Quantum Advantage Even Without Entanglement

Gil Ratsaby

Technion - Computer Science Department - M.Sc. Thesis MSC-2006-28 - 2006

Quantum Advantage Even Without Entanglement

Research Thesis

Submitted in Partial Fulfillment of The Requirements
for the Degree of Master of Science in Computer Science

Gil Ratsaby

Submitted to the Senate of
the Technion — Israel Institute of Technology
Sivan, 5766 Haifa May 2006

This Research Thesis Was Done Under The Supervision of Dr. Tal Mor in the Computer Science Department.

I would like to thank Dr. Tal Mor for his guidance and support throughout the course of this research. I would also like to thank Dan Kenigsberg, Zvi Devir and Netanel Ratner for fruitful discussions and help in disentanglement of problems.

The Generous Financial Help Of the Technion Is Gratefully Acknowledged.

Contents

List of Figures

Abstract

Two main motivations inspired this work. The first was to explore the role of entanglement in quantum computation, and more specifically whether significant quantum advantage over classical computation can exist without entanglement. The second was to explore the existence of new problems for which there is a quantum advantage, and the role of entanglement in them.

We focused mainly on the case of exact pure-state quantum computing with oracles and promises, and obtained the following results: for the Deutsch-Jozsa problem we find the *maximal* subproblem that can be solved without entanglement, and show that the Deutsch-Jozsa algorithm still has a non-negligible advantage over the best classical algorithm. We show that this subproblem is of greater significance, by proving that it contains *all* the Boolean functions whose quantum phase-oracle is non-entangling.

This approach is then generalized: we show that a Deutsch-Jozsa-like promise problem exists for any Boolean function, and find those that have an entanglement-free subproblem. We then present a method for constructing a large amount of new promise problems for which there is a quantum advantage, and investigate entanglement-free subproblems.

Chapter 1

Introduction

The incorporation of quantum theory into computer science has introduced new non-classical tools to the world of computation and communication. Clever usage of these tools led to some spectacular results: Shor's factorization algorithm [29], Grover's quantum search algorithm [15], Bennett-Brassard's quantum key distribution [2] and quantum teleportation [3], all extend beyond widely-believed bounds.

In the following we briefly review several topics in computer science and quantum information theory, providing the basis for our work.

1.1 Classical computation

Computability and *Complexity* theories deal with two fundamental questions with regard to computational tasks: what tasks can be performed in principle on a computer, and how much resources are required to perform a given task. In order to give meaningful answers to these questions, we must define a clear and general model for the concept of computation. Although there are many models that formalize the notion of a (classical) algorithm, they have all been shown to be equivalent, in the sense that any task that can be performed on one model can also be carried out on any of the other models. This fact led to the famous Church-Turing thesis: *all "reasonable" models of computation are equivalent.*

Computability theory deals with the question of what problems can be solved in principle on a computer. One may imagine that given enough time and memory (conventionally termed *space*), a modern computer can

perform any computational task. However, it is known that many problems, even seemingly simple ones, can not be solved on a computer. An example for such a problem is the *halting problem*: given an algorithm and an input to it, will the algorithm stop? It was shown by Turing [31] that the halting problem can not be solved in general, no matter how much resources are available.

Given a solvable problem, it is natural to ask how much resources are required in order to solve it. The theory of computational complexity classifies computational problems into classes according to the amount of resources required to solve them, and studies the relations between these classes. In complexity terms, we consider a polynomial-time algorithm[1] as an efficient one, while an algorithm that requires exponential resources is considered inefficient. Although this definition may sometimes be coarse, it usually points out the practically inefficient algorithms with much success.

It is customary to formulate computational problems as decision problems, i.e., problems with only two possible answers, *yes* and *no*. For example, the *primality* problem is the problem of deciding whether a given integer is a prime number or not. Decision problems are conveniently represented by languages, where a language L over an alphabet Σ is the set of strings $L \subseteq \Sigma^*$ for which the answer of the problem is *yes*. We classify languages into complexity classes according to the amount of resources needed to decide them, where the most treated resource is *time*, represented by the number of computational steps. The class of languages decided deterministically in polynomial time is denoted by **P**. The class of languages decided probabilistically in polynomial time is denoted by **BPP** and is considered the class of problems efficiently solvable on a computer. Another class of great importance is the class of languages that can be verified polynomially, i.e., it is possible to verify in polynomial time that an input string belongs to the language, given a *witness* — some extra information represented by a polynomially-long string. This class, denoted[2] by **NP**, contains many im-

[1]Note that when we describe an algorithm as *polynomial-time* (or sometimes just *polynomial*), we refer to its asymptotic running time as a function of input length. See [25] for a broader introduction to computational complexity.

[2]The reason for this notation is that this class of languages was proven equal to the class of languages decided non-deterministically in polynomial-time. In this context, a non-deterministic computation is an abstract non-realistic procedure in which many computations are carried out in parallel, and the right one (if exists) is chosen correctly. For simplicity we prefer the above definition of **NP**. For more details see [25].

portant problems for which no efficient (i.e. polynomial) algorithm is known, such as *Satisfiability*, *Hamiltonian cycle* and *factoring* [25, 24]. One of the most important open questions in computer science is whether **P**=**NP**.

1.1.1 Oracles

It is possible to define computation models in which during its execution, an algorithm may ask whether certain strings are members of a language, and instantly receive the correct answers. We may imagine that the algorithm has access to an *oracle* which responds to *queries* by supplying the correct answers. For example, an algorithm may ask whether certain numbers are primes, or whether certain boolean formulas are satisfiable. The notation ${C_1}^A$ denotes the class of languages solved by algorithms with complexity C_1 but with access to an oracle A. The notation ${C_1}^{C_2}$ denotes the class of languages solved by algorithms with complexity C_1 but with access to oracles of languages from C_2. For example, the class P^{NP} denotes the class of languages solvable in polynomial time using an oracle of a language in NP.

It is known that there exists an oracle A for which $P^A = NP^A$, and there exists an oracle B for which $P^B \neq NP^B$, thus results about oracles do not supply immediate results about the base classes. However, oracles are useful as "research tools": many important proof techniques transcend to the oracle case. Thus results of the type ${C_1}^A \neq {C_2}^A$ are an indication that such proof techniques can not be used to prove $C_1 = C_2$.

1.2 Quantum computation

Although classical computability and complexity theories are mathematical theories, they are based on human intuitions about the physical world. A Turing machine [31] — a common model of computation — can almost be seen as a physical object, and so are many other models. Our intuition is based on our every-day experiences which are (mostly) classical in the physical sense. Since we know that classical physics is only an approximation of real-world behavior, it makes sense to review the theory of computation in light of quantum theory.

1.2.1 Qubits

We may think of classical computing in terms of bit strings and operations on them, where a single-bit operation can either flip a bit or not change it. The analogous concept for a classical bit in quantum computation is the quantum bit — the *qubit*. Mathematically, the state of a qubit is not limited to just 0 and 1 (denoted in quantum information theory as $|0\rangle$ and $|1\rangle$), but can be a linear combination, or a *superposition* of these two states. The general state of a qubit is $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$.

Qubits are based on 2-state quantum systems, where their state is described in quantum mechanics by unit vectors in 2-dimensional Hilbert space. $|0\rangle$ and $|1\rangle$ denote the basis states, and the *standard basis* is composed of the vectors

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Another common basis is $|+\rangle = \frac{(|0\rangle+|1\rangle)}{\sqrt{2}}$ and $|-\rangle = \frac{(|0\rangle-|1\rangle)}{\sqrt{2}}$, or in vector notation:

$$|+\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \qquad |-\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix}$$

We say that we *measure the qubit's state in a certain basis*. Such a measurement yields one of the two results in our classical measuring tool. For example, if we measure in the standard basis, we obtain the results "0" and "1" corresponding to the basis states $|0\rangle$ and $|1\rangle$, where "0" is measured with probability $|\alpha|^2$, and "1" with probability $|\beta|^2$.

Quantum theory also gives the description of the state of an n-qubit system. An n-qubit system is a unit vector in a 2^n dimensional Hilbert space. The general state of such a system is $\sum_{x=0}^{2^n-1} \alpha_x|x\rangle$, where $|x\rangle$ are basis states, and α_x are complex coefficients such that $\sum_{x=0}^{2^n-1} |\alpha_x|^2 = 1$. The standard basis for an n-qubit system is obtained when each qubit is either in the state $|0\rangle$ or $|1\rangle$ of the 1-qubit standard basis. Here too, a measurement of the state may only yield a basis state, where the state $|x\rangle$ is measured with probability $|\alpha_x|^2$. When we have several qubits, each in a given 1-qubit state, quantum theory tells us how to describe the joint system. The state of such a system is the tensor product of the qubits' states. For example, the state of a 2-qubit system composed of 2 qubits, both in the state $|0\rangle$ of the standard basis, is given by $|0\rangle \otimes |0\rangle = |00\rangle = (1\,0)^T \otimes (1\,0)^T = (1\,0\,0\,0)^T$, which is

4-dimensional unit vector as required. Such states are called *tensor-product states*.

1.2.2 Quantum gates

Any non-trivial computation involves an input and computational steps to produce a certain output. The theory of quantum mechanics tells us how quantum systems may change over time. We use these rules to describe the required changes in the quantum register state, such that the desired output will be measured.

Any change of a quantum state can be described by the application of a unitary transformation to it. In theory, any unitary operator specifies a valid quantum operator. We refer to quantum operators as quantum gates, in analogy to classical gates that perform transformations on classical registers. The *Hadamard* gate is a useful 1-qubit gate:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{1.1}$$

which transforms the standard basis states as follows:

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{1.2}$$

$$H|1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \tag{1.3}$$

A simple example of a 2-qubit quantum gate is the $C-NOT$ gate, described by the following matrix:

$$U_{C-NOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{1.4}$$

This gate performs the *controlled-not* operation on the 2-qubit standard basis states:

$$\begin{array}{ll} U_{C-NOT}|00\rangle = |00\rangle, & U_{C-NOT}|01\rangle = |01\rangle, \\ U_{C-NOT}|10\rangle = |11\rangle, & U_{C-NOT}|11\rangle = |10\rangle. \end{array} \tag{1.5}$$

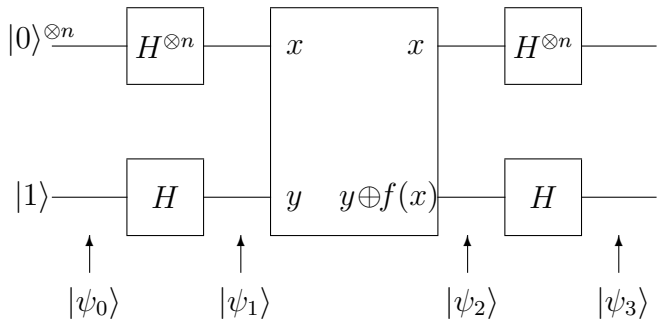

Figure 1.1: The Deutsch-Jozsa algorithm: a quantum subroutine common to Simon, Grover, Bernstein-Vazirani and other algorithms
אלגוריתם דויטש-ג'וזסה: פרוצדורה קוונטית המשותפת לאלגוריתמי סיימון, גרובר, ברנשטיין-וזירני ואחרים.

For classical circuits there are finite sets of gates that are universal in the sense that they can be used to construct any Boolean function. There are also (finite) universal sets for the continuous quantum case. These sets allow approximation to arbitrary accuracy of any unitary operation [5, 24].

1.2.3 The Deutsch-Jozsa algorithm

Let f be a Boolean function $f : \{0,1\}^n \to \{0,1\}$, with a promise that f is either *constant* or *balanced*, namely, the value of f is either the same for all the members in its domain, or it is 1 for exactly half of it and 0 for the other half. The function f is given as an oracle, where the quantum oracle is described by the unitary transformation $|x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$, for $x \in \{0,1\}^n$, $y \in \{0,1\}$. Our goal is to discover whether it is constant or balanced. Note that a deterministic classical algorithm that solves this problem must perform $2^{n-1}+1$ oracle queries. The Deutsch-Jozsa algorithm [13], represented by the quantum circuit in Figure 1.1, distinguishes the two possible types of f using only one query of the quantum oracle.

The quantum register is changed by the algorithm steps as follows:

1. The initial state is $|\psi_0\rangle = |0\rangle^{\otimes n}|1\rangle$.

2. Hadamard gate is applied to each qubit (for n qubits denoted by $H^{\otimes n}$):

$$|\psi_1\rangle = \sum_{x\in\{0,1\}^n} \frac{|x\rangle}{\sqrt{2^n}}[\frac{|0\rangle - |1\rangle}{\sqrt{2}}]. \tag{1.6}$$

3. Applying the quantum oracle query yields:

$$|\psi_2\rangle = \sum_{x\in\{0,1\}^n} \frac{(-1)^{f(x)}|x\rangle}{\sqrt{2^n}}[\frac{|0\rangle - |1\rangle}{\sqrt{2}}]. \tag{1.7}$$

4. Finally, after applying the last $n+1$ Hadamard gates:

$$|\psi_3\rangle = \sum_{z\in\{0,1\}^n} \sum_{x\in\{0,1\}^n} \frac{(-1)^{x\cdot z+f(x)}|z\rangle}{2^n}[\frac{|0\rangle - |1\rangle}{\sqrt{2}}]. \tag{1.8}$$

The amplitude of $|0\rangle^{\otimes n}|1\rangle$ in $|\psi_3\rangle$ is $\sum_x (-1)^{f(x)}/2^n$. Thus, if f is constant we obtain this state with certainty when measuring $|\psi_3\rangle$, and if f is balanced it is certain that some other state is measured. It follows that after just one query to the oracle, we are able to determine with certainty whether f is constant or balanced.

1.2.4 Entanglement

Multi-qubit systems may have non-tensor-product states, i.e., states that are not a tensor product of two subsystems states. Such a state is called *entangled*.

As an example for entanglement, suppose we have two qubits in the states $|\psi_1\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$ and $|\psi_2\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$. Their tensor product is $\alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|00\rangle + \beta_1\alpha_1|00\rangle + \beta_1\beta_2|00\rangle$. A closer look at this product shows it is far from general: the multiplication of the first and last terms, and that of the second and third are equal, which is of course not required in general. Since the only restriction for general 4-state systems is that the squares of their amplitudes in a certain state would sum up to 1, then clearly the set of tensor product states is a subset of all possible states. Thus the following state is entangled:

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

What makes entangled states so special? The answer to that was given by Bell, with his famous inequality. Consider first a system with classical correlations, e.g., suppose we have a box containing a red ball and a blue ball. One ball is chosen randomly, wrapped and sent to Alice, and the other is wrapped and sent to Bob, which is very far from Alice. In some sense, Alice and Bob's balls now have no defined state of their own, but only a joint state: red-blue with probability $\frac{1}{2}$ and blue-red with probability $\frac{1}{2}$. However, our classical logic tells us that each of the balls is in a defined state (i.e. it is either red or blue), and all we need to do is unwrap it and check it's color. We take it for granted that the ball is in a well defined state, independently of whether we check it or not. As we shall see, quantum systems may have states with subsystems that *really* have no defined state of their own, which leads to correlations that are classically impossible.

Imagine we prepare (in a repeatable manner) two particles, and send one particle to Alice and one to Bob, who are very far apart. Alice has two possible measurements of physical properties she may choose to perform on her particle. The results of these measurements are denoted by Q and R, and may have values of either 1 or -1. Bob has similar measurements, with outcomes denoted by S and T, and values of either 1 or -1. Alice and Bob perform their measurements at the same time, and they are far enough from each other that their measurements cannot disturb each other's results.

Consider the quantity:

$$QS + RS + RT - QT = (Q + R)S + (R - Q)T.$$

Following the two-ball classical logic, we assume that if Alice chooses to measure the Q property, the R property has a defined value (which is not measured) and vice versa. It follows that it must hold that either $(Q+R) = 0$ or $(R - Q) = 0$, and $QS + RS + RT - QT = \pm 2$. Let $p(q, r, s, t)$ be the probability that before measuring the system has the properties: $Q = q$, $R = r$, $S = s$, and $T = t$, where the probabilities may depend on system preparation and experimental noise. We thus have:

$$\begin{aligned} E(QS + RS + RT - QT) &= \sum_{qrst} p(q, r, s, t)(qs + rs + rt - qt) \\ &\leq \sum_{qrst} p(q, r, s, t) \cdot 2 = 2 \end{aligned}$$

On the other hand we have:

$$\begin{aligned} E(QS+RS+RT-QT) &= \sum_{qrst} p(q,r,s,t)qs + \sum_{qrst} p(q,r,s,t)rs \\ &+ \sum_{qrst} p(q,r,s,t)rt - \sum_{qrst} p(q,r,s,t)qt \\ &= E(QS)+E(RS)+E(RT)-E(QT). \end{aligned}$$

We thus obtain the Bell inequality:

$$E(QS)+E(RS)+E(RT)-E(QT) \leq 2. \tag{1.9}$$

Consider now a 2-qubit quantum system in the state:

$$|\psi\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}.$$

The first qubit is sent to Alice and the second to Bob, who perform the measurements corresponding to the following observables[3]:

$$Q = Z, \quad R = X$$

$$S = \frac{-Z-X}{\sqrt{2}}, \quad T = \frac{Z-X}{\sqrt{2}}.$$

where

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

The expected value of QS can be found by:

$$\begin{aligned} E(QS) &= \frac{1}{2}(\langle 01| - \langle 10|)\frac{-ZZ-ZX}{\sqrt{2}}(|01\rangle - |10\rangle) \\ &= \frac{1}{2\sqrt{2}}(\langle 01| - \langle 10|)(|01\rangle - |10\rangle + -|00\rangle - |11\rangle) = \frac{1}{\sqrt{2}}. \end{aligned}$$

Similarly, we get:

$$E(RS) = \frac{1}{\sqrt{2}}, E(RT) = \frac{1}{\sqrt{2}}, E(QT) = -\frac{1}{\sqrt{2}}.$$

[3]See [24] for an elaboration on quantum measurements and observables.

Which gives:

$$E(QS) + E(RS) + E(RT) - E(QT) = 2\sqrt{2},$$

in contradiction to equation 1.9! It follows that something[4] in our perception of the physical world is inconsistent with quantum theory, and in particular with entanglement. However, it has been shown that indeed equation 1.9 doesn't hold experimentally. This strange phenomenon of quantum mechanics is the source of many surprising outcomes in quantum computation and information theories.

1.2.5 Entanglement in quantum computing

We have seen that entanglement is an utterly quantum phenomenon which has no classical counterpart. Entanglement appears in most of the great achievements of quantum information theory, from Shor's factorization algorithm [29] and other algorithms [15, 13, 27] to quantum teleportation [3], superdense coding [6], quantum error correction [28], and some quantum key distribution schemes [14, 4, 9]. It is unquestionable that entanglement is a key resource of quantum computation. In the words of Horodecki [16], entanglement is "the corner-stone of the quantum information theory". When no entanglement exists in a pure-state quantum algorithm (without the use of an oracle), the computation can be simulated efficiently and exactly using classical means [17]. Furthermore, when entanglement exists but its amount is bounded, Jozsa and Linden showed [18] that the computation can still be efficiently simulated classically by a coin-tossing algorithm. However, their work does not rule out significant advantage of quantum computation without entanglement (**QCWE**) in an oracle-based setting (e.g. for the Deutsch-Jozsa algorithm). It also does not rule out exponential advantage of mixed-state QCWE over probabilistic classical computation.

Some cases have been found where even without entanglement, quantum computation outperforms classical computation. Collins, Kim and Holton [12] solve the Deutsch-Jozsa (DJ) [13] problem without entanglement, but only for $n = 2$ bits, and prove that entanglement is required for any larger n; Braunstein and Pati [10] show that using pseudo-pure states, Grover's search

[4]Most physicists believe that this result means that either *realism* or *locality* are the improper assumptions we make. See [24] for further discussion on these concepts and the implications of Bell's inequality.

problem can be solved without entanglement for $n \leq 3$ bits more efficiently than classically; Lloyd [21] suggests an entanglement-free implementation of Grover's algorithm, but with exponential spatial complexity; Biham, Brassard, Kenigsberg and Mor [8] use a non-standard computation model, with a limitation on the number of allowed queries, to prove a tiny advantage for any n in the context of Deutsch-Jozsa's and Simon's [27] problems (with mixed states); Meyer [22] notes that Bernstein-Vazirani algorithm [7, 30] requires no entanglement.

1.3 Preliminaries

Let $f, f' : \{0,1\}^n \to \{0,1\}$ be boolean functions. The following conventions are used throughout the rest of this work:

1. Unless otherwise stated, discussed oracles are boolean functions oracles.
2. Denote by $\overline{f}$ the boolean function such that $\forall x \in \{0,1\}^n, \overline{f}(x) = 1 - f(x)$, i.e., the negative of f.
3. The *quantum oracle* of f is the black-box unitary operation which for an $(n+1)$-qubit input state $|x\rangle|y\rangle$, outputs $|x\rangle|y \oplus f(x)\rangle$.
4. The *quantum phase-oracle* of f, denoted by U_f, is the black-box unitary operation which for an n-qubit input state $|x\rangle$, outputs $(-1)^{f(x)}|x\rangle$. Note that when the last qubit is in the state $|-\rangle$, the quantum oracle becomes a quantum phase-oracle by disregarding this qubit.

1.4 The structure of this thesis

The rest of this report is organized as follows. Chapter 2 presents the maximal entanglement-free subproblem of the Deutsch-Jozsa problem, and shows that for this subproblem the Deutsch-Jozsa algorithm still has an advantage over the best classical algorithm. It is shown that this subproblem is of greater significance, by proving that it contains *all* the Boolean functions whose quantum phase-oracle is non-entangling [19]. Chapter 3 generalizes this approach, showing that a Deutsch-Jozsa-like promise problem and a corresponding entanglement-free subproblem exists for any boolean function [23]. Chapter 4 presents a method for constructing a huge amount of new

promise problems for which there is a quantum advantage, and investigates the existence of entanglement-free subproblems. We conclude by discussing the implications of these results and bringing some interesting open questions [23].

Appendix A provides simple proofs that no non-trivial subproblems of Simon's and Grover's problems can be solved by the corresponding algorithms without entanglement [19].

Chapter 2

Deutsch-Jozsa Problem without Entanglement

In this chapter we analyze the occurrence of entanglement in the execution of the Deutsch-Jozsa Algorithm. We present a restricted version of the problem, which is the maximal entanglement-free subproblem, and show that the algorithm is still advantageous over the best exact classical algorithm.

2.1 Deutsch-Jozsa Problem without Entanglement

Bernstein and Vazirani defined a promise problem [7] that can be solved by the Deutsch-Jozsa subroutine using a single oracle call. Classically, this problem requires n oracle calls. Later, Meyer [22] noted that entanglement is not generated during the execution of the algorithm on their problem. Since the BV promise set is a subset of the DJ promise set, a corollary from these results is that a particular subclass of the DJ problem requires no entanglement.

Our research followed a different route, aimed at finding the maximal entanglement-free set. Looking at the Deutsch-Jozsa algorithm, we note that the only step in which entanglement can be generated is the third step, where the f oracle is applied to the quantum register. We define the following restricted DJ promise problem: the function f is again either balanced or constant, but it is also promised to be ***non-entangling*** for the DJ algorithm, i.e., applying the oracle to $|\psi_1\rangle$ yields a separable state (the state $|\psi_2\rangle$).

In order to solve this promise problem we execute the original algorithm. Since this is a sub-problem of the original one, the algorithm's correctness is assured and one quantum query is enough to determine whether the function is balanced or constant. Note also that if there exist such non-entangling balanced functions (i.e., the constant functions are not the only possible non-entangling functions), classical algorithms cannot solve the problem with only one query, and so are inferior compared with the quantum one.

Definition 2.1.1 *Let f be the Boolean function $f : \{0,1\}^n \to \{0,1\}$ of the following form:*

$$f(x) = (a \cdot x) \oplus c, \tag{2.1}$$

where $a \in \{0,1\}^n$, $c \in \{0,1\}$ and $a \cdot x = \sum_j a_j x_j \bmod 2$ is the inner product modulo 2 of bit strings a and x. For a given a and c, we denote the corresponding function by $f_{\vec{a},c}$.

Definition 2.1.2 *Let $F^{\otimes}$ be the set of all Boolean functions of the form $f_{\vec{a},c}$.*

Proposition 2.1.3 *Any $f_{\vec{a},c} \in F^{\otimes}$ is either constant or balanced.*

Proof: First note that if $a = 00\cdots0$, $f_{\vec{a},c}$ is constant. For non-zero a the function $f_{\vec{a},c}(x)$ is balanced. This is clear since the inhomogeneous linear equation system $a \cdot x \oplus c = 0$ implies one linear constraint over an n-dimensional x, and therefore the solution space is $(n-1)$-dimensional. This means that out of 2^n possible xs, 2^{n-1} (half) are solutions. □

We now define a restricted version of the DJ problem, and show that the DJ algorithm uses no entanglement to solve it.

Definition 2.1.4 *Denote by $DJ^{\otimes}$ the "entanglement-free"[1] Deutsch-Jozsa problem, such that in addition to the original problem definition, we add the requirement that all functions in the promise are in $F^{\otimes}$.*

Proposition 2.1.5 *Entanglement is never generated when executing the Deutsch-Jozsa algorithm for $DJ^{\otimes}$.*

[1] Proposition 2.1.5 provides the justification for this name.

Proof: For a string $x = x_1 x_2 \cdots x_n$, we denote by J_x the support of x, i.e., the set of indexes of nonzero elements in x. Applying the oracle $f_{\vec{a},c}$ on $|\psi_1\rangle$ yields:

$$
\begin{aligned}
|\psi_2\rangle &= \sum_{x\in\{0,1\}^n} \frac{(-1)^{f_{\vec{a},c}(x)}|x\rangle}{\sqrt{2^n}} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] && (2.2)\\
&= \sum_{x\in\{0,1\}^n} \frac{(-1)^{c}(-1)^{x\cdot a}|x\rangle}{\sqrt{2^n}} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] && (2.3)\\
&= \frac{(-1)^c}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} (-1)^{\sum_{J_x} a_i}|x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] && (2.4)\\
&= \frac{(-1)^c}{\sqrt{2^n}} (|0\rangle + (-1)^{a_1}|1\rangle)\cdots(|0\rangle + (-1)^{a_n}|1\rangle) \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right], && (2.5)
\end{aligned}
$$

thus for any $f_{\vec{a},c}$, $|\psi_2\rangle$ is not entangled. □

Having established these properties of $DJ^{\otimes}$, we would like to show that it is maximal, in the sense that it includes any case of separable computation for the Deutsch-Jozsa algorithm.

Proposition 2.1.6 *Any function f which is non-entangling for the Deutsch-Jozsa algorithm, is in $F^{\otimes}$.*

Proof: Let $|\psi_2\rangle$ be as defined for the Deutsch-Jozsa algorithm. If f is non-entangling, then:

$$
\begin{aligned}
|\psi_2\rangle &= \sum_{x\in\{0,1\}^n} \frac{(-1)^{f(x)}|x\rangle}{\sqrt{2^n}} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] && (2.6)\\
&= e^{i\varphi_0} \bigotimes_{k=1}^{n} (\cos\theta_k|0\rangle + e^{i\varphi_k}\sin\theta_k|1\rangle) \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]. && (2.7)
\end{aligned}
$$

where φ_k, θ_k are reals. We compare the coefficients of Eqs. (2.6) and (2.7). First note that φ_0 must be 0 or π since the phase of the state $|00\cdots0\rangle$ is ±1. Suppose that for some $1 \le k \le n$, $|\cos\theta_k| \neq |\sin\theta_k|$. Then the coefficients of the states $|00\cdots000\cdots0\rangle$ and $|00\cdots010\cdots0\rangle$ where the 1 is on the kth position, are not of the same magnitude, in contradiction to Eq. 2.6. Similarly, if for some $1 \le k \le n$, $\varphi_k \notin \{0,\pi\}$, then the state $|00\cdots010\cdots0\rangle$ where

the 1 is on the kth position, has a complex coefficient, in contradiction. It follows that $|\psi_2\rangle$ is a product state of exactly the form of Eq. (2.5), which corresponds to a function $f_{\vec{a},c}$ as claimed. □

To establish the quantum advantage, we now show that solving $DJ^{\otimes}$ on an exact classical computer is not a trivial task.

Proposition 2.1.7 *The classical computational complexity of $DJ^{\otimes}$ is $\Theta(n)$.*

Proof: For a function of the form $f(x) = (a \cdot x) \oplus c$, one must know all the bits of a in order to check whether f is constant or balanced. Even one missing bit can determine the function's type either way. An oracle query of f yields an equation of the form $(a \cdot x) \oplus c = b$ for $b \in \{0, 1\}$, which is a linear Boolean equation in n variables: the n bits composing a. Therefore, in order to find a, one must consider at least n equations. This means that any classical algorithm will require at least n evaluations of f in order to exactly find a. We note that n queries are enough, since the evaluation of $f(2^k)$ yields the kth bit of a, so evaluating f on the n powers of 2 will determine it uniquely. □

Note that much like DJ, $DJ^{\otimes}$ can be classically solved with a constant number of oracle queries if errors are permitted.

2.2 General Phase Oracles and $DJ^{\otimes}$

We now show that the $DJ^{\otimes}$ problem is related to the whole set of separability-conserving quantum phase-oracles.

Definition 2.2.1 *An operation U is* separability-conserving*, if for any separable state $|\psi\rangle$, $U|\psi\rangle$ is separable.*

Proposition 2.2.2 *A quantum phase oracle U_f of a Boolean function f is separability conserving if and only if $f = f_{\vec{a},c} \in F^{\otimes}$.*

Proof: For the first direction assume that $f = f_{\vec{a},c} \in F^{\otimes}$. Note that the a general separable n-qubit state is of the form:

$$|\psi_{sep}\rangle = e^{i\varphi_0} \otimes_{k=1}^{n} (\cos\theta_k|0\rangle + e^{i\varphi_k} \sin\theta_k|1\rangle) \tag{2.8}$$

where $\varphi_0, \varphi_k, \theta_k$ are real numbers. Let $s_0(x)$ and $s_1(x)$ denote the sets of indexes for which $x \in \{0,1\}^n$ equals 0 and 1 respectively. It holds that

$$\begin{aligned} U_f|\psi_{sep}\rangle &= U_f e^{i\varphi_0} \otimes_{k=1}^{n} (\cos\theta_k|0\rangle + e^{i\varphi_k}\sin\theta_k|1\rangle) && (2.9)\\ &= U_f e^{i\varphi_0} \sum_{x=0}^{2^n-1} [\prod_{m\in s_0(x)} \cos\theta_m \prod_{m\in s_1(x)} e^{i\varphi_m}\cos\theta_m]|x\rangle && (2.10)\\ &= e^{i\varphi_0} \sum_{x=0}^{2^n-1} \prod_{m\in s_0(x)} \cos\theta_m \prod_{m\in s_1(x)} e^{i\varphi_m}\cos\theta_m (-1)^{(a\cdot x+c)}|x\rangle && (2.11)\\ &= e^{i\varphi_0}(-1)^c \otimes_{k=1}^{n} (\cos\theta_k|0\rangle + (-1)^{a_k} e^{i\varphi_k}\sin\theta_k|1\rangle) && (2.12) \end{aligned}$$

which is a separable state. Note that the last equality follows from the fact that the bits a_j that contribute to the coefficient of $|x\rangle$ are exactly those for which $x_j \neq 0$.

For the second direction, suppose that U_f is separability conserving. Thus applying it on any sparable state yields a separable state. In particular, applying it on the state $H^{\otimes n}|0\rangle$ (which is of course separable) yields a separable state. We note that these are exactly the conditions of proposition 2.1.6, thus the same proof holds for f, and $f \in F^{\otimes}$. □

In conclusion, we identified the *maximal* subset of the Deutsch-Jozsa problem, which is solved with one quantum query by the Deutsch-Jozsa algorithm *without entanglement*, while the best exact classical algorithm requires a linear number of calls. We showed that the significance of this subset reaches beyond the scope of the DJ problem: *any* non-entangling phase oracle is an oracle of a function from this subset. Note also that here the quantum-to-classical gap in the exact case diminishes from $O(2^n)$:$O(1)$ to $O(n)$:$O(1)$. This is the price we pay for not using entanglement.

We remark that it can now be seen that the function set of [7, 22] contains *exactly* half of the possible non-entangling functions.

2.3 Separable Implementation of the Oracle

It may be claimed that even though there is no entanglement after any step in the algorithm, the oracle must use entanglement during the computation of the function. We show here that there is an entanglement-free implementation for the oracle. Consider the following transformation:

$$|x\rangle \rightarrow (-1)^{c+1} \bigotimes_{i=1}^{n} (-1)^{a_i \cdot x_i} |x_i\rangle \tag{2.13}$$

It is easy to see that the tensor product on the right-hand side gives exactly the requested result of applying $f_{\vec{a},c}$ on input $|x\rangle$, i.e., $(-1)^{f_{\vec{a},c}}|x\rangle$. The oracle operation can be done locally, using n single-qubit transformations such as

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{ia_i\xi_i(t)} \end{pmatrix}, \tag{2.14}$$

where $\xi(t)$ ascends from 0 to 2π, maintaining separability even in continuous time.

Chapter 3

Detsch-Jozsa Problem for Any Boolean Function

In this chapter we show that for **any** boolean function there exists a DJ-like promise problem and a corresponding deterministic quantum algorithm which solves the problem exponentially faster than any deterministic classical algorithm. This means that instead of one problem (the DJ problem), we have 2^{2^n} equivalent problems. Following the problem $DJ^{\otimes}$ defined in the previous chapter, here we find 2^n problems for which similar quantum advantage exists without entanglement.

This wider look on the DJ problem will allow us to define new promise problems more comfortably in the next chapter.

3.1 Detsch-Jozsa Problem for Any Boolean Function

Let f, f' be boolean functions, $f, f' : \{0,1\}^n \to \{0,1\}$, and let U_f and U'_f be the corresponding quantum phase oracles.

Definition 3.1.1 *Denote the number of inputs $x \in \{0,1\}^n$ for which $f(x) \neq f'(x)$ by $d(f, f')$.*

Note that if we think of function values as a vector, then d is the Hamming distance between the value vectors of f and f'.

Definition 3.1.2 *Denote by $F_{\frac{1}{2}}(f)$ the set of boolean functions $f' : \{0,1\}^n \to \{0,1\}$, such that $d(f,f') = 2^{n-1}$.*

We now define a set of DJ-like promise problems, one for each Boolean function. Much like DJ, we are given an oracle as an input and we have to decide whether it has a certain property.

Definition 3.1.3 *The promise problem P_f is the problem of deciding whether a given oracle for a boolean function g is one of $\{f, \overline{f}\}$ or not, where it is promised that either $g \in \{f, \overline{f}\}$ or $g \in F_{\frac{1}{2}}(f)$.*

Note that for any boolean function f, P_f and $P_{\overline{f}}$ are exactly the same problem, since by definition 3.1.2, $f' \in F_{\frac{1}{2}}(f)$ if and only if $f' \in F_{\frac{1}{2}}(\overline{f})$.

Example 3.1.4 The Deutsch-Jozsa problem is P_f for $f \equiv 0$. f and $\overline{f}$ are the constant functions, and $F_{\frac{1}{2}}(f)$ is the set of functions that equal 1 on half their inputs and 0 for the other half, i.e. the balanced functions. □

Example 3.1.5 Consider the problem P_{AND}, where AND is the Boolean function of multiplying all the input bits[1]. $F_{\frac{1}{2}}(AND)$ is the set of functions that equal 1 on the input string $1 \cdots 11$ and on exactly 2^{n-1} more inputs, or are a negative of such a function. □

We now show that there is no efficient way to solve P_f classically. This is shown by formalizing the same basic arguments used in the DJ problem.

Proposition 3.1.6 *For any boolean function $f : \{0,1\}^n \to \{0,1\}$, any deterministic classical algorithm that solves P_f correctly requires $\Theta(2^n)$ oracle queries.*

Proof: Suppose a deterministic classical algorithm queries the oracle up to 2^{n-1} times on a set of inputs I_q, such that $|I_q| = q \leq 2^{n-1}$. Let f' be an oracle that equals f for all the inputs in I_q and on $2^{n-1} - q$ more arbitrary inputs, and equals $\overline{f}$ for the rest of the inputs. The algorithm *can not* distinguish f from f' deterministically without performing more queries, since both oracles yield exactly the same answers for its set of queries. By definition 3.1.2, $f' \in F_{\frac{1}{2}}(f)$, thus the algorithm must perform more than 2^{n-1} queries in order to solve P_f deterministically.

[1]Since P_f is defined for *any* boolean function, AND is just an arbitrary example.

However, since the functions in $F_{\frac{1}{2}}(f)$ differ from f for exactly 2^{n-1} inputs, it follows that $2^{n-1}+1$ queries suffice to answer P_f with certainty: if all $2^{n-1}+1$ queries comply with f, answer *yes*, otherwise answer *no*. Thus, any deterministic classical algorithm that solves P_f correctly requires $\Theta(2^n)$ oracle queries. □

To establish the quantum advantage, we show that any promise problem P_f can be solved very efficiently by a quantum procedure.

Proposition 3.1.7 *For any boolean function $f : \{0,1\}^n \to \{0,1\}$, the following quantum procedure solves P_f correctly.*
Procedure** M_f**:

1. *Given an oracle g, produce:*
$$|\psi_1\rangle = U_g H^{\otimes n}|0\rangle^{\otimes n}$$
2. *Apply $H^{\otimes n}U_f$ on $|\psi_1\rangle$:* [2]
$$|\psi_2\rangle = H^{\otimes n}U_f|\psi_1\rangle = H^{\otimes n}U_f U_g H^{\otimes n}|0\rangle^{\otimes n}$$
3. *Measure $|\psi_2\rangle$. If the resulting state is $|0\rangle^{\otimes n}$ answer "$g \in \{f, \overline{f}\}$", otherwise answer "$g \in F_{\frac{1}{2}}(f)$". Denote these answers $M_f(g) = 1$ and $M_f(g) = 0$ respectively.*

Proof: For the first direction, we note that both U_f and $H^{\otimes n}$ are their own reverse, and that $U_f U_{\overline{f}} = -I$. Thus if $g \in \{f, \overline{f}\}$, we have

$$|\psi_2\rangle = H^{\otimes n}U_f U_g H^{\otimes n}|0\rangle^{\otimes n} = \pm H^{\otimes n}H^{\otimes n}|0\rangle^{\otimes n} = \pm|0\rangle^{\otimes n}. \tag{3.1}$$

Assume now that $g \in F_{\frac{1}{2}}(f)$. It holds that

$$|\psi_2\rangle = H^{\otimes n}U_f U_g H^{\otimes n}|0\rangle^{\otimes n} = H^{\otimes n}U_f U_g \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle. \tag{3.2}$$

$$= H^{\otimes n}\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}(-1)^{f(x)+g(x)}|x\rangle. \tag{3.3}$$

$$= \frac{1}{2^n}\sum_{y=0}^{2^n-1}\sum_{x=0}^{2^n-1}(-1)^{y\cdot x+f(x)+g(x)}|y\rangle \tag{3.4}$$

[2] Note that f is given (it defines the problem), thus we may use U_f. We are only concerned with the query complexity of g, all other operations are ignored.

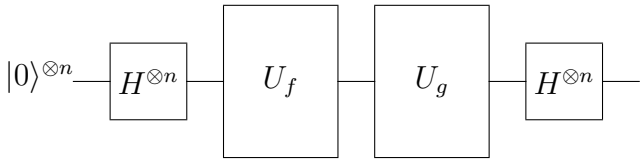

Figure 3.1: Procedure M_f. A circuit for solving the problem P_f using only one oracle query on the input oracle g.
פרוצדורה M_f. מעגל לפתרון הבעייה P_f המשתמש רק בקריאת אורקל יחידה על אורקל הקלט g.

Where $y \cdot x$ is the bitwise inner product of y and x. The coefficient of $|0\rangle^{\otimes n}$ is $\frac{1}{2^n}\sum_{x=0}^{2^n-1}(-1)^{f(x)+g(x)}$. From the promise on g, we have that half of the elements in this sum are 1 and the other half are -1. Thus the coefficient is 0 and the probability to measure $|0\rangle^{\otimes n}$ is 0. □

3.2 Quantum advantage without entanglement

In chapter 2 it was shown that by restricting the promise we may create a *subproblem* of the Deutsch-Jozsa problem, such that the Deutsch-Jozsa quantum algorithm uses *no entanglement*, yet it has an advantage over any classical algorithm ($O(1)$ vs. $O(n)$). In this section we show that the same holds for any boolean function f such that U_f itself does not create entanglement, i.e., there are 2^n $DJ^{\otimes}$-like problems for which quantum advantage exists without entanglement.

Recall our definition 2.1.2 of $F^{\otimes}$ which is the set of non-entangling functions. We wish to define a subproblem for problems of the type P_f, such that the promise is restricted to functions in $F^{\otimes}$. Observe that for any $f = f_{\vec{a},c}$ and $f' = f_{\vec{a}',c'}$ such that $a' \neq a$, it holds that $d(f, f') = 2^{n-1}$. To see this,

note that

$$\begin{aligned}(f(x)+f'(x))\bmod 2 &= (a\cdot x + a'\cdot x + c + c')\bmod 2 && (3.5)\\ &= (\sum_{i=1}^{n}(a_i+a'_i)x_i + c + c')\bmod 2 && (3.6)\\ &= ((a+a')\cdot x + (c+c'))\bmod 2, && (3.7)\end{aligned}$$

which by definition is a function in $F^{\otimes}$. When $a \neq a'$ this function is not constant, thus by proposition 2.1.3, it is balanced, which means that $d(f, f') = 2^{n-1}$.

Thus, for any $f \in F^{\otimes}$ we obtain that any $g \in F_{\frac{1}{2}}(f) \cap F^{\otimes}$ is a function in $F^{\otimes} \setminus \{f, \overline{f}\}$, namely, $F_{\frac{1}{2}}(f) \cap F^{\otimes} = F^{\otimes} \setminus \{f, \overline{f}\}$, which allows us to define:

Definition 3.2.1 *Let $f = f_{\vec{a},c} \in F^{\otimes}$. The promise problem $P_f^{\otimes}$ is the problem of deciding whether a given oracle for a boolean function g is one of $\{f, \overline{f}\}$ or not, where it is promised that either $g \in \{f, \overline{f}\}$ or $g \in F^{\otimes} \setminus \{f, \overline{f}\}$.*

Observe that the procedure M_f solves $P_f^{\otimes}$ correctly since we only restrict the set $F_{\frac{1}{2}}(f)$. An example for such a problem is the $DJ^{\otimes}$ problem, where it is promised that the given oracle is either constant or an oracle for $f_{\vec{a},c} \in F^{\otimes}$ for $a \in \{1, ..., 2^n - 1\}$ and $c \in \{0, 1\}$. We now prove the properties of such promise problems.

Proposition 3.2.2 *There is no entanglement in the execution of M_f on $P_f^{\otimes}$.*

Proof: From our definition of M_f (see prop. 3.1.7), the only operators that may be non-separable are the oracle operators. From the promise that $g, f \in F^{\otimes}$, proposition 2.2.2, and the fact the the initial state is $|0\rangle^{\otimes n}$, it follows that no entanglement is present in the execution of M_f on $P_f^{\otimes}$. □

We now show that $P_f^{\otimes}$ is maximal with respect to the procedure M_f, i.e., expanding the promise set of $P_f^{\otimes}$ will introduce entanglement to the circuit.

Proposition 3.2.3 *For function g which is non-entangling for the M_f algorithm, it holds that $g = g_{\vec{a},c} \in F^{\otimes}$.*

Proof: First note that we are promised that $f = f_{\vec{a'},c'}$ for some $a' \in \{0,1\}^n$ and $c' \in \{0,1\}$. Secondly, $U_g U_f = U_{g \oplus f}$, i.e., applying U_f and then U_g is equivalent to applying the phase-oracle of the function $g \oplus f$, which brings us back to the DJ case. Proposition 2.1.6 ensures us that $g \oplus f = f_{\vec{a''},c''}$ for some $a'' \in \{0,1\}^n$ and $c'' \in \{0,1\}$. It thus holds that

$$(f \oplus g)(x) = (a' \cdot x + c' + g(x)) \bmod 2 = (a'' \cdot x + c'') \bmod 2,$$

and thus

$$g(x) = ((a' + a'') \cdot x + (c' + c'')) \bmod 2 = (a \cdot x + c) \bmod 2,$$

as required.

□

Proposition 3.2.4 *For any boolean function $f \in F^{\otimes}$, any deterministic classical algorithm that solves $P_f^{\otimes}$ correctly requires $\Theta(n)$ oracle queries.*

Proof: Let a classical algorithm query g on a set of n-bit string inputs $S = \{x_1, x_2, ..., x_q\}$. The queries yield the equations $a \cdot x_i + c = b_i$, where $1 \leq i \leq q$, and b_i are the oracle answers. We thus have a set of q linear equations with n variables (the bits of a) over Z_2. If $q < n$, then there are at least two a values (i.e., two different functions in $F^{\otimes}$) that solve the system, thus the algorithm can't determine with certainty whether $g \in \{f, \overline{f}\}$ or not. It follows that in order to solve $P_f^{\otimes}$ correctly, a classical algorithm must query g at least n times.

On the other hand, n queries are enough: querying g on all the powers of 2 yields the n bits of a (or $\overline{a}$, depending on c), which solves the problem. □

Chapter 4

Building Promises

In this chapter we introduce a method for defining new promise problems. We start with a simple example to provide some intuition.

Example 4.1 Consider the following boolean functions over $\{0,1\}^n$. Let f_0 be the constant function $f_0 \equiv 0$, and f_{x_0} the function that equals the MSB bit of the input: $\forall x = x_0 x_1 ... x_{n-1} \in \{0,1\}^n$, $f_{x_0}(x) = x_0$. Note that $d(f_0, f_{x_0}) = 2^{n-1}$. Denote $F = \{f_0, f_{x_0}, \overline{f_0}, \overline{f_{x_0}}\}$, and denote by $F_{\frac{1}{2}}(f_0, f_{x_0})$ the set of all boolean functions f' such that $d(f_0, f') = d(f_{x_0}, f') = 2^{n-1}$, i.e., the set of functions that differ from **both** f_0 and f_{x_0} on 2^{n-1} inputs [1].

Given an oracle g, our promise problem is to decide whether $g \in F$ or not, where it is promised that either $g \in F$ or $g \in F_{\frac{1}{2}}(f_0, f_{x_0})$. We will later prove that a deterministic classical algorithm must query g more than 2^{n-2} times in order to solve the problem correctly. However, a simple quantum procedure solves the problem using only two queries on g:

1. Run[2] M_{f_0} on g.
2. Run $M_{f_{x_0}}$ on g.
3. If $M_{f_0}(g) = 1$ or $M_{f_{x_0}}(g) = 1$ answer "$g \in F$", otherwise answer "$g \in F_{\frac{1}{2}}(f_0, f_{x_0})$".

[1] Observe that the set $F_{\frac{1}{2}}(f_0, f_{x_0})$ is not empty. It contains functions that differ from f on exactly half of the inputs for which $f = f_{x_0}$, and half of the inputs for which $f \neq f_{x_0}$. For example, it contains the function $f_{x_{n-1}}$: the function that equals the LSB bit of the input.

[2] Recall proposition 3.1.7 for the definition of M_f.

The quantum procedure M_f determines with certainty whether a given oracle g is one of $\{f, \overline{f}\}$ or not, thus $g \in \{f_0, f_{x_0}, \overline{f_0}, \overline{f_{x_0}}\}$ if and only if one of the procedures M_{f_0} and $M_{f_{x_0}}$ answers positively. □

4.1 Constructing new Promises

In the following we generalize the approach presented in the above example. We present a large set of promise problems for which there exists an exponential quantum advantage.

Definition 4.1.1 *Denote the number of inputs $x \in S \subseteq \{0,1\}^n$ for which $f(x) \neq f'(x)$ by $d_S(f, f')$.*

Definition 4.1.2 *Let $f_1, f_2, ..., f_r$, $2 \leq r \leq n-1$, be boolean functions over $\{0,1\}^n$, where f_i is associated with a partition $S_1^i, S_2^i, ..., S_{2^{i-1}}^i$ of the input domain $\{0,1\}^n$ as follows:*

1. *f_1 is an arbitrary function, and $S_1^1 = \{0, 1, ..., 2^n - 1\}$, the set of all inputs.*
2. *Given f_i and its corresponding input sets $S_1^i, ..., S_{2^{i-1}}^i$, the function f_{i+1} is arbitrarily chosen from the following set of functions:*

$$F_{S_{\frac{1}{2}}}(f_i) = \{f' : \{0,1\}^n \to \{0,1\} \mid d_{S_j}(f_i, f') = 2^{n-i},\ 1 \leq j \leq 2^{i-1}\} \tag{4.1}$$

3. *The input sets associated with f_{i+1} are determined by those of f_i and the choice of f_{i+1}: for $1 \leq j \leq 2^{i-1}$, $S_{2j-1}^{i+1} = \{x \in S_j^i \mid f_i(x) = f_{i+1}(x)\}$ and $S_{2j}^{i+1} = \{x \in S_j^i \mid f_i(x) \neq f_{i+1}(x)\}$.*

The promise problem P_F is the problem of deciding whether an oracle for a given boolean function g is one of $F = \{f_1, f_2, ..., f_r, \overline{f_1}, \overline{f_2}, ..., \overline{f_r}\}$ or not, where it is promised that either $g \in F$ or $g \in F_{S_{\frac{1}{2}}}(f_r)$.

Intuitively, starting from any arbitrary boolean function f_1, we add functions to F by splitting in half all the input sets of the previous function, and flipping the function value of one of every two halves.

Lemma 4.1.3 *The following holds:*

1. *Let* $f_i \in F,\ 1 \leq i \leq r$ *and* $f' \in F_{S_{\frac{1}{2}}}(f_i)$. *Then* $d(f_k, f') = 2^{n-1}$, $1 \leq k \leq i$.

2. *For any* $f_i, f_j \in F,\ 1 \leq i < j \leq r$, *it holds that* $d(f_i, f_j) = 2^{n-1}$.

3. $|F_{S_{\frac{1}{2}}}(f_i)| = \left(\begin{array}{c} 2^{n-i+1} \\ 2^{n-i} \end{array} \right)^{2^{i-1}}$

Proof: Let $f_i \in F$, $1 \leq i \leq r$. For the first claim, note that functions $f' \in F_{S_{\frac{1}{2}}}(f_i)$ are created by flipping function values of half of every input set $S_j^i,\ 1 \leq j \leq 2^{i-1}$, associated with f_i. Consider a function f_k, $1 \leq k \leq i$, and the associated input sets $S_m^k,\ 1 \leq m \leq 2^{k-1}$. By the definition of the input sets, we have that

$$\begin{aligned} S_m^k &= S_{2m-1}^{k+1} \cup S_{2m}^{k+1} \\ &= S_{4m-3}^{k+2} \cup S_{4m-2}^{k+2} \cup S_{4m-1}^{k+2} \cup S_{4m}^{k+2} \\ &= \ldots \\ &= S_{2^{i-k}m-2^{i-k}+1}^{i} \cup \cdots \cup S_{2^{i-k}m}^{i}. \end{aligned} \tag{4.2}$$

From Eq. 4.2 and the construction of P_f we see that for the values of each S_j^i, either $f' = f_k$ or $f' = \overline{f_k}$. It follows that by flipping the function values for half of the inputs in each S_j^i, we are also flipping the function values for half of the inputs in each S_m^k. Thus, since $\{S_m^k\}$ are a partition of $\{0,1\}^n$, it holds that $d(f_k, f') = 2^{n-1}$.

The second claim follows directly from the proof of the first one, since f_{i+1} is chosen from $F_{S_{\frac{1}{2}}}(f_i)$, thus we add to F only functions that differ from the functions in F by half of all the inputs.

For the third claim, we note that f_i is associated with input sets S_j^i, $1 \leq j \leq 2^{i-1}$, each of size 2^{n-i+1}. $F_{S_{\frac{1}{2}}}(f_i)$ contains all the functions that differ from f_i on half of each set S_j^i. There are $\left(\begin{array}{c} 2^{n-i+1} \\ 2^{n-i} \end{array} \right)$ possibilities to choose half of each set, thus, since there are 2^{i-1} sets, we have that $F_{S_{\frac{1}{2}}}(f_i)$ contains $\left(\begin{array}{c} 2^{n-i+1} \\ 2^{n-i} \end{array} \right)^{2^{i-1}}$ different functions. □

Proposition 4.1.4 *Any classical deterministic algorithm requires at least* 2^{n-r} *oracle queries in order to solve* P_F *correctly.*

Proof: To solve P_F correctly, we have to decide whether $g \in F$ or $g \in F_{S_{\frac{1}{2}}}(f_r)$. Suppose a classical algorithm queries g on a set of inputs I_q, such that $|I_q| = q \leq 2^{n-r}$. The size of the input sets $S_j^r,\ 1 \leq j \leq 2^{r-1}$ is 2^{n-r+1}, therefore the algorithm may query the oracle on at most half of a set S_j^r. Let f' be a function that equals f on all the inputs in I_q, and on more inputs such that for each set S_j^r, $d_{S_j^r}(f_q, f') = 2^{n-r}$. By definition, $f' \in F_{S_{\frac{1}{2}}}(f_r)$, thus the algorithm can not distinguish between the case where $g = f' \in F$, and the case where $g = f' \in F_{S_{\frac{1}{2}}}(f_r)$. □

We bring a simple quantum procedure that solves P_F problems very efficiently. Note that in some cases (as we shall see in the next section) there may exist even more efficient quantum procedures.

Proposition 4.1.5 *The following quantum procedure solves P_F correctly using only r oracle calls.*
Procedure** M_F**:
Run M_{f_i} on g for $1 \leq i \leq r$. If $M_{f_i}(g) = 1$ for any i, answer "$g \in F$", otherwise answer "$g \notin F$" (denote these answers $M_F(g) = 1$ and $M_F(g) = 0$ respectively).

Proof: Suppose first that $g = f_j \in F$. By proposition 3.1.7, applying M_{f_i} on g for $1 \leq i \leq r$ assures that $M_{f_i}(g) = 1$ when $i = j$.

Suppose now that $g \in F_{S_{\frac{1}{2}}}(f_r)$, and let $F_{\frac{1}{2}}(\mathrm{g})$ be as defined in definition 3.1.2. From the first claim of lemma 4.1.3, for each function $f_j \in F$, it holds that $d(f_j, g) = 2^{n-1}$. It follows that $F \subseteq F_{\frac{1}{2}}(g)$, which by proposition 3.1.7 assures that $M_{f_i}(g) = 0$, $1 \leq i \leq r$. □

4.1.1 The number of new problems

In order to count the number of problems that can be constructed in the above manner, it helps to think of the building procedure as a path in a tree. The vertices in the tree are functions and the root is f_1 — the first function we choose. For each vertex f in the tree, its children are all the functions in $F_{S_{\frac{1}{2}}}(f)$, and we limit the tree's depth to r. Obviously, the set F of any problem P_F is a path in this tree from the root to a leaf.

From the third claim of lemma 4.1.3, in the ith step of the algorithm we choose one function out of

$$\left(\begin{array}{c} 2^{n-i+1} \\ 2^{n-i} \end{array} \right)^{2^{i-1}}$$

possible functions. It follows that there are

$$\prod_{i=1}^{r} \left(\begin{array}{c} 2^{n-i+1} \\ 2^{n-i} \end{array} \right)^{2^{i-1}}$$

possible root-to-leaf paths in the tree. We note that there can be no two identical such paths in the tree, and that a given F can have at most $(r-1)!$ possible permutations that would lead to its selection in the tree (the selection order of all the functions except the first can be changed arbitrarily). We may thus conclude that there are at least:

$$\frac{\prod_{i=1}^{r} \left(\begin{array}{c} 2^{n-i+1} \\ 2^{n-i} \end{array} \right)^{2^{i-1}}}{(r-1)!}$$

different P_F problems.

4.2 Quantum advantage without entanglement

Following the previous chapters, it is only natural to ask whether there are subsets of P_F that require no entanglement in order to be solved on a quantum computer, and still have an advantage over any classical algorithm.

Recall that in Chapter 3 we saw that for any $f, f' \in F^{\otimes}$, $f = f_{\vec{a},c}$ and $f' = f_{\vec{a}',c'}$ such that $a' \neq a$, it holds that $d(f, f') = 2^{n-1}$. It follows that any partition of $F^{\otimes}$ into two non-empty sets F and $F^{\otimes} \setminus F$, can be thought of as a promise (and can be solved by the procedure M_F using $|F|$ oracle calls).

Definition 4.2.1 *Let F be a non-trivial subset of $F^{\otimes}$. The promise problem $P_F^{\otimes}$, is the problem of deciding whether a given oracle g is in F or not, where it is promised that $g \in F^{\otimes}$.*

As we shall see below, there is a quantum procedure that solves any $P_F^{\otimes}$ problem using only one quantum query. However, a classical algorithm which uses only one oracle query learns at most one bit of the input oracle's a, and thus can only solve problems for which all oracles with one value of this bit are in F, and all oracles with the other value are in $F^{\otimes} \setminus F$. Except for this very special type of problem, the quantum procedure will have an advantage of up to n queries — as in the $DJ^{\otimes}$ problem.

Since $P_F^{\otimes}$ is a subproblem of P_F, procedure M_F solves it using $|F|$ oracle queries. As the promise contains only non-entangling functions, M_F uses no entanglement to solve $P_F^{\otimes}$. However, we shall now see a much more efficient (and familiar) quantum procedure that solves any $P_F^{\otimes}$ using only one oracle query, and no entanglement. We go back to the DJ procedure and note that after applying it on a quantum oracle U_f we are left with the state:

$$\frac{1}{2^n} \sum_{z \in \{0,1\}^n} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot z + f(x)} |z\rangle$$

The coefficient of the state $|z\rangle$ is thus:

$$\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot z + f(x)}$$

If f is a linear function, i.e. $f(x) = (a \cdot x) \oplus c$, then the coefficient of the state $|a\rangle$ is:

$$\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot a + f(x)} = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot a + x \cdot a + c} = \pm 1.$$

In other words, to learn what a is, we need only measure the output bits[3]. It follows that one quantum query is enough for solving the problem $P_F^{\otimes}$, and even learning exactly what f is. Note that once f is non-linear, the above sum never collapses so nicely to ± 1, and this method becomes probabilistic.

[3] As opposed to the DJ algorithm, in which we only check whether the register is $|0\rangle$ or not. Note also that this is exactly the Bernstein-Vazirani algorithm [7].

Chapter 5

Summary and Discussion

We studied the advantage of exact quantum algorithms in the oracle-based setting, focusing on the role of entanglement and on the creation of new promise problems. We showed that the Deutsch-Jozsa problem is a special case of a huge set of promise problems. We found which of these problems has an entanglement-free subproblem. For these subproblems, we showed maximality and that there exists an $O(1)$ to $\Theta(n)$ quantum advantage over the best exact classical algorithm.

We then brought a method for constructing a large amount of new promise problems, all with a quantum advantage over the best classical algorithm, and gave a lower bound for the number of these problems. This advantage ranges from constant to exponential, depending on the size of the problem. We examined the entanglement-free case for these problems, and identified the subproblems that can be solved without entanglement and still have a quantum advantage of up to $O(1)$ to $\Theta(n)$ over the classical case.

5.1 Entanglement in quantum computing

The basic non-classical feature that allows quantum advantage is superposition. The ability of quantum operations to act in parallel on a superposition of a huge set of states is the source for the quantum computational power. Our results show that there are cases where quantum superposition alone is enough for a non-negligible quantum advantage over classical means. However, looking at algorithms with exponential quantum advantage, it seems very likely that a quantum algorithm must make an intensive use of entan-

glement to achieve an exponential advantage.

While our work deals with pure-state quantum algorithms without any amount of entanglement, related works deal with pure states with a small amount of entanglement, mixed states without entanglement, and other usages of pure-state quantum computing without entanglement (see for example [18, 1, 33, 11, 26, 20, 8, 32]).

5.2 Creating new promises

Our method for constructing promise problems is based on finding sets of functions that differ from one another by half of their values. This allows the quantum algorithm to deterministically distinguish between such functions, using a simple extension of the same basic method used in the original Deutsch-Jozsa algorithm. In general, one may construct promise problems as follows: for any given set of functions F, find the set of functions $F^{\frac{1}{2}}$ such that for all $f \in F$ and $f' \in F^{\frac{1}{2}}$ it holds that f and f' differ on half of their values (assuming that $F^{\frac{1}{2}} \neq \emptyset$). Executing the procedure M_f for all the functions in F will distinguish between oracles from F and $F^{\frac{1}{2}}$, and in case the $F^{\frac{1}{2}}$ set is large enough this would lead to a significant quantum advantage. However, it seems difficult to define or even bound $F^{\frac{1}{2}}$ in general (see open questions below). Note that in our construction method we add the constraint that the functions in F also differ from one another by half their values, which allows to precisely define $F^{\frac{1}{2}}$, thus avoiding the difficulty.

5.3 Conclusions and open questions

Our work can be summarized into two main conclusions about oracle-based exact quantum computing. The first is that non-negligible quantum advantage exists even without advantage, and the second is that it is possible to construct a very large amount of different promise problems for which there is a quantum advantage.

Further research on the role of entanglement in quantum information processing may illuminate some of the following questions:

Is there a subproblem larger than $DJ^{\otimes}$ and a corresponding algorithm (other than the Deutsch-Jozsa algorithm) that solves it without entanglement, and yet has an advantage over any classical algorithm? Note that in

our work we showed maximality of the subproblem only with regard to the Deutsch-Jozsa-like algorithms.

Alternatively, Can it be proven that exponential advantage of exact QCWE over classical-exact computation is impossible in oracle-based settings? Note that this is not known yet, even for the Deutsch-Jozsa problem, since there may be some other quantum procedure for which the QCWE advantage holds, and the subset is larger than $F_{DJ}^{\otimes}$.

Given an arbitrary set of Boolean functions F, is it possible to characterize precisely the set of functions that differ on half their values from *all* the functions in F? This will provide a method for constructing an even larger set of promise problems, some of which as we saw, may have a significant quantum advantage. Furthermore, could the promise problems we found in chapter 4 (or by this more general method), be used as computation tools in quantum computation?

Looking beyond the world of Boolean functions, is there a restricted form of the Simon problem (or more generally of the hidden subgroup problem [24]), and a corresponding quantum algorithm that presents a quantum advantage without entanglement?

Appendix A

Entanglement in Simon's and Grover's problem

A.1 Simon's Problem

Simon [27] presented the following oracle problem:
Let $f : \{0,1\}^n \to \{0,1\}^n$ be a 2-to-1 function[1], such that

$$\forall x \neq y : \ f(x) = f(y) \Leftrightarrow y = x \oplus a, \tag{A.1}$$

where a is a fixed n-bit string called the function's period, and $\oplus$ is the bitwise XOR operation. The goal is to determine the value of a.

In order to solve this problem classically with high probability, the oracle must be queried an exponential number of times. However, the following quantum procedure solves it with high probability using a polynomial number of queries.

1. The initial state is $|\psi_0\rangle = |0\rangle^{\otimes n}|0\rangle^{\otimes n}$.

2. After applying n Hadamard gates on the first n qubits:

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle|0\rangle^{\otimes n}. \tag{A.2}$$

[1] In some versions it is also allowed to be constant.

3. Applying the quantum oracle f yields:

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle|f(x)\rangle. \tag{A.3}$$

4. We now measure the last n qubits and obtain a certain $f(x_0) \in \{0,1\}^n$, thus leaving the first n qubits in the state:

$$|\psi_3\rangle = \frac{1}{\sqrt{2}}(|x_0\rangle + |x_0 \oplus a\rangle). \tag{A.4}$$

5. Applying n Hadamard gates on the n qubits yields:

$$|\psi_4\rangle = \frac{1}{2^{(n+1)/2}} \sum_{y\in\{0,1\}^n} [(-1)^{x_0\cdot y} + (-1)^{(x_0\oplus a)\cdot y}]|y\rangle \tag{A.5}$$

$$= \frac{1}{2^{(n-1)/2}} \sum_{a\cdot y=0} (-1)^{x_0\cdot y}|y\rangle. \tag{A.6}$$

6. Measure $|\psi_4\rangle$ to get a y such that $a \cdot y = 0$.

Repeating steps 1-6 a polynomial number of times will, with high probability, result in n linearly-independent values $\{y_1, y_2, \ldots, y_n\}$ such that $y_i \cdot a = 0$, which determines a.

We now use a counting argument to show that if no entanglement is used in the algorithm, $f(x)$ must be constant, and therefore it is a trivial case of the problem.

Proposition A.1.1 *Any instance of the Simon problem generates entanglement in Simon's algorithm.*

Proof: To see this, it suffices to examine the $2n$-qubit state $|\psi_2\rangle$. If $|\psi_2\rangle$ is separable (i.e. over all cuts), it can clearly be decomposed in the computational basis as:

$$|\psi_2\rangle = \Big(\otimes_{i=1}^{n} (\alpha_i|0\rangle + \beta_i|1\rangle)\Big) \otimes \Big(\sum_{y\in\{0,1\}^n} \delta_y|y\rangle\Big) \tag{A.7}$$

$$= \Big(\sum_{x=0}^{2^n-1} \gamma_x|x\rangle\Big) \otimes \Big(\sum_{y\in\{0,1\}^n} \delta_y|y\rangle\Big). \tag{A.8}$$

The number of terms in $|\psi_2\rangle$ in the computational basis, according to Eq. (A.3), is exactly 2^n. Note that none of the α_is and β_is in Eq. (A.7) can be zero, otherwise at least one value of x would be missing from the final sum, in contradiction to $|\psi_2\rangle$'s definition (Eq. (A.3)). Suppose that the right-hand side of the tensor product in Eq. (A.8) has more than one term in the sum. Since the left-hand side of the tensor product has exactly 2^n terms, it would follow that the final sum has more than 2^n terms, in contradiction to $|\psi_2\rangle$'s definition as having *exactly* 2^n terms in the computational basis. Thus the last n qubits must be in a single state, or in other words, f must be constant. □

We conclude that the usage of the Simon algorithm for any subproblem of the Simon problem requires entanglement. However, an interesting open question in this context, is whether there is a restricted version of the problem solvable by some *other* quantum algorithm without entanglement, and achieves an advantage over the classical case.

A.2 Grover's Search Algorithm

Grover presented an algorithm [15], which for a binary function $f : \{0,1\}^n \to \{0,1\}$, finds an x such that $f(x) = 1$ with only $O(\sqrt{2^n})$ oracle calls.

The first step of the algorithm is identical to the Deutsch-Jozsa subroutine:

$$\frac{1}{\sqrt{N}} \sum |x\rangle \to \frac{1}{\sqrt{N}} \sum (-1)^{f(x)} |x\rangle = |\psi_2\rangle. \tag{A.9}$$

We require that the resulting state ($|\psi_2\rangle$) will be separable. In the following we show that this may be true only for trivial instances of the search problems and therefore we do not need to follow additional steps of the algorithm. From the separability of $|\psi_2\rangle$ it follows that if the $n-1$ most significant bits are measured, the state of the least significant bit is independent of the measurement outcome. Writing $x = w0$ or $x = w1$ depending on the value of the LSB,

$$|\psi_2\rangle = 1/\sqrt{N} \sum_w \left[(-1)^{f(w0)} |w0\rangle + (-1)^{f(w1)} |w1\rangle \right] \tag{A.10}$$

and measuring all but the LSB, we are left with the state

$$\frac{(-1)^{f(w0)}|0\rangle + (-1)^{f(w1)}|1\rangle}{\sqrt{2}}. \tag{A.11}$$

Up to a global phase, the state is

$$\frac{|0\rangle + (-1)^{f(w0)\oplus f(w1)}|1\rangle}{\sqrt{2}} \tag{A.12}$$

and has to be independent of the measured w. That means that $\forall x$: $(-1)^{f(x)\oplus f(x\oplus\overbrace{00\cdots0}^{n-1}1)} = k$, or more conveniently put as $\forall x : f(x) \oplus f(x \oplus 1) = y \cdot e_1$. Similarly, entanglement should also be avoided when regarding the jth qubit for $1 \leq j \leq n$, which leads to

$$\forall x, j : f(x) \oplus f(x \oplus e_j) = y \cdot e_j. \tag{A.13}$$

This means that it is true for a couple of js combined

$$f(x\oplus e_j)+f(x\oplus e_j\oplus e_i) = y\cdot e_i \Longrightarrow f(x)\oplus f(x\oplus e_j\oplus e_i) = y\cdot e_i\oplus y\cdot e_j \tag{A.14}$$

In the same manner, we can see that for every $J \in \{0,1\}^n$ it holds that

$$\forall x : f(x) \oplus f(x \oplus J) = y \cdot J \tag{A.15}$$

even for $x = 0$. From this follows that functions that do not generate entanglement must satisfy

$$f(J) = y \cdot J \oplus f(0), \tag{A.16}$$

which is the exact definition of the set $F^{\otimes}_{DJ}$. These functions correspond to trivial search problems where none, all, or half of the elements are to be found. Hence, any interesting instance of Grover's search problem would generate entanglement in Grover's algorithm.

One may confirm that this is true even for $n = 2$ bits, regardless of the fact that only a single oracle call is required in that case, and unlike what is commented by [10]. With two bits, $|\psi_1\rangle = \frac{1}{2}[|00\rangle + |01\rangle + |10\rangle + |11\rangle]$. If only one of $\{00, 01, 10, 11\}$ is "marked", then $|\psi_2\rangle$ has 3 positive coefficients and a single negative coefficient. This means that $|\psi_2\rangle = A|00\rangle + B|01\rangle + C|10\rangle + D|11\rangle$ is entangled, as it cannot satisfy the separability constraint $AD = BC$.

Bibliography

[1] S. Aaronson, "Multilinear Formulas and Skepticism of Quantum Computing," *Proc. 36th ACM STOC* , pp. 118–127, 2004. Also in quant-ph/0311039.

[2] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," in *Proc. of IEEE Int. Conf. on Computers, Systems and Signal Processing*, pp. 175–179, Dec. 1984.

[3] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, "Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels," *Phys. Rev. Lett.* **70**, pp. 1895–1899, Mar. 1993.

[4] C. H. Bennett, G. Brassard, and N. D. Mermin, "Quantum cryptography without bell's theorem," *Phys. Rev. Lett.* **68**, pp. 557–559, 1992.

[5] P. O. Boykin, T. Mor, M. Pulver, V. Roychowdhury, and F. Vatan, "A New Universal and Fault-Tolerant Quantum Basis," *Info. Proc. Lett.* **75**, pp. 101–107, 2000. Also in quant-ph/9906054.

[6] C. H. Bennett and S. J. Wiesner, "Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states," *Phys. Rev. Lett.* **69**(20), pp. 2881–2884, 1992.

[7] E. Bernstein and U. Vazirani, "Quantum complexity theory†," *SIAM J. Comp.* **26**(5), pp. 1411–1473, 1997.

[8] E. Biham, G. Brassard, D. Kenigsberg, and T. Mor, "Quantum computing without entanglement," *Theor. Comput. Sci.* **320**, pp. 13–33, 2004.

[9] E. Biham, B. Huttner, and T. Mor, "Quantum cryptographic networks based on quantum memories," *Phys. Rev. A* **54**, pp. 2651–2658, 1996.

[10] S. L. Braunstein and A. K. Pati, "Speed-up and entanglement in quantum searching," *Quantum Inf. Comput.* **2**, pp. 399–409, 2002. Also in quant-ph/0008018.

[11] J. I. Cirac, W. Dür, B. Kraus and M. Lewenstein, "Entangling operations and their implementation using a small amount of entanglement," *Phys. Rev. Lett.* **86**, pp. 544–547, 2001.

[12] D. Collins, K. W. Kim, and W. C. Holton, "Deutsch-Jozsa algorithm as a test of quantum computation," *Phys. Rev. A* **58**, p. 1633(R), Sep. 1998.

[13] D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation," *Proc. R. Soc. Lond., Series A* **A439**, pp. 553–558, 1992.

[14] A. K. Ekert, "Quantum cryptography based on bell's theorem," *Phys. Rev. A* **67**, pp. 661–663, 1991.

[15] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th ACM STOC*, pp. 212–219, May 1996.

[16] M. Horodecki, "Entanglement measures," *Quantum Inf. Comput.* **1**(1), pp. 3–26, 2001.

[17] R. Jozsa, *Entanglement and Quantum Computation.* Oxford University Press (Oxford, England), Jan. 1998. Also in quant-ph/9707034.

[18] R. Jozsa and N. Linden, "On the role of entanglement in quantum computation speed-up," *Proc. R. Soc. Lond. A* **459**, pp. 2011–2032, Aug. 2003.

[19] D. Kenigsberg, Tal Mor and G. Ratsaby, "Quantum Advantage without Entanglement," accepted for publication in *Quantum Inf. Comput.*.

[20] N. Linden and S. Popescu "Good Dynamics versus Bad Kinematics: Is Entanglement Needed for Quantum Computation?," *Phys. Rev. Lett.* **87**, 047901, 2001.

[21] S. Lloyd, "Quantum search without entanglement," *Phys. Rev. A* **61**, p. 010301(R), Dec. 1999.

[22] D. A. Meyer, "Sophisticated quantum search without entanglement," *Phys. Rev. Lett.* **85**, pp. 2014–2017, 2000.

[23] T. Mor and G. Ratsaby, "On Oracles, Promises and Exact Quantum Computation,", in preparation.

[24] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press (Cambridge), 2000.

[25] C. Papadimitriou, *Computational Complexity*, Addison-Wesley, 1994.

[26] S. L. Braunstein, C. M. Caves, R. Jozsa, N. Linden, S. Popescu and R. Schack, "Separability of Very Noisy Mixed States and Implications for NMR Quantum Computing," *Phys. Rev. Lett.* **83**, pp. 1054–1057, 1999. Also in quant-ph/9811018.

[27] D. R. Simon, "On the power of quantum computation," *SIAM J. Comp.* **26**(5), pp. 1474–1483, 1997.

[28] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Phys. Rev. A* **52**, pp. R2493–R2496, Oct. 1995.

[29] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comp.* **26**(5), pp. 1484–1509, 1997.

[30] B. Terhal and J. A. Smolin, "Single quantum querying of a database," *Phys. Rev. A* **58**, pp. 1822–1826, 1998.

[31] A. M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem.," *Proc. Lond. Math. Soc.* **Ser. 2 42**,pp. 230–265, 1937.

[32] L. Vaidman and Z. Mitrani, "Qubits versus Bits for Measuring an Integral of a Classical Field," *Phys. Rev. Lett.* **92**, 217902, 2004.

[33] G. Vidal, "Efficient classical simulation of slightly entangled quantum computations," *Phys. Rev. Lett.* **91**, 147902, 2003. Also in quant-ph/0301063.

יתרון קוונטי, גם ללא שזירות

גיל רצבי

יתרון קוונטי, גם ללא שזירות

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת תואר
מגיסטר למדעים במדעי המחשב

גיל רצבי

הוגש לסנט הטכניון – מכון טכנולוגי לישראל
סיון ה'תשס"ו חיפה מאי 2006

המחקר נעשה בהנחיית דר. טל מור בפקולטה למדעי המחשב.

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי.

תוכן עניינים

רשימת איורים

תקציר

חישוביות קוונטית היא תחום שהתפתח כתוצאה משילוב תורת הקוונטים בתיאוריה של מדעי המחשב, ובפרט בתורת החישוביות ותורת הסיבוכיות החישובית. תורת החישוביות עוסקת בשאלה: איזה בעיות ניתנות לפתרון באופן עקרוני בעזרת מחשב. ניתן היה לשער שבהינתן מספיק זמן וזיכרון, מחשב רגיל יוכל לבצע כל משימת חישוב שתוטל עליו. אולם מסתבר כי לא כך הוא: בעיות רבות אינן ניתנות לפתרון באופן עקרוני, וביניהן אף מספר בעיות אשר נראות פשוטות למראית עין, למשל בעיית העצירה: "בהינתן אלגוריתם וקלט אליו, האם האלגוריתם יעצור?" טיורינג הראה כי לא קיים אלגוריתם הפותר בעיה זו. בהינתן בעיה פתירה, אך טבעי הוא לשאול מהם המשאבים הנדרשים לפתרונה. תורת הסיבוכיות עוסקת בסיווג בעיות חישוביות לפי כמות המשאבים הנדרשת לפתרונן, וביחסים בין מחלקות אלו. נהוג לדבר בעיקר על משאב הזמן ומשאב הזיכרון, למרות שניתן לחשוב גם על משאבים נוספים, כגון כמות המעבדים. בתורת הסיבוכיות אנו מודדים זמן במספר צעדי החישוב הנדרשים לפתרון בעיה כפונקציה של אורך הקלט. גישה מקובלת היא לסווג בעיות הניתנות לפתרון בזמן פולינומיאלי כאל בעיות הפתירות ביעילות, ואילו בעיות אשר ידוע עבורן אך ורק פתרון אקספוננציאלי נחשבות בלתי ניתנות לפתרון יעיל. למרות שאבחנה זו עלולה להיות גסה, היא בדרך כלל מזהה בצורה טובה את הבעיות הקשות לפתרון.

אף על פי שקיימים מודלים רבים לחישוב (וביניהם מכונת טיורינג - אחד

המודלים המקובלים ביותר למחקר בתורת הסיבוכיות), נמצא כי כל המודלים הללו שקולים, במובן שכל בעיה הניתנת לפתרון במודל אחד ניתנת לפתרון גם במודל שני. תיזת צ'רץ'-טיורינג קובעת כי "כל המודלים הסבירים לחישוב שקולים". הגרסה החזקה של תיזת צ'רץ'-טיורינג אף קובעת שכל המודלים הסבירים של חישוב שקולים פולינומיאלית, כלומר כל בעיה הניתנת לפתרון פולינומיאלי במודל אחד ניתנת לפתרון פולינומיאלי גם במודל שני.

בראשית שנות השמונים של המאה שעברה, נוכח ריצ'רד פיינמן שסימולציה של מערכות קוונטיות באמצעות מחשב קלאסי היא בעיה קשה. נראה כי סימולציה של מערכות קוונטיות, אפילו כאלה פשוטות יחסית, דורשת זמן אקספוננציאלי ולכן אינן פתירות מעשית. פיינמן הציע, כי מחשב המשתמש באפקטים קוונטיים יוכל בעקרון לפתור בעיות כאלה, ולכן יכול להיות שמחשב כזה הוא בעל כוח חישובי רב יותר מזה של מחשב רגיל, ואולי אף בעל יתרון אקספוננציאלי - בניגוד לגרסה החזקה של תיזת צ'רץ'-טיורינג.

בדומה לביט הקלאסי, גם לביט הקוונטי - הקיוביט - יש שני מצבים בסיסיים. אולם בניגוד לביט רגיל, הביט הקוונטי יכול להיות גם בסופרפוזיציה של שני מצבי הבסיס שלו, תכונה המאפשרת מקביליות עצומה בחישוב הקוונטי. מכיוון שכל חישוב הוא סדרת שינויים במצב של רגיסטרי המחשב, יש צורך להגדיר "פעולות חישוביות" על רגיסטרים קוונטיים אשר מבצעות (לפחות) את כל הפעולות הניתנות לביצוע על ידי מחשב קלאסי. שינויים במצב של מערכת קוונטית ניתנים לתיאור ע"י אופרטור אוניטרי, ואכן לכל פעולת חישוב קלאסית ניתן להגדיר אופרטור אוניטרי המבצע את אותה פעולה במחשב קוונטי.

תכונה חשובה של מערכות קוונטיות אשר אין לה מקבילה בעולם הקלאסי היא השזירות. תורת הקוונטים מורה לנו כיצד לתאר את המצב המשותף של מערכות קוונטיות נפרדות בעלות מצב מוגדר. אולם תורת הקוונטית מאפשרת גם למערכת קוונטית להימצא במצבים אשר אינם ניתנים לתיאור ע"י מצבי

תת-המערכות שלה. מצבים כאלה נקראים מצבים שזורים. אי-הקלאסיות של מצבים כאלה ניתנת לבחינה ע"י אי-שיוויון בל. בל הוכיח כי ניסויים בעולם הקלאסי אשר בודקים קשר בין חלקיקים מרוחקים, נשמעים לאי-שוויון המגביל את הקורלציה בין החלקיקים. מצבים שזורים מפרים את אי-שוויון בל, ואכן בשנות השישים של המאה שעברה נעשו ניסוים רבים המאמתים תכונה זו של מצבים שזורים.

רבים מאמינים כי שזירות היא היא המקור ליתרונו של החישוב הקוונטי על פני חישוב קלאסי, כיוון שזו התכונה הלא-קלאסית היסודית בתורת הקוונטים. ואכן, בכל האלגוריתמים הקוונטים המציגים יתרון משמעותי על פני חישוב קלאסי, נמצא כי נעשה שימוש בכמות גדולה של שזירות.

בשנת 1992, דויטש וג'וזסה הציגו את הבעיה הראשונה אשר הוכח עבורה יתרון קוונטי אקספונציאלי על פני חישוב קלאסי. דויטש וג'וזסה הניחו קיומו של אורקל - קופסה שחורה העונה לשאילתות - עבור פונקציה בוליאנית, כאשר מובטח כי הפונקציה היא או קבועה או מאוזנת (כלומר או שכל ערכיה שווים, או שחצי מהם שווים 0 וחצי שווים 1). מטרת החישוב היא לזהות האם האורקל הנתון הוא של פונקציה קבועה או של פונקציה מאוזנת. ניתן לראות בקלות כי במקרה הקלאסי יש להציג לאורקל מספר אקספוננציאלי של שאילתות על מנת לפתור בוודאות את בעיית דויטש-ג'וזסה. אולם במקרה הקוונטי יש צורך אך ורק בהצגת שאילתה יחידה על מנת לפתור את הבעיה - ומכאן היתרון האקספונציאלי. גם בבעיית דויטש-ג'וזסה נעשה שימוש רב בשזירות, כיוון שהרוב המוחלט של אורקלים של פונקציות בוליאניות יוצר ברוב המקרים שזירות לאחר שאילתה אליו.

במהלך השנים האחרונות פותחו אלגוריתמים ויישומים קוונטים מפתיעים רבים אשר השתמשו בכלים המתאפשרים מתורת הקוונטים, והציגו יתרונות שנחשבו בלתי אפשריים בעולם הקלאסי. החל בטלפורטציה של מצבים קוונטיים, קידוד

סופר-צפוף של מידע, אלגוריתמי הפצת מפתחות קוונטים, וכלה באלגוריתם החיפוש של גרובר. אולם ההישג המפורסם ביותר של החישוביות הקוונטית הוא האלגוריתם של שור אשר מפרק מספר לגורמיו הראשוניים ומחשב לוגריתם בדיד בזמן פולינומיאלי - פעולות אשר לא ידוע עבורן אלגוריתם קלאסי יעיל. אלגוריתם זה משך עניין פרקטי רב הנובע מיכולתו לתקוף את צפני המפתח הציבורי העיקריים שבשימוש.

שני מניעים עיקריים היוו את התמריץ לעבודה זו. הראשון היה לחקור את תפקידה של השזירות בחישוב קוונטי, ובפרט האם יכול להתקיים יתרון קוונטי משמעותי על פני חישוב קלאסי, גם ללא שזירות. המניע השני היה מציאת בעיות חדשות בעלות יתרון קוונטי (מניע המשותף לכל העוסקים בתחום), וחקירת התפקיד של שזירות בהן.

המחקר שלנו התרכז בעיקר בחישוב קוונטי עם מצבים טהורים המשתמש באורקלים והבטחות (כלומר הגבלת הקלטים האפשריים). בפרק 2, אנו מוצאים את תת-הבעיה הגדולה ביותר של בעיית דויטש-ג'וזסה, הניתנת לפתרון ללא שזירות. אנו מזהים את קבוצת הפונקציות הבוליאניות אשר אורקל הפאזה שלהן אינו יוצר שזירות (שאר הרכיבים באלגוריתם גם הם אינם יוצרים שזירות) ומוסיפים את המגבלה שפונקציות הקלט, מלבד היותן קבועות או מאוזנות, ישתייכו לקבוצה זו. אנו מראים כי אלגוריתם דויטש-ג'וזסה עדיין מחזיק ביתרון לא-זניח על פני האלגוריתם הקלאסי הטוב ביותר האפשרי. אנו אף מראים כי תת-בעיה זו היא בעלת חשיבות נוספת, כיוון שהיא מכילה את כל הפונקציות הבוליאניות אשר אורקל הפאזה שלהן אינו יוצר שזירות.

בפרק 3 גישה זו מוכללת: אנו מראים כי לכל פונקציה בוליאנית קיימת בעיה דמוית בעיית דויטש-ג'וזסה ואלגוריתם קוונטי מתאים הפותר אותה ביתרון אקספונציאלי, כלומר קיימות מספר סופר-אקספונציאלי של בעיות דמויות דויטש-ג'וזסה בעלות יתרון קוונטי זהה לבעיית דויטש-ג'וזסה המקורית. אנו

מוצאים את תת הקבוצה של בעיות אלו הניתנות לפתרון ללא שזירות, ועדיין בעלות יתרון קוונטי משמעותי, וכן מראים שיש מספר אקספונציאלי של בעיות כאלה.

בחלק האחרון של העבודה אנו מציגים שיטה לבניית מספר גדול של בעיות הבטחה, אשר ניתנות לפתרון יעיל קוונטי. היתרון הקוונטי לבעיות אלו נע בין אקספונציאלי ללינארי, כתלות בבעיה הספציפית. אנו חוקרים את המקרה נטול השזירות בבעיות אלה, ומוצאים את התנאים ההכרחיים לכך שהבעיה ניתנת לפתרון ללא שזירות ועדיין תהיה בעלת יתרון קוונטי על פני האלגוריתם הקלאסי הטוב ביותר האפשרי.