Ambainis. A nearly optimal discrete query quantum algorithm for evaluating NAND formulas, *arXiv:0704.3628 [quant-ph]*, 2007.

[AB+13] A. Ambainis, A. Bačkurs, N. Nahimovs, A. Rivosh. Grover's algorithm with errors, *Proceedings of MEMICS 2012*, 180-189, 2013.

[BH+00] G. Brassard, P. Hoyer, M. Mosca, A. Tapp. Quantum Amplitude Amplification and Estimation, *arXiv:quant-ph/0005055*, 2000.

[BN+05] H. Buhrman, I. Newman, H. Roehrig, R. de Wolf. Robust Polynomials and Quantum Algorithms, *Proceedings of STACS'2005*, 593-604, 2005.

[Gud08] S. Gudder. Quantum Markov chains, *Journal of Mathematical Physics*, 49(7), 2008.

[HJ06] R. Horn, C. Johnson. Matrix Analysis, *Cambridge University Press*, 2006.

[KNR12] D. Kravchenko, N. Nahimovs, A. Rivosh. On fault-tolerance of Grover's algorithm, *Scientific Papers University of Latvia*, 787:135-145, 2012.

[LL+00] G. L. Long, Y. S. Li, W. L. Zhang, C. C. Tu. An intrinsic limitation on the size of quantum database, *Physical Review A*, 61, 2000.

[LP11] C. Liu, N. Petulante. On limiting distributions of quantum Markov chains, *arXiv:1010.0741 [quant-ph]*, 2011.

[NR10] N. Nahimovs, A. Rivošs. A note on the optimality of the Grover's algorithm, *Scientific Papers University of Latvia*, 756:221-225, 2010.

[RS08] O. Regev, L. Schiff. Impossibility of a Quantum Speed-up with a Faulty Oracle, *Proceedings of ICALP'2008*, 773-781, 2008.

[SBW03] N. Shenvi, K. R. Brown, K. B. Whaley. Effects of Noisy Oracle on Search Algorithm Complexity, *Physical Review A*, 68(5), 2003.

[SMB03] D. Shapira, S. Mozes, O. Biham. Effect of unitary noise on Grover's quantum search algorithm, *Physical Review A*, 67(4), 2003.

[Zal99] C. Zalka. Grovers quantum searching algorithm is optimal, *Physical Review A*, 60:2746-2751, 1999.

Part IV references

[AA03] S. Aaronson, A. Ambainis. Quantum search of spatial regions, *Proceedings of FOCS'03*, 200-209, 2003.

[AB+12] A. Ambainis, A. Bačkurs, N. Nahimovs, R. Ozols, A. Rivosh. Search by quantum walks on two dimensional grid without amplitude amplification, *Proceedings of TQC 2012*, 87-97, 2012.

[Amb03] A. Ambainis. Quantum walks and their algorithmic application, *International Journal of Quantum Information*, 1(2003):507-518, 2003.

[Amb07] A. Ambainis. Quantum walk algorithm for element distinctness, *SIAM Journal on Computing*, 37:210-239, 2007.

[AKR05] A. Ambainis, J. Kempe, A. Rivosh. Coins make quantum walks faster, *Proceedings of SODA'05*, 1099-1108, 2005.

[AR08] A. Ambainis, A. Rivosh. Quantum random walks with multiple or moving marked locations, *Proceedings of SOFSEM'08*, 485-496, 2008.

[Ben02] P. Benioff. Space searches with a quantum robot, *AMS Contemporary Math Series*, 305(2002):1-12, 2002.

[BS06] H. Buhrman, R. Spalek. Quantum Verification of Matrix Products, *Proceedings SODA'06*, 880-889, 2006.

[BV97] E. Bernstein, U. Vazirani. Quantum complexity theory, *SIAM Journal on Computing*, 26:1411-1473, 1997.

[CC+03] A.M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, D. A. Spielman. Exponential algorithmic speedup by a quantum walk, *Proceedings of the 35th ACM STOC*, 59-68, 2003.

[Kem03] J. Kempe. Quantum random walks – an introductory overview, *Contemporary Physics*, 44(4):302-327, 2003.

[KM+10] H. Krovi, F. Magniez, M. Ozols, J. Roland. Finding is as easy as detecting for quantum walks, *Proceedings of ICALP'10*, 540-551, 2010.

[MPA10] F. L. Marquezino, R. Portugal, G. Abal. Mixing times in quantum walks on two-dimensional grids, *Physical Review A*, 82(4), 2010.

[MSS05] F. Magniez, M. Santha, M. Szegedy. An $O(n^{1.3})$ quantum algorithm for the triangle problem, *Proceedings of SODA'05*, 413-424, 2005.

[MN+09] F. Magniez, A. Nayak, P. Richter, M. Santha. On the hitting times of quantum versus random walks, *Proceedings of SODA'09*, 86-95, 2009.

[Sze04] M. Szegedy. Quantum speed-up of Markov Chain based algorithms, *Proceedings of FOCS'04*, 32-41, 2004.

[Tul08] A. Tulsi. Faster quantum-walk algorithm for the two-dimensional spatial search, *Physical Review A*, 78(1), 2008.