Quantum algorithms for string matching

Thesis by

Jeffrey A. Aborot
Bachelor of Science in Computer Science

Submitted to the National Graduate School of Engineering
College of Engineering
University of the Philippines

In Partial Fulfillment of the Requirements
For the Degree of Master
in Computer Science

National Graduate School of Engineering
College of Engineering
University of the Philippines Diliman
Quezon City

July 2016

This dissertation, entitled QUANTUM ALGORITHMS FOR STRING MATCHING, prepared and submitted by JEFFREY A. ABOROT, in partial fulfillment of the requirements for the degree of MASTER IN COMPUTER SCIENCE is hereby accepted.

HENRY N. ADORNA, PhD
Thesis Adviser

Prospero C. Naval Jr., PhD
Thesis Panel Chairman

Eric A. Galapon, PhD
Thesis Panel Member

Accepted as partial fulfillment of the requirements for the degree MASTER IN COMPUTER SCIENCE.

AURA C. MATIAS, Ph.D.
Dean

UNIVERSITY OF THE PHILIPPINES

Master in Computer Science

Jeffrey A. Aborot

Quantum algorithms for string matching

Thesis Adviser:
Henry N. Adorna, PhD
Department of Computer Science
University of the Philippines Diliman

Date of Submission:
July 2016

Permission is given for the following people to have access to this dissertation:

Available to the general public	Yes
Available only after consultation with author/dissertation adviser	No
Available only to those bound by confidentiality agreement	No

Signature of student

Signature of adviser

Acknowledgements

Abstract

ANTINERO ABOROT, JEFFREY. QUANTUM ALGORITHMS FOR STRING MATCHING. (Under the direction of HENRY N. ADORNA, PhD) In this thesis we present three quantum algorithms for the exact and approximate string matching problem. Our first algorithm is an amplitude amplification-based quantum algorithm for the exact string matching problem which has time complexity in $\mathcal{O}\left(\left\lfloor \frac{\pi}{4}\sqrt{N}\right\rfloor (\log_2 N + \log_2 |\Sigma|)\right)$ with additional logarithmic terms for error of approximation and space complexity in $\mathcal{O}\left(NM \log_2 |\Sigma| + M \log_2 |\Sigma| + M + \log_2 N\right)$. It outputs a solution index in a text with probability ≈ 1. We provided a quantum circuit construction for the oracle function for comparing sequence of symbols in an alphabet which is absent in the quantum algorithms reviewed for this study. We also provided deeper complexity analysis as compared to reviewed related quantum algorithm for the same problem.

Our second algorithm is a quantum Fourier transform-based quantum algorithm for the approximate string matching problem with time complexity in $\mathcal{O}\left(K\left(\log_2^2(N+M) + q\log_2(N+M)\right)\right)$ and space complexity in $\mathcal{O}\left(|\Sigma| \log_2(N+M) + KM \log_2 |\Sigma|\right)$. It returns a solution index in a text with probability proportional to the number of matching symbols between the solution subsequence of the text and pattern. We used the concept of convolution in digital signal processing to compute the number of matching symbols between two sequences. Lastly, our third algorithm is a quantum algorithm for approximate string matching based on a classical filtering method. It has time complexity in $\mathcal{O}\left(\log_2 N + |\Sigma| \log_2 M\right)$ and space complexity in $\Omega\left(N \log_2 |\Sigma| + \log_2 N + \log_2 |\Sigma|\right)$. The algorithm outputs a solution index with probability proportional to the number of matching symbols between the solution subsequence of the text and the pattern. We designed quantum symbol operators for identifying the first occurrence index of distinct symbols in the pattern. This provides reusability when searching for the same pattern on different input texts which is comparable to the level of reusability provided by a reviewed related quantum algorithm for the same problem. We also provided quantum circuit design for the quantum operators used in each of our algorithms' sub-routines. In comparison to the reviewed related quantum algorithms for the same problem, our algorithms provide lower time complexity.

Keywords: *Unconventional computing, Quantum computing, String matching, Quantum algorithms, Pattern matching, Computing model*

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed, and that I have marked any citations accordingly.

Quezon City, Philippines, July 2016

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Part I

Preliminaries

Chapter 2

String matching

Chapter 3

Quantum computing

Part II

Body of Work

Chapter 4

An oracle design for Grover's quantum search algorithm for exact string matching

4.1 Problem

people's contact arranged or indexed in no particular order and we want to search for a specific person from the set. We cannot instantaneously identify the index of the name of the person from the space since the elements of the space are not sorted in any way. Assuming the existence of a so called *oracle*, Grover's quantum algorithm will return the index of the name of the person we are searching for using $\mathcal{O}\left(\sqrt{N}\right)$ queries to the oracle. The oracle is an abstract construct which returns either a *Yes* (1) or *No* (0) answer when given input. In our example, the oracle in Grover's quantum search algorithm is given as input an index from the set $\{0, 1, \ldots, N-1\}$. The oracle gives a *Yes* answer if the element of the space at the input index is the name of the person we are searching for. Otherwise, it outputs a *No* answer.

The use of the concept of an oracle in Grover's quantum search algorithm provides the means for analyzing the number of iterations of the Grover iterate operator G by encapsulating the actual process of identifying the solution index i_x into a black-box structure. This black-box structure is assumed to have a constant time complexity $O(1)$ and that it always returns the correct output $f(\beta(i_x)) \in \{0, 1\}$ when provided with the binary input $\beta(i_x) \in \{0, 1\}^{\log(N)}$.

4.2 Idea

4.3 Contributions

In this chapter we provide a unitary operator which provides the details of the oracle in Grover's quantum search algorithm. This unitary operator provides oracle work specific to the exact string matching problem described in the previous chapter. In this chapter provide the following contributions.

- We describe in detail a unitary operator U_g which given the binary representation of an index i, $\beta(i) \in \{0,1\}^{\log(N)}$, in T as input identifies if index i is a starting position of the pattern P in T, $g(\beta(i)) \in 0, 1$.
- We prove that the unitary operator U_g marks an input index i in T if and only if $H\left(T[i : i + M - 1], P\right) = 0$.
- We show that we can use Grover's quantum search algorithm for the exact string matching problem by replacing the oracle with the unitary operator U_g and that the solution index is identified in time complexity $O\left(\sqrt{N}\left(\log(|\Sigma|) + M\right)\right)$ with probability ≈ 1.

4.4 Notation

We use the following notations in the succeeding sections of this chapter.

- T = T[0],T[1],. . .,T[N-1] - a text of length N where $\mathrm{T}[i] \in \Sigma$ for $0 \leq i \leq N - 1$
- T[i],T[i+1],. . .,T[i+M-1] - an M-length substring of T starting at index i; we also use the equivalent shorthand notation $\mathrm{T}[i : i + M - 1]$
- $\mathrm{T}_s = \{\mathrm{T}[i : i + M - 1] \mid 0 \leq i \leq N - M + 1\}$ - the set of all M-length substrings of T
- P = P[0],P[1],. . .,P[M-1] - a pattern of length M where $\mathrm{P}[i] \in \Sigma$ for $0 \leq i \leq M - 1$

4.5 Quantum registers

Given the superposition state of the index register which encodes each index in T,

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle$$

and a *substring register* initialized in the state

$$|0\rangle^{\otimes M \lceil \log(|\Sigma|) \rceil}$$

we encode each M-length substring of T into a superposition state of the substring register,

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle |\mathrm{T}[i : i + M - 1]\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \left(\otimes_{j=0}^{M-1} \otimes_{k=0}^{\lceil \log(|\Sigma|) \rceil - 1} |\beta(\mathrm{T}[i + j])_k\rangle \right)$$

where

$$\beta(\cdot) : \Sigma \rightarrow \{0,1\}^{\lceil \log(|\Sigma|) \rceil}$$

is a function which maps a symbol in Σ into a unique binary sequence of length $\lceil \log(|\Sigma|) \rceil$. This step assumes the existence of a *quantum random access memory* which keeps a copy of all M-length substrings in T. One model of such quantum memory is the *qRAM* [**Giovanetti2008**, **Giovannetti2008a**]. A qRAM facilitates the memory access operation

$$\sum_j \alpha_j |j\rangle_a |0\rangle_d \rightarrow \sum_j \alpha_j |j\rangle_a |D_j\rangle_d$$

where $|D_j\rangle_d$ is the state which encodes the data stored in the qRAM's data register. This data is indexed with the state $|j\rangle_a$ of the qRAM's address register. If the state of a qRAM's address register is a superposition state $\sum_j \alpha_j |j\rangle_a |0\rangle_d$, it puts its data register into a superposition of states which encode the data indexed by the superposition state of its address register, $\sum_j \alpha_j |j\rangle_a |D_j\rangle_d$.

Likewise, we encode P into a pattern register of size $M\lceil \log(|\Sigma|) \rceil$. The state of the index register, substring register and pattern register is then the superposition state

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle |\mathrm{T}[i : i+M-1]\rangle |\mathrm{P}\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \left(\otimes_{j=0}^{M-1} \otimes_{k=0}^{\lceil \log(|\Sigma|) \rceil - 1} |\beta(\mathrm{T}[i+j])_k\rangle \right) \left(\otimes_{j=0}^{M-1} \otimes_{k=0}^{\lceil \log(|\Sigma|) \rceil} |\beta(\mathrm{P}[j])_k\rangle \right)$$

We also prepare a *scratch register* with size $M(\lceil \log(|\Sigma|) \rceil + 1)$ in the state

$$|0\rangle^{\otimes M(\lceil \log(|\Sigma|) \rceil + 1)}$$

and an *output register* in the state $|0\rangle$. The initial states of the registers prior to the consultation to the oracle will be the superposition state

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \left(\otimes_{j=0}^{M-1} \otimes_{k=0}^{\lceil \log(|\Sigma|) \rceil - 1} |\beta(\mathrm{T}[i+j])_k\rangle \right) \left(\otimes_{j=0}^{M-1} \otimes_{k=0}^{\lceil \log(|\Sigma|) \rceil} |\beta(\mathrm{P}[j])_k\rangle \right) \left(|0\rangle^{\otimes M(\lceil \log(|\Sigma|) \rceil + 1)} \right) \otimes |0\rangle \tag{4.1}$$

Figure **??** shows an arrangement of the registers superimposed against an empty quantum circuit which we will fill up with unitary operators in the succeeding section.

4.6 Oracle operator U_g

We replace the oracle in Grover's quantum search algorithm with a deterministic unitary operator which we denote as U_g. We define operator U_g with the functional definition

$$U_g |x\rangle = \sum_{x \in \{\mathrm{T}_s\}} (-1)^{g(x)} |x\rangle$$

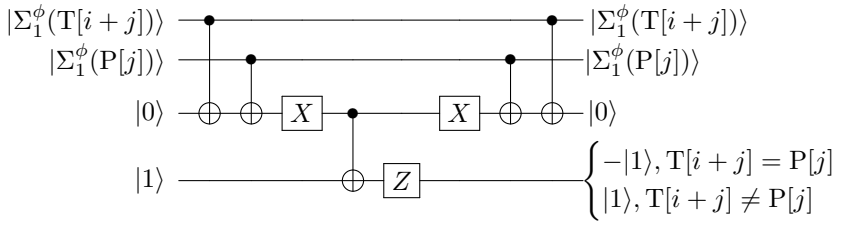

Figure 4.1: A 2-qubit symbol comparator circuit (with ancillary qubits) for comparing two symbols $\mathrm{T}[i+j]$ and $\mathrm{P}[j]$ assuming $|\Sigma| = 2$ for example purposes. The top qubit is initialized in state $|\Sigma^\phi(\mathrm{T}[i+j])\rangle$ and the next qubit is initialized in state $|\Sigma^\phi(\mathrm{P}[j])\rangle$. The bottom two qubits serve as ancillary bits for keeping the results of comparison and are initialized in state $|1\rangle$. The bottommost qubit serves as a binary indicator and is put into state $|1\rangle$ if symbols $\mathrm{T}[i+j]$ and $\mathrm{P}[j]$ do not match and $-|1\rangle$ otherwise. The circuit can be extended to accommodate larger $|\Sigma|$ by adding qubits to the substring, pattern and ancillary register.

where

$$g(x) = \begin{cases} 1 & x = \beta(\mathrm{P}) \\ 0 & \text{otherwise} \end{cases}$$

We describe the operation of U_g as a quantum circuit composed of 2-qubit and 1-qubit quantum gates acting on the registers. Figure 4.1 is a quantum circuit for operator U_g when $|\Sigma| = 2$. The circuit compares a pair of symbols $\mathrm{T}[i+j]$ and $\mathrm{P}[j]$ in T and P. Figure 4.2 is a quantum circuit for U_g for the case $|\Sigma| = 4$. Each symbol in Σ is represented by 2 qubits. The circuit compares each pair of qubits of the pair of symbols $\mathrm{T}[i+j]$ and $\mathrm{P}[j]$ in T and P.

Figure **??** shows a quantum circuit for operator U_g for arbitrary alphabet size $|\Sigma|$. The circuit compares all pairs of bits representing each pair of symbols in T and P.

The unitary operators in the quantum circuit in Figure **??** can be grouped into the following groups of operators

1. $M(\lceil \log(|\Sigma|) \rceil)$ CNOT operators acting on the substring and scratch register
2. $M(\lceil \log(|\Sigma|) \rceil)$ CNOT operators acting on the pattern and scratch register
3. $M(\lceil \log(|\Sigma|) \rceil)$ Pauli-X operators acting on the scratch register
4. M multiple-control $C^{\log(|\Sigma|)}NOT$ operators acting on the scratch register
5. One multiple-control $C^M NOT$ operator acting on the scratch register
6. One Pauli-Z operator acting on the output register
7. a mirror of the group of operators in the left half of the quantum circuit for resetting the states of the registers

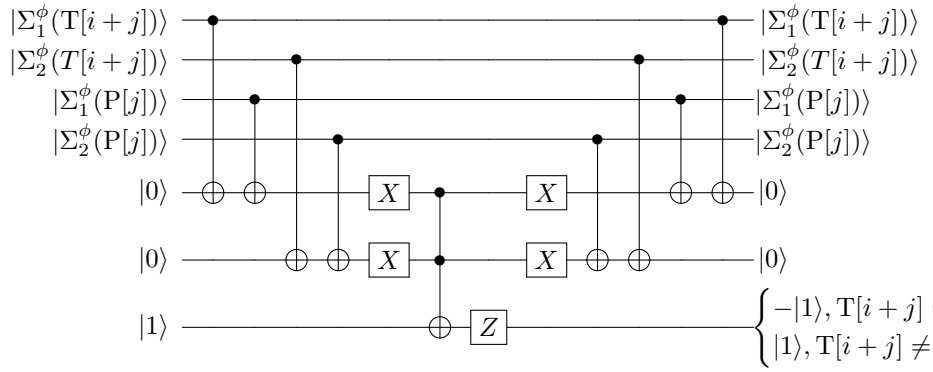

Figure 4.2: A 4-qubit symbol comparator circuit (with ancillary qubits) for comparing two symbols $\mathrm{T}[i+j]$ and $\mathrm{P}[j]$ assuming $|\Sigma| = 4$ for example purposes. The top two qubits are initialized in state $|\Sigma^\phi(\mathrm{T}[i+j])\rangle$ and the next two qubits are initialized in state $|\Sigma^\phi(\mathrm{P}[j])\rangle$. The bottom three qubits serve as ancillary qubits for holding the results of comparison and are initialized in state $|1\rangle$. The bottommost qubit serves as a binary indicator and is put into state $|1\rangle$ if the symbols do not match and into state $-|1\rangle$ otherwise.

The first set of CNOT operators operate on the substring and scratch register. The control of these CNOT operators will be the $M(\lceil\log(|\Sigma|)\rceil)$ qubits of the substring register and their target will be the $M(\lceil\log(|\Sigma|)\rceil)$ qubits of the scratch register. The entangled state of the index, substring and scratch register from Equation 4.1 (where the state of the pattern and output register are not shown for brevity) can be written as

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \left(\otimes_{j=0}^{M-1} \otimes_{k=0}^{\lceil\log(|\Sigma|)\rceil-1} |\beta(\mathrm{T}[i+j])_k\rangle \right) \left(|0\rangle^{\otimes M(\lceil\log(|\Sigma|)\rceil+1)} \right) \tag{4.2}$$

and the operation of the first set of CNOT operators is defined such that

$$\begin{aligned}
&\frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \left(\otimes_{j=0}^{M-1} \otimes_{k=0}^{\lceil\log(|\Sigma|)\rceil-1} |\beta(\mathrm{T}[i+j])_k\rangle \right) \left(|0\rangle^{\otimes M(\lceil\log(|\Sigma|)\rceil+1)} \right) \\
&\rightarrow \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \left(\otimes_{j=0}^{M-1} \otimes_{k=0}^{\lceil\log(|\Sigma|)\rceil-1} |\beta(\mathrm{T}[i+j])_k\rangle \right) \left(\left(\otimes_{j=0}^{M-1} \otimes_{k=0}^{\lceil\log(|\Sigma|)\rceil-1} |0 \oplus \beta(\mathrm{T}[i+j])_k\rangle \right) \otimes |0\rangle^{\otimes M} \right) \\
&= \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \left(\otimes_{j=0}^{M-1} \otimes_{k=0}^{\lceil\log(|\Sigma|)\rceil-1} |\beta(\mathrm{T}[i+j])_k\rangle \right) \left(\left(\otimes_{j=0}^{M-1} \otimes_{k=0}^{\lceil\log(|\Sigma|)\rceil-1} |\beta(\mathrm{T}[i+j])_k\rangle \right) \otimes |0\rangle^{\otimes M} \right)
\end{aligned}$$

The second set of CNOT operators operate on the pattern and scratch register. The control of these operators will be the $M(\lceil\log(|\Sigma|)\rceil)$ qubits of the pattern register and their target will be the $M(\lceil\log(|\Sigma|)\rceil)$ qubits of the scratch register. Likely, their entangled state (where the state of the

substring and output register are not shown for brevity) can be written as

$$\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle\left(\otimes_{j=0}^{M-1}\otimes_{k=0}^{\lceil\log(|\Sigma|)\rceil}|\beta(\mathrm{P}[j])_k\rangle\right)\left(\left(\otimes_{j=0}^{M-1}\otimes_{k=0}^{\lceil\log(|\Sigma|)\rceil-1}|\beta(\mathrm{T}[i+j])_k\rangle\right)\otimes|0\rangle^{\otimes M}\right) \quad (4.3)$$

and the operation of the second set of CNOT operators as

$$\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle\left(\otimes_{j=0}^{M-1}\otimes_{k=0}^{\lceil\log(|\Sigma|)\rceil}|\beta(\mathrm{P}[j])_k\rangle\right)\left(\left(\otimes_{j=0}^{M-1}\otimes_{k=0}^{\lceil\log(|\Sigma|)\rceil-1}|\beta(\mathrm{T}[i+j])_k\rangle\right)\otimes|0\rangle^{\otimes M}\right)$$
$$\rightarrow\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle\left(\otimes_{j=0}^{M-1}\otimes_{k=0}^{\lceil\log(|\Sigma|)\rceil}|\beta(\mathrm{P}[j])_k\rangle\right)\left(\left(\otimes_{j=0}^{M-1}\otimes_{k=0}^{\lceil\log(|\Sigma|)\rceil-1}|\beta(\mathrm{T}[i+j])_k\oplus\beta(\mathrm{P}[j])_k\rangle\right)\otimes|0\rangle^{\otimes M}\right)$$

The third set of operators will be the $M\lceil\log(|\Sigma|)\rceil$ Pauli-X operators which will operate on the scratch register alone. These operators will flip the state of the $M\lceil\log(|\Sigma|)\rceil$ qubits of the scratch register. Their operation on the scratch register can be written as

$$\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle\left(\left(\otimes_{j=0}^{M-1}\otimes_{k=0}^{\lceil\log(|\Sigma|)\rceil-1}|\beta(\mathrm{T}[i+j])_k\oplus\beta(\mathrm{P}[j])_k\rangle\right)\otimes|0\rangle^{\otimes M}\right)$$
$$\rightarrow\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle\left(\left(\otimes_{j=0}^{M-1}\otimes_{k=0}^{\lceil\log(|\Sigma|)\rceil-1}|\neg\left(\beta(\mathrm{T}[i+j])_k\oplus\beta(\mathrm{P}[j])_k\right)\rangle\right)\otimes|0\rangle^{\otimes M}\right)$$

The next set of operators are multiple-control M $C^{\lceil\log(|\Sigma|)\rceil}NOT$ operators which will operate on the scratch register. The control qubits of these operators will be the $(j(\lceil\log(\Sigma)\rceil+1)+k)$-th qubits of the scratch register, for $0\leq j<M, 0\leq k<\lceil\log(\Sigma)\rceil$. The targets of these operators will be the $(j(\lceil\log(\Sigma)\rceil+1)+\lceil\log(\Sigma)\rceil)$-th qubits of the scratch register, for $0\leq j<M$. Operation of these operators on the scratch register can be written as

$$\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle\left(\left(\otimes_{j=0}^{M-1}\otimes_{k=0}^{\lceil\log(|\Sigma|)\rceil-1}|\neg\left(\beta(\mathrm{T}[i+j])_k\oplus\beta(\mathrm{P}[j])_k\right)\rangle\right)\otimes|0\rangle^{\otimes M}\right)$$
$$\rightarrow\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle\left(\left(\otimes_{j=0}^{M-1}\otimes_{k=0}^{\lceil\log(|\Sigma|)\rceil-1}|\neg\left(\beta(\mathrm{T}[i+j])_k\oplus\beta(\mathrm{P}[j])_k\right)\rangle\right)\right.$$
$$\left.\otimes_{j=0}^{M-1}|0\oplus\left(\wedge_{k=0}^{\lceil\log(|\Sigma|)\rceil-1}\neg\left(\beta(\mathrm{T}[i+j])_k\oplus\beta(\mathrm{P}[j])_k\right)\right)\rangle\right)$$
$$=\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle\left(\left(\otimes_{j=0}^{M-1}\otimes_{k=0}^{\lceil\log(|\Sigma|)\rceil-1}|\neg\left(\beta(\mathrm{T}[i+j])_k\oplus\beta(\mathrm{P}[j])_k\right)\rangle\right)\right.$$
$$\left.\otimes_{j=0}^{M-1}|\wedge_{k=0}^{\lceil\log(|\Sigma|)\rceil-1}\neg\left(\beta(\mathrm{T}[i+j])_k\oplus\beta(\mathrm{P}[j])_k\right)\rangle\right)$$

The succeeding operator is a single multiple-control $C^{M}NOT$ operator which will operate on the scratch register and the output register. The control of this operator will be the $(j(\lceil\log(\Sigma)\rceil+$

$1) + \lceil \log(\Sigma) \rceil)$-th qubits of the scratch register and its target will be the single qubit of the output register. Given the current state of the scratch register, this operator will transform the state of the output register such that

$$\begin{aligned}
&\frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \left(\left(\otimes_{j=0}^{M-1} \otimes_{k=0}^{\lceil \log(|\Sigma|) \rceil - 1} |\neg \left(\beta(\mathrm{T}[i+j])_k \oplus \beta(\mathrm{P}[j])_k \right) \rangle \right) \right. \\
&\qquad \left. \otimes_{j=0}^{M-1} \left| \wedge_{k=0}^{\lceil \log(|\Sigma|) \rceil - 1} \neg \left(\beta(\mathrm{T}[i+j])_k \oplus \beta(\mathrm{P}[j])_k \right) \right\rangle \right) |0\rangle \\
&\rightarrow \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \left(\left(\otimes_{j=0}^{M-1} \otimes_{k=0}^{\lceil \log(|\Sigma|) \rceil - 1} |\neg \left(\beta(\mathrm{T}[i+j])_k \oplus \beta(\mathrm{P}[j])_k \right) \rangle \right) \right. \\
&\qquad \left. \otimes_{j=0}^{M-1} \left| \wedge_{k=0}^{\lceil \log(|\Sigma|) \rceil - 1} \neg \left(\beta(\mathrm{T}[i+j])_k \oplus \beta(\mathrm{P}[j])_k \right) \right\rangle \right) \\
&\qquad \left| 0 \oplus \left(\wedge_{j=0}^{M-1} \wedge_{k=0}^{\lceil \log(|\Sigma|) \rceil - 1} \neg \left(\beta(\mathrm{T}[i+j])_k \oplus \beta(\mathrm{P}[j])_k \right) \right) \right\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \left(\left(\otimes_{j=0}^{M-1} \otimes_{k=0}^{\lceil \log(|\Sigma|) \rceil - 1} |\neg \left(\beta(\mathrm{T}[i+j])_k \oplus \beta(\mathrm{P}[j])_k \right) \rangle \right) \right. \\
&\qquad \left. \otimes_{j=0}^{M-1} \left| \wedge_{k=0}^{\lceil \log(|\Sigma|) \rceil - 1} \neg \left(\beta(\mathrm{T}[i+j])_k \oplus \beta(\mathrm{P}[j])_k \right) \right\rangle \right) \\
&\qquad \otimes \left| \left(\wedge_{j=0}^{M-1} \wedge_{k=0}^{\lceil \log(|\Sigma|) \rceil - 1} \neg \left(\beta(\mathrm{T}[i+j])_k \oplus \beta(\mathrm{P}[j])_k \right) \right) \right\rangle
\end{aligned}$$

where the state of the preceeding $M \log(|\Sigma|)$ qubits of the scratch register are omitted for brevity.

Lastly, a single Pauli-Z operator will operate on the output register. The operation of the Z operator on the output register can be written as

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \left| \left(\wedge_{j=0}^{M-1} \wedge_{k=0}^{\lceil \log(|\Sigma|) \rceil - 1} \neg \left(\beta(\mathrm{T}[i+j])_k \oplus \beta(\mathrm{P}[j])_k \right) \right) \right\rangle \tag{4.4}$$

$$\rightarrow \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle (-1)^{\left(\wedge_{j=0}^{M-1} \wedge_{k=0}^{\lceil \log(|\Sigma|) \rceil - 1} \neg \left(\beta(\mathrm{T}[i+j])_k \oplus \beta(\mathrm{P}[j])_k \right) \right)} \left| \left(\wedge_{j=0}^{M-1} \wedge_{k=0}^{\lceil \log(|\Sigma|) \rceil - 1} \neg \left(\beta(\mathrm{T}[i+j])_k \oplus \beta(\mathrm{P}[j])_k \right) \right) \right\rangle \tag{4.5}$$

$$\tag{4.6}$$

We can write the state of the index register in this step as

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (-1)^{\left(\wedge_{j=0}^{M-1} \wedge_{k=0}^{\lceil \log(|\Sigma|) \rceil - 1} \neg \left(\beta(\mathrm{T}[i+j])_k \oplus \beta(\mathrm{P}[j])_k \right) \right)} |i\rangle \tag{4.7}$$

where the states of the substring, pattern, scratch and output registers are not shown.

4.7 Correctness

Lemma 4.7.1. *Let Σ be an alphabet, $T \in \Sigma^N$, $P \in \Sigma^M$ and suppose*

$$H(T[i_x : i_x + M - 1], P) = 0$$

for $0 \leq i_x < N$. Then

$$U_g\left(\alpha_{i_x}|i_x\rangle\right) \rightarrow -\alpha_{i_x}|i_x\rangle$$

and

$$U_g\left(\alpha_i|i\rangle\right) \rightarrow \alpha_i|i\rangle$$

for $0 \leq i, i_x < N, i \neq i_x$.

Proof. Let Σ be an alphabet, $\mathrm{T} \in \Sigma^N$, $\mathrm{P} \in \Sigma^M$ and suppose

$$H(\mathrm{T}[i_x : i_x + M - 1], \mathrm{P}) = 0 \tag{4.8}$$

for $0 \leq i_x < N$. Given the initial superposition state of an index register,

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle$$

which encodes the indices in T, the operation of the operator U_g on the index register will transform its state such that

$$U_g : U_g\left(\frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle\right) \rightarrow \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (-1)^{\left(\wedge_{j=0}^{M-1} \wedge_{k=0}^{\lceil \log(|\Sigma|)\rceil - 1} \neg(\beta(\mathrm{T}[i+j])_k \oplus \beta(\mathrm{P}[j])_k)\right)} |i\rangle$$

Equation 4.8 implies that

$$\beta(\mathrm{T}[i_x + j])_k = \beta(\mathrm{P}[j])_k, \quad \forall\ 0 \leq j < M,\ 0 \leq k < \lceil \log(|\Sigma|)\rceil \tag{4.9}$$

Equation 4.9 implies that

$$\beta(\mathrm{T}[i_x + j])_k \oplus \beta(\mathrm{P}[j])_k = 0, \quad \forall\ 0 \leq j < M,\ 0 \leq k < \lceil \log(|\Sigma|)\rceil \tag{4.10}$$

Equation 4.10 implies that

$$\wedge_{j=0}^{M-1} \wedge_{k=0}^{\lceil \log(|\Sigma|)\rceil - 1} \neg\left(\beta(\mathrm{T}[i_x + j])_k \oplus \beta(\mathrm{P}[j])_k\right) = 1 \tag{4.11}$$

Equation 4.11 implies that

$$(-1)^{\left(\wedge_{j=0}^{M-1}\wedge_{k=0}^{\lceil\log(|\Sigma|)\rceil-1}\neg(\beta(\mathrm{T}[i_x+j])_k\oplus\beta(\mathrm{P}[j])_k)\right)}=-1 \tag{4.12}$$

Following the same argumentation, the premise

$$H(\mathrm{T}[i:i+M-1],\mathrm{P})\neq 0 \tag{4.13}$$

for $0\leq i<N, i\neq i_x$ will imply

$$(-1)^{\left(\wedge_{j=0}^{M-1}\wedge_{k=0}^{\lceil\log(|\Sigma|)\rceil-1}\neg(\beta(\mathrm{T}[i_x+j])_k\oplus\beta(\mathrm{P}[j])_k)\right)}=1 \tag{4.14}$$

Then,

$$U_g\left(\alpha_{i_x}|i_x\rangle\right)\rightarrow-\alpha_{i_x}|i_x\rangle$$

and

$$U_g\left(\alpha_i|i\rangle\right)\rightarrow\alpha_i|i\rangle$$

for $0\leq i,i_x<N, i\neq i_x$. □

4.8 Complexity

4.8.1 Time

We compute the time complexity of operator U_g with respect to its corresponding quantum circuit. The first group of $M\lceil\log(|\Sigma|)\rceil$ CNOT operators which act on the qubits of the substring and scratch register will execute in a single time step since the target qubits of the operators in the scratch register are mutually exclusive. Likewise, the second group of $M\lceil\log(|\Sigma|)\rceil$ CNOT operators which act on the qubits of the pattern and scratch register will execute on mutually exclusive qubits of the scratch register and so will require a single time step for execution, $O(1)$.

A singe C^kNOT operator can be decomposed into a sequence of $5k-4$ 2-qubit unitary operators [**Chuang2000**] with time complexity $O\left(k\right)$. The third group of M $C^{\lceil\log(|\Sigma|)\rceil}NOT$ operators can then be decomposed into M $5\lceil\log(|\Sigma|)\rceil-4$ 2-qubit unitary operators. Since the target of these multiple-control operators are mutually exclusive, their operation can be carried out in parallel and thus will have time complexity in $O(\log(|\Sigma|))$. Likewise, the succeeding C^MNOT operator can be decomposed into $5M-4$ 2-qubit unitary operators and thus will have time complexity in $O(M)$.

Lastly, the operation of the Pauli-Z operator on the output register will be executed in single time step, $O(1)$. The total time complexity of the U_g operator will then be in

$$O(\log(|\Sigma|))+O(M)\in O(\log(|\Sigma|)+M)$$

4.8.2 Space

In the operation of operator U_g, we introduce the auxiliary registers substring, pattern, scratch and output register in addition to the index register. The index register is composed of $\lceil \log(|\Sigma|) \rceil$ qubits for representing indices i in T. The substring and pattern register will require $M\lceil \log(|\Sigma|) \rceil$ qubits each for encoding the substrings in T_s and the pattern P. The scratch register will have the size of $M\lceil \log(|\Sigma|) \rceil + M$ for encoding the results of comparison of each symbol in P and each symbol in the substrings in T_s. Lastly, the output register will only have a single qubit for encoding the final result of the comparisons. The total space complexity of the operation of U_g will then be

$$O(\log(|\Sigma|)) + O(M \log(|\Sigma|)) + O(M \log(|\Sigma|) + M) \in O(M \log(|\Sigma|))$$

4.8.3 Gate

In the left half of the quantum circuit for operator U_g in Figure **??**, the first and second set of operators will be composed of $2M\lceil \log(|\Sigma|) \rceil$ CNOT operators. The third set of operators composed of M $C^{\lceil \log(|\Sigma|) \rceil}NOT$ can be decomposed into $5M\lceil \log(|\Sigma|) \rceil - 4M$ 2-qubit unitary operators [**Chuang2000**]. Likewise, the next operator, $C^M NOT$, can be decomposed into $5M - 4$ 2-qubit unitary operators. Lastly, the quantum circuit will include only a single Pauli-Z operator to operate on the output register. All the operators in the left half of the quantum circuit for U_g, except the Z operator, will have their counterpart in the right half of the quantum circuit for resetting the scratch register into its initial state. Thus, the quantum circuit for operator U_g will have a gate complexity of

$$2(7M\lceil \log(|\Sigma|) \rceil + M - 4) + 1 \in O(M\lceil \log(|\Sigma|) \rceil)$$

2-qubit unitary operators.

Lemma 4.8.1. *Let Σ be an alphabet, $T \in \Sigma^N$ and $P \in \Sigma^M$. Suppose*

$$H\left(T[i_x : i_x + M - 1], P\right) = 0$$

for $0 \leq i_x < N$. Furthermore, suppose the oracle in Grover's quantum search algorithm is replaced with operator U_g. Then Grover's quantum search algorithm outputs index i_x in time complexity $O\left(\sqrt{N}\left(\log(|\Sigma|) + M\right)\right)$ with probability $\left|\frac{1}{2}\left(\left(\lambda_0^k - \lambda_1^k\right)\sqrt{\frac{1}{N} - 1} + \left(\lambda_0^k + \lambda_1^k\right)\frac{1}{\sqrt{N}}\right)\right|^2$ where

$$\begin{aligned} k &= \left\lfloor \frac{\pi}{4}\sqrt{N} \right\rfloor \\ \lambda_0 &= 1 - \frac{2}{N} - \frac{2}{N}\sqrt{1 - N^2} \\ \lambda_1 &= 1 - \frac{2}{N} + \frac{2}{N}\sqrt{1 - N^2} \end{aligned}$$

Proof. Let Σ be an alphabet, $\mathrm{T} \in \Sigma^N$ and $\mathrm{P} \in \Sigma^M$. Suppose

$$H\left(\mathrm{T}[i_x : i_x + M - 1], \mathrm{P}\right) = 0$$

for $0 \leq i_x < N$. Furthermore, suppose the oracle in Grover's quantum search algorithm is replaced with operator U_g. Operator U_g will be called $\frac{\pi}{4}\sqrt{N} \in O\left(\sqrt{N}\right)$ times to optimize the amplification of the probability of occurrence of the marked state $|i_x\rangle$. Each call to operator U_g will take

$$O\left(\log(|\Sigma|) + M\right)$$

time complexity. Then Grover's quantum search algorithm outputs index i_x in time complexity $O\left(\sqrt{N}\left(\log(|\Sigma|) + M\right)\right)$ with probability $\left|\frac{1}{2}\left(\left(\lambda_0^k - \lambda_1^k\right)\sqrt{\frac{1}{N} - 1} + \left(\lambda_0^k + \lambda_1^k\right)\frac{1}{\sqrt{N}}\right)\right|^2$ where

$$\begin{aligned} k &= \left\lfloor \frac{\pi}{4}\sqrt{N} \right\rfloor \\ \lambda_0 &= 1 - \frac{2}{N} - \frac{2}{N}\sqrt{1 - N^2} \\ \lambda_1 &= 1 - \frac{2}{N} + \frac{2}{N}\sqrt{1 - N^2} \end{aligned}$$

□

4.9 Simulation

4.9.1 Increasing N

4.9.2 Increasing M

4.9.3 Increasing $|\Sigma|$

Chapter 5

Convolution-based algorithm for approximate string matching

Chapter 6

Filtering-based algorithm for approximate string matching

Part III

Final Remarks

Chapter 7

Conclusions