An Introduction to Quantum Computing

Michal Charemza
University of Warwick

March 2005

Acknowledgments

Special thanks are given to Steve Flammia and Bryan Eastin, authors of the LaTeX package, *Qcircuit*, used to draw all the quantum circuits in this document. This package is available online at http://info.phys.unm.edu/Qcircuit/.

Also thanks are given to Mika Hirvensalo, author of [11], who took the time to respond to some questions.

Contents

Chapter 1

Introduction

The classical computer was originally a largely theoretical concept usually attributed to Alan Turing, called the *Turing Machine*. The first computers to realise this theory used valves as the core component for processing. This eventually progressed to transistors, which are the building block of current processors. In 1965 GordoSn Moore made an empirical observation and prediction that the density of transistors on a chip doubles roughly every 18 months. As

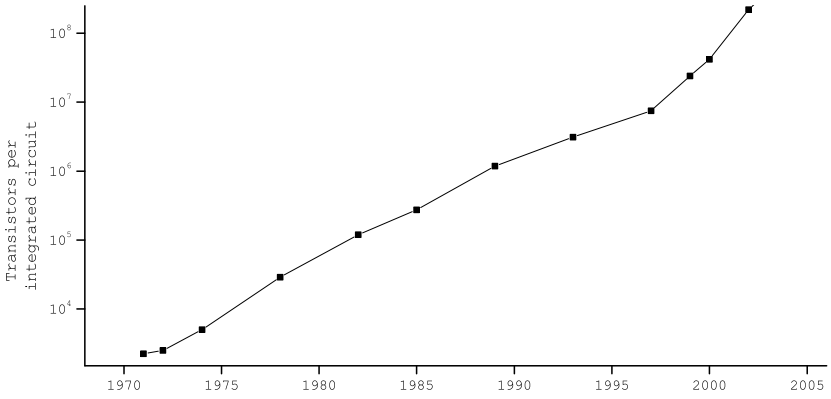

Figure 1.1: Moore's Law. Source: Intel.

can be seen in Figure 1.1, this law has roughly held for about 40 years. However, this trend cannot continue indefinitely. There are differing estimates as to when the trend will reach its limit, one even within 6 years [13], but it is generally agreed that a limit *will* be reached.

As the transistor size becomes close to the atomic scale, quantum effects, i.e. the physical laws of the very small, will begin to dominate how the transistors act. However, if some, at the moment rather major, technical hurdles can be overcome, a computer that exploits these effects, a quantum computer, could be built that have some powerful properties. It is our aim to give a brief introduction to some of these properties.

In order to do this we will describe quantum states, and how they are represented mathematically in Chapter 2. In Chapter 3 possible operations on these states are discussed. Chapter 4 describes how a quantum computer could do

anything a classical computer could do. The remaining chapters focus on algorithms that a classical computer *cannot do*, or at the very least, cannot do efficiently.

Chapter 2

Quantum States

In this chapter we introduce the concepts of how information is stored in a quantum computer, how to describe this formally, and make comparisons to the classical case. We will also briefly, and rather informally, describe the quantum physics that make our descriptions reasonable. Recall a *Hilbert space* is a complete inner product space over the complex numbers. For vectors in a Hilbert space will use the Dirac notation $|\psi\rangle$.

Definition 2.0.1 (Qubit)**.** A *quantum bit*, or *qubit* for short, is a 2-dimensional Hilbert space H_2. We label an orthonormal basis of H_2 by $\{|0\rangle, |1\rangle\}$. The *state* of the qubit is an associated unit length vector in H_2. If a state is equal to a basis vector then we say it is a *pure* state. If a state is any other linear combination of the basis vectors we say it is a *mixed* state, or that the state is a *superposition* of $|0\rangle$ and $|1\rangle$.[1]

If the associated state of some qubit, that we label A say, is state $|\phi\rangle$ at some moment in time, then we often say that A *has state* $|\phi\rangle$, or A *is in state* $|\phi\rangle$.

Example. An example of a mixed state of a qubit is

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle).$$

Clearly this is a unit length vector in H_2.

Two states are seen as equivalent up to a scale factor of some complex number α such that $|\alpha| = 1$. This leads to the fact a qubit can be seen as a sphere in 3D space, and the state of the qubit as a point on the sphere. In this case the sphere is known as the *Bloch* sphere [16].

A classical bit is traditionally defined as $\mathbb{F}_2$, and the state of the classical bit as a member of $\mathbb{F}_2$, i.e. either 0 or 1. Thus there are only two possible states a classical bit can be in. However, a qubit can be in any state $a|0\rangle + b|1\rangle$ where $a, b \in \mathbb{C}$ and $|a|^2 + |b|^2 = 1$.

There are only two possible *pure* states for a qubit, $|0\rangle$ and $|1\rangle$. Thus there is a similarity between a qubit in a pure state and a classical bit. A further similarity is related to observation, which will be mentioned later. We can, and will, often think about classical bits as qubits in pure states.

[1] There is a more general definition of quantum states as *operators* on Hilbert spaces. For our purposes this definition is unnecessary.

Remark. Note that what we are actually doing is trying to model a physical system. For the classical case we say that the state of a bit is 0 or 1. This is just trying to represent some physical state that can be in one of two possible states. If we are modelling a compact disc, 0 would represent a 'hole' (albeit very small) and 1 would represent 'no hole'. If we are modelling a CPU, 1 would represent current going round part of a circuit, and 0 would represent no current going round that part of a circuit. We could even just be modelling the state of a light, either on or off. We do not want to try to model every tiny physical part of each of these systems. If we are just interested in seeing what can be done with some physical system that has the property that can be in one of two states, we just model that very property. Thus whatever we discover could, in theory, be applied to any physical system that has that property.

We are doing a similar thing here. We are making the assumption that some physical system has a property that can be in a superposition of states (along with some other assumtions relating to observation and operations, which will be discussed later), and seeing if that could lead to anything useful.

Remark 2.0.1. The Nuclear Magnetic Resonance, or NMR, quantum computers, such as the one that successfully factored 15 in 2001 [23], use millions of identical molecules as the quantum computer. Roughly speaking the computer had the same number of qubits as atoms in the molecule, its just that there were millions of copies of the molecule. The *spin* of each of the atoms in the molecule represented the state of a qubit. We present a simplification of *spin*, but one could say that the spin in state *up* could represent $|0\rangle$, and *down* could represent $|1\rangle$. We could label an orthonormal basis of H_2 by $|\uparrow\rangle$ and $|\downarrow\rangle$ to make this clearer. Due to the laws of quantum mechanics, the spin of the atom can be in a superposition of up and down at the same time, i.e. a linear combination of $|\uparrow\rangle$ and $|\downarrow\rangle$.

Remark 2.0.2. There is an important technical problem that must be solved if quantum computers are ever to be practical. A superposition of states $|0\rangle$ and $|1\rangle$ is known as a *coherent* state. A coherent state is extremely 'unstable'. That is to say that it tends to interact with its environment, and collapse into a pure state. This process is known as *decoherence*, and is seen to be inevitable. Algorithms that exploit quantum effects such as superposition, that we will describe later, are known as quantum algorithms. To apply these quantum algorithms in the real world, decoherence time must be longer than the time to run the algorithm. Thus ways of making decoherence time longer are trying to be found.

Remark 2.0.3. When describing quantum algorithms we will make the assumption that we can always initialize the state of qubits, usually to state $|0\rangle$. It would be very difficult to construct any sort of algorithm where the initial state was not controllable.

Obviously we would like to think about systems larger than those of just one qubit. We now generalise the notion of a qubit as a two-dimensional Hilbert space H_2, to an n-dimensional Hilbert space H_n.

Definition 2.0.2 (Quantum System)**.** A *quantum system* is an n-dimensional Hilbert Space H_n. We label an orthonormal basis on H_n by $\{|x_1\rangle, |x_2\rangle \ldots, |x_n\rangle\}$ such that $x_i \in X$ for some finite X. The associated *state* of the system is a unit length vector in H_n.

Just as for the qubit, if the associated state of some quantum system, that we label S say, is state $|\phi\rangle$ at some moment in time, then we often say that S *has state* $|\phi\rangle$, or S *is in state,* $|\phi\rangle$. We see that a general form for the state of a quantum system is

$$\sum_{i=1}^{n} \alpha_i |x_i\rangle \text{ where } \alpha_i \in \mathbb{C} \text{ and } \sum_{i=1}^{n} |\alpha_i|^2 = 1.$$

For clarity for pure states we will usually use English letters, such as $|x\rangle$, and for states that may be mixed we will usually use Greek letters, such as $|\psi\rangle$. We will also only ever use quantum systems that are some *compound* of some collection of qubits. What is meant by *compound* is explained in the next section.

2.1 Compound States

Often we are interested in describing two quantum systems H_n and H_m at the same time. For example we may, and in fact will, wish to describe the state of two qubits. Let us label an orthonormal basis of H_n by $\{|x_1\rangle, |x_2\rangle \ldots, |x_n\rangle\}$ and label an orthonormal basis of H_m by $\{|y_1\rangle, |y_2\rangle \ldots, |y_m\rangle\}$. For basis states $|x_i\rangle$ and $|y_j\rangle$ we write $|x_i\rangle |y_j\rangle$ or just $|x_i y_j\rangle$ for the cartesian product $(|x_i\rangle, |y_j\rangle)$.

Definition 2.1.1 (Tensor Product of Spaces)**.** Let H_n and H_m be quantum systems with orthonormal bases labelled as above. The *tensor product* $H_n \otimes H_m$ is defined to be the Hilbert space with basis $\{|x_i\rangle |y_j\rangle : i = 1 \ldots n, j = 1 \ldots m\}$.

We call $H_n \otimes H_m$ the *compound* system of H_n and H_m. Clearly $H_n \otimes H_m \cong H_{nm}$, and we will usually write $H_n \otimes H_m = H_{nm}$. Thus a quantum state in H_{nm} must be of the form

$$\sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_{ij} |x_i\rangle |y_j\rangle \text{ where } \alpha_{ij} \in \mathbb{C} \text{ and } \sum_{i=1}^{n}\sum_{j=1}^{m} |\alpha_{ij}|^2 = 1.$$

Example 2.1.1. Recall that we label an orthonormal basis of H_2 by $\{|0\rangle, |1\rangle\}$. Thus an example of a quantum state in $H_2 \otimes H_2$ (two qubits) is

$$\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle).$$

Example 2.1.2. Another example of a quantum state in $H_2 \otimes H_2$ is

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

Two qubits in this state are called an *EPR* pair. This is a state that has caused controversy, and will be mentioned again later.

Consider quantum systems A and B, with A in state $|\psi\rangle$ and B in state $|\phi\rangle$. We would like some way of describing the state in the compound system $A \otimes B$. Fortunately there is an easy way in which to achieve this, using the tensor product $\otimes$.

Definition 2.1.2 (Tensor Product of Vectors)**.**

$$\left(\sum_{i=1}^{n} \alpha_i |x_i\rangle\right) \otimes \left(\sum_{j=1}^{m} \beta_j |y_j\rangle\right) = \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i\beta_j |x_i y_j\rangle .$$

Using this we can say that $A \otimes B$ is in state $|\psi\rangle \otimes |\phi\rangle$. Tensor product sends unit length vectors to unit length vectors, so it does send quantum states to quantum states. Since the basis representations for any states $|\phi\rangle$ and $|\psi\rangle$ are unique, it should be clear that $A \otimes B$ is in state $|\psi\rangle \otimes |\phi\rangle$ if and only if A is in state $|\psi\rangle$ and B is in state $|\phi\rangle$. We call the state $|\psi\rangle \otimes |\phi\rangle$ in $A \otimes B$ the *compound* state of $|\psi\rangle$ and $|\phi\rangle$. Another important notational point is that often the symbol $\otimes$ is omitted. Because for basis states the tensor product $|x\rangle \otimes |y\rangle$ is equal to the cartesian product, there is no conflict of notation when we write $|x\rangle\,|y\rangle$.

We can easily apply the definitions of tensor products of spaces recursively to describe the compound of more than just two quantum systems, which we now do for the specific case where each system is a qubit.

Definition 2.1.3 (Quantum Register)**.** We will call $H_{2^m} = H_2 \otimes H_2 \ldots \otimes H_2$ (m times) a quantum register of length m. Put another way, it is the compound system of m qubits.

We see a quantum register of length m has dimension 2^m. Recall that $\{|0\rangle, |1\rangle\}$ is a basis of H_2. Thus noting the earlier definition of tensor product, a basis of a quantum register of length m is

$$\{|x_0\rangle\,|x_1\rangle \ldots |x_{m-1}\rangle : x_i \in \{0,1\}\}. \tag{2.1}$$

Example 2.1.3. A quantum register $H_2 \otimes H_2$ has basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. Given two states each in H_2

$$|\phi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \in H_2 \qquad\qquad |\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \in H_2,$$

we can see that their tensor product in $H_4 = H_2 \otimes H_2$ is

$$|\phi\rangle \otimes |\psi\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle). \tag{2.2}$$

Remark 2.1.1. The spin of two atoms could be considered as a quantum register of length two. As discussed in Remark 2.0.1 , the state of each can be a linear combination of the states $|\uparrow\rangle$ and $|\downarrow\rangle$. Therefore the state of the compound system can be linear combination of the states $|\uparrow\uparrow\rangle$, $|\uparrow\downarrow\rangle$, $|\downarrow\uparrow\rangle$ and $|\downarrow\downarrow\rangle$ (remember it must be a unit length linear combination).

2.2 Observation

We will now discuss what happens when we observe a quantum system. We will use the words *measure* and *observe* interchangeably to mean the same thing. We say that an observation of the state in H_n,

$$\sum_{j=1}^{n} \alpha_j |x_j\rangle ,$$

results in x_k for some k with probability $P(x_k) = |\alpha_k|^2$. This is essentially why the vector must be unit length: the probabilities $|\alpha_i|^2$ must sum to 1.

The act of observation itself causes the state to collapse to the pure state $|x_k\rangle$. This is a very important point! Another very important point is that the result of measurement is probabilistic. We do not know what the result will be, only the probabilities of possible results.

If a quantum system in some pure state $|x\rangle$ is observed, there is only one possible result, x, and the act of observation does not cause the state to change. This is another reason why we can view classical bits as quantum bits in pure states. We can observe classical bits repeatedly and they will not change - just as with quantum bits in pure states.

Remark 2.2.1. Consider the NMR quantum computer as discussed in Remarks 2.0.1 and 2.1.1. Let us assume that the spin of an atom is in some superpositioned state, i.e. a superposition of both up and down. Describing this state formally we say that the state is $a\,|\uparrow\rangle + b\,|\downarrow\rangle$ for some $a, b \in \mathbb{C}$ such that $|a|^2 + |b|^2 = 1$.

An observation of the system, using the NMR scanner, which is similar to the MRI scanners used in hospitals, would result in our knowledge that the spin of the atom is either up or down. That is, would result in $\uparrow$ with probability $|a|^2$, or with $\downarrow$ with probability $|b|^2$.

The post-observation spin of the atom would then correspond exactly to the result of the observation. Thus the post-observation state would be $|\uparrow\rangle$ if the observation resulted in $\uparrow$, or $|\downarrow\rangle$ if the observation resulted in $\downarrow$.

We can also see that given and $k_0, k_1, \ldots, k_{s-1} \in 0 \ldots n-1$ distinct, and using rudimentary probability, the probability of observing any of $x_{k_1}, x_{k_2}, \ldots, x_{k_s} \in 0 \ldots n-1$ is equal to

$$\sum_{l=1}^{s} P(x_{k_l}) = \sum_{l=1}^{s} |\alpha_{k_l}|^2.$$

We will now consider a partial measurement of a compound system. That is to say, observing one system out of a compound system. Let us have a compound state in $H_n \otimes H_m$

$$\sum_{j=1}^{n} \sum_{i=1}^{m} \alpha_{ij}\,|x_j\rangle\,|y_i\rangle\,.$$

If we observe the first system, i.e. the one on the left, we will observe some x_k. We will observe this with probability

$$P(x_k) = \sum_{i=1}^{m} P(x_k y_i) = \sum_{i=1}^{m} |\alpha_{ik}|^2.$$

The state will then change to the state that corresponds to our observation. Specifically it will be projected onto the subspace spanned by $|x_k\rangle$ and normalised to the unit norm. Thus our post-observation state is

$$\frac{1}{\sqrt{P(x_k)}} \sum_{i=1}^{m} \alpha_{ik}\,|x_k\rangle\,|y_i\rangle\,.$$

This change of state due to observation is known as the *projection postulate*. No reason for this change has been found.

2.3 Entanglement

Definition 2.3.1. Let $H_n \otimes H_m = H_{nm}$. A state $|\psi\rangle \in H_{nm}$ is *decomposable* (into H_n and H_m) if there are $|\mu\rangle \in H_n, |\nu\rangle \in H_m$ such that $|\psi\rangle = |\mu\rangle \otimes |\nu\rangle$. A state $|\psi\rangle \in H_n m$ is *entangled* if such a decomposition does not exist.

Example 2.3.1. From Example 2.1.3 we see that by construction the state (2.2) is decomposable.

Example 2.3.2. Consider the pair $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle$. This state is entangled. To prove this we argue by contradiction, and assume it is decomposable. Thus for some $a_0, a_1, b_0, b_1 \in \mathbb{C}$

$$\begin{aligned}\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle &= (a_0 |0\rangle + a_1 |1\rangle)(b_0 |0\rangle + b_1 |1\rangle) \\ &= a_0 b_0 |00\rangle + a_0 b_1 |01\rangle + a_1 b_0 |10\rangle + a_1 b_1 |11\rangle .\end{aligned}$$

Note that we omit the tensor product symbol $\otimes$. Comparing coefficients on the left and rights sides we arrive at a contradiction.

This state shows that entanglement can lead to rather bizarre results. Observation of one qubit results in 0 or 1 with equal probability. However, once this observation is made, the post-observation state of the compound system is $|00\rangle$ if we observed 0, or $|11\rangle$ if we observed 1. If we now measure the other qubit we thus find it is in exactly the same state as the first qubit was found in. The act of observation on one qubit has determined the state of the other qubit.

Remark 2.3.1. Two qubits in state as in Example 2.3.2 is known as an EPR pair, after the scientists Einstein, Podolsky and Rosen. These qubits appear to share a link - measuring one determines the other. This appears to be independent of how far apart the qubits are physically. In 1935 EPR postulated [8] that this link between the particles must be due to some property of that both of the particles have, but is unknown, i.e. some hidden variable. However in 1964 J. S. Bell [2] showed that if there is such a hidden variable, then experimental results should adhere to a particular inequality. However, repeatedly results have been found that violate this inequality, thus strongly suggesting that EPR were wrong.

Remark 2.3.2. Entanglement has been observed in particles when have been separated by distances of over 50km [15] when sent through optical cable, and 7.8km when transmitted through the atmosphere over Vienna, Austria [19]!

2.4 Representing Groups

One of the main reasons of using a computer is to work out results that actually represent something. Very often we wish states to represent members of $\mathbb{Z}_n$. If $n = 2^m$ for some m there is a very simple representation for each number. We can see that for $x \in \mathbb{Z}_{2^m}$

$$x = 2^{m-1}x_{m-1} + 2^{m-2}x_{m-2} + \ldots + 2x_1 + x_0. \tag{2.3}$$

where each $x_i \in \{0, 1\}$. The x_i form the binary representation of the number. So we then say that the basis vector in H_{2^m} that represents x is

$$|x_{m-1}\rangle |x_{m-2}\rangle \ldots |x_1\rangle |x_0\rangle ,$$

and we often just write $|x\rangle$ for this basis vector. Using this notation, we see that the set of all basis vectors (2.1) is equal to $\{|x\rangle : x \in \mathbb{Z}_{2^m}\}$.

Remember that H_{2^m} is actually the compound system of m qubits, and so observation of this system means we are actually observing m individual qubits. Each qubit will be observed in state $|0\rangle$ or $|1\rangle$, so by equation (2.3) we can deduce which member of $\mathbb{Z}_{2^m}$ the compound system of the qubits, i.e. the register, represents.

We sometimes do not care about what integer the register may represent if taken to be a binary representation of a number. For a register of length m we can say that a basis vector represents a member of $\mathbb{F}_m^2$. For example let $\mathbf{x} = (0, 1, 1, 1, 0) \in \mathbb{F}_5^2$ The basis vector in H_{2^5} that represents this is $|01110\rangle$. We normally write $|\mathbf{x}\rangle$ for this vector. Using this notation we see that the set of all basis vectors (2.1) is equal to $\{|\mathbf{x}\rangle : \mathbf{x} \in \mathbb{F}_m^2\}$.

Chapter 3

Operations on Quantum States

In this chapter we describe how the state of a quantum system changes in time. It may go without saying, but in order for a quantum computer to produce meaningful results, it needs to operate on quantum states and somehow change them.

We will also show how this time evolution can be represented using matrices, we will give some examples of time evolution, and show how multiple operations on a quantum system can be represented by diagrams.

3.1 Time Evolution

We will make the assumption that there is some function that depends on time that describes the time evolution of the system. This assumption is called the *causality principle*. More formally this means that for all $t \geq 0$ there are functions $U_t : H_n \to H_n$ such that if the state of the system at time t is $|\psi(t)\rangle$, then $|\psi(t)\rangle = U_t |\psi(0)\rangle$. We will also assume that any such time evolution must be norm preserving, i.e. $\|U_t(|\psi(t)\rangle)\| = \||\psi(t)\rangle\|$, so each U_t sends quantum states to quantum states. We will also make the hopefully reasonable assumption that for all $t_1, t_2 \geq 0$ we have $U_{t_1+t_2} = U_{t_2} U_{t_1}$. We will also make the very important assumption that each U_t is linear. To justify this assumption we would have to go more into physics than is necessary for our purposes. We ask the reader to just accept it. We say a set of maps U_t that satisfies these assumptions is called a *quantum time evolution*. Quantum time evolution satisfies an important property.

Definition 3.1.1 (Unitary)**.** A map $U : H_n \to H_n$ is *unitary* if for all $|\psi\rangle, |\phi\rangle \in H_n$ we have $\langle \psi | U\phi \rangle = \langle U^{-1}\psi | \phi \rangle$, where $\langle \, . \, , \, . \, \rangle : H_n \times H_n \to \mathbb{C}$ is the inner product of H_n.

Theorem 3.1.1. *Any quantum time evolution U_t on H_n is unitary for all $t \geq 0$.*

Proof. Immediate from the fact that any norm preserving linear map on a finite dimensional Hilbert space must be unitary. □

Informally we will take time to be discrete, as 'between operations' we will assume the system does not change. For this reason we will not refer to the time explicitly. It should be clear that, to give an example, $U_2U_1 |x\rangle$ means to apply U_1 to $|x\rangle$ at some time, and then apply U_2 at some time after to the resulting state.

Remark 3.1.1. As already mentioned in Remark 2.0.2, decoherence time must be longer than the time taken to run quantum algorithms. At the moment, decoherence time is still far too short to run anything but very short algorithms. When we later present quantum algorithms, we will *not* take into account decoherence time. This is because our aim is to show what may be possible with quantum computers, if the technical problems of actually constructing them are solved.

Remark 3.1.2. Given any unitary map we will also make the assumption that there is some way it can be implemented, even if it has not yet actually been done physically. In an NMR computer as mentioned in Remarks 2.0.1, 2.1.1 and 2.2.1, the spins of the atoms can be controlled with radio waves, and thus have *some* operations performed on them.

We would like to be able to check if certain maps can be part of a quantum time evolution. Fortunately, this is not difficult. We present the following theorem without proof.

Theorem 3.1.2. *A map $U : H_n \to H_n$ is unitary if and only if the matrix that represent is in some coordinate representation, A say, satisfies $A^*A = AA^* = I_n$, where $*$ is the complex conjugate transpose. This property is independent of the chosen coordinate representation.*

In order to use the above theorem, we need to pick a coordinate representation of H_n. We will do this in the following sections for H_2, i.e. a qubit, and for $H_2 \otimes H_2 = H_4$, i.e. a compound system of two qubits. As we will always use the same coordinate representation for H_n, we will use the same symbol for both a linear map and its matrix (and for a vector and its coordinate representation).

3.2 Unary Quantum Gates

Recall that a qubit is H_2 with orthonormal basis labelled $\{|0\rangle, |1\rangle\}$. We will assign these the natural coordinate representation

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

Given a coordinate representation we can describe a linear map on a qubit by a matrix.

Definition 3.2.1 (Unary Quantum Gate)**.** A *unary quantum gate* is a unitary linear map on one qubit, i.e. a unitary map $H_2 \to H_2$.

We now define some important unary gates

Definition 3.2.2.

$$M^{\neg} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad F_\theta = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix} \qquad H = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

We call $M^{\neg}$ the *not* gate, F_θ *phase flip* gates, and H the Hadamard gate. We also define $F = F_\pi$ as *the* phase flip gate.

The not gate maps $|0\rangle \mapsto |1\rangle$ and $|1\rangle \mapsto |0\rangle$. The phase flip gate maps $|0\rangle + |1\rangle \mapsto |0\rangle - |1\rangle$. The Hadamard gate maps $|0\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|1\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Note that in both of these mixed states, the probabilities of observing 0 or 1 are equal. What happens when we apply the Hadamard gate again is quite interesting:

$$\begin{aligned}\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) &\overset{H}{\longmapsto} \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) + \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right)\\ &= \frac{1}{2}(|0\rangle + |0\rangle - |1\rangle - |1\rangle)\\ &= |0\rangle\,.\end{aligned}$$

We see that the coefficients of $|1\rangle$ have cancelled each other out. This is known as *destructive interference*. The coefficients of $|0\rangle$ have summed together to make 1. This is known as *constructive interference*.

3.3 Binary Quantum Gates

Definition 3.3.1. A binary quantum gate is a unitary operation on two qubits, i.e. a unitary map $H_2 \otimes H_2 \to H_2 \otimes H_2$.

To describe unitary maps in the compound system $H_2 \otimes H_2$ we need give the basis, $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, a coordinate representation. We will assign

$$|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \qquad |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix} \qquad |10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \qquad |11\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}.$$

Example 3.3.1. Consider the binary gate

$$M = \begin{pmatrix}\frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}}\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ \frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{2}}\end{pmatrix}.$$

We can see that $M|00\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, and thus M can generate an EPR pair from two qubits in state $|00\rangle$.

Two unary gates, M_A acting on A, and M_B acting on another qubit B, can be seen as each acting on the compound system $A \otimes B$, and so each can in fact be seen as a binary gate. Thus the combined action of M_A followed by M_B is a binary gate, and has an associated unitary matrix. Using the matrices for M_A and M_B we can calculate this matrix explicitly using the tensor product.

Definition 3.3.2 (Tensor Product of Matrices)**.** Let A be an $r \times s$ matrix, and let B be a $t \times u$ matrix, so

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1s} \\ a_{21} & a_{22} & \dots & a_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ a_{r1} & a_{r2} & \dots & a_{rs} \end{pmatrix} \qquad B = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1u} \\ b_{21} & b_{22} & \dots & b_{2u} \\ \vdots & \vdots & \ddots & \vdots \\ b_{t1} & b_{t2} & \dots & b_{tu} \end{pmatrix}.$$

We define their tensor product to be the $rt \times st$ matrix

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1s}B \\ a_{21}B & a_{22}B & \dots & a_{2s}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{r1}B & a_{r2}B & \dots & a_{rs}B \end{pmatrix}.$$

It can be verified that the action of $M_A \otimes M_B$ acting on $A \otimes B$ is the same as M_A acting on A followed by M_B acting on B. To do this all that is required is to verify that for $x_1, x_2 \in \{0, 1\}$

$$M_A \otimes M_B \left|x_1 x_2\right\rangle = M_A \left|x_1\right\rangle \otimes M_B \left|x_2\right\rangle.$$

Note that we have defined two tensor products for vectors: Definition 2.1.2, and the above Definition 3.3.2 for the coordinate representation of a vector (since a coordinate representation of a vector is just single column matrix). However, we have chosen coordinate representations so that they are equivalent. A specific example of this is

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \left|01\right\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \left|0\right\rangle \otimes \left|1\right\rangle.$$

It can also be shown that M_A and M_B unitary implies that $M_A \otimes M_B$ is unitary. These ideas can be easily extended to general unitary maps acting on general quantum systems H_n. Thus we can say that a unitary map can act on some specific qubits of a larger quantum system H_n. The map that this actually defines on H_n is still unitary and has the same affect as just applying the map to the required qubits, apply the identity map to all the others, and then considering the final compound system. This is how we will usually think about gates acting on parts of a larger quantum system.

Using this idea of tensor product we can introduce the idea of *controlled* gates. These are gates where an actions on the basis states are defined such that the action on a qubit (the *target* qubit) is applied if and only some other qubit (the *control* qubit) is in state $\left|1\right\rangle$. We formalise this in the following definition.

Definition 3.3.3. Let A and B be qubits. Let M be a unary quantum gate acting on B. The *controlled-M* gate is the binary gate on $A \otimes B$ defined by

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes I_2 + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes M.$$

Example 3.3.2. We can find the matrix for the controlled-*not* gate. The *not* gate is

$$M_{\neg} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

so the controlled-*not* gate is

$$M_{\text{CNOT}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Thus M_{CNOT} maps $|00\rangle \mapsto |00\rangle$, $|01\rangle \mapsto |01\rangle$, $|10\rangle \mapsto |11\rangle$ and $|11\rangle \mapsto |10\rangle$.

Remark 3.3.1. Controlled-*not*, or CNOT gates have been developed. In 1999 researchers constructed a device that acted on electrons in a solid state device [26]. In this case a qubit represents the charge of an electron. In 2003 a CNOT gate was developed that acted on photons [17]. In this case a qubit represents the polarization of a photon.

3.4 Quantum Circuits

To begin with we will give a general definition of a quantum gate.

Definition 3.4.1. A quantum gate is a unitary mapping acting on a quantum register of length m that acts on a fixed number of qubits, that is independent of m.

For example, a Hadamard gate acts on one qubit, and is thus a quantum gate. No matter what size register the qubit may be part of, a Hadamard gate always acts on one qubit. A controlled-*not* gate acts on two qubits (although it always leaves one unchanged), thus a controlled-*not* gate is a quantum gate. However a map defined to be "apply the not gate to every qubit in the register," is *not* a quantum gate, as it acts on a number of qubits that is not independent of the size of the register.

For our purposes we will just roughly define a quantum circuit as some concatenation of quantum gates that act on a quantum register of some length m, i.e. acting on $H_2 \otimes H_2 \dots H_2$ (m times).

Remark 3.4.1. It is generally thought that it is easier to build quantum circuits up from smaller building blocks, i.e. gates, than to design each circuit from scratch. Thus given some unitary map $H_n \to H_n$ that would help us solve some problem, we are interested in how to decompose that map into a finite number of gates. We will do this in Chapter 7 for both quantum Fourier transforms in $\mathbb{Z}_n$ and $\mathbb{F}_m^2$.

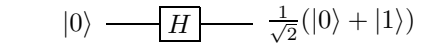

Figure 3.1: Example of Hadamard gate acting on one qubit.

We can visually represent a quantum circuit with a quantum circuit diagram. A line represents a qubit. A rectangle with any given letter, M say, on on any

number of lines, represents the action of the gate M on the qubits represented by the lines. Inputs are on the left, and outputs on the right. Figure 3.1 represents a circuit of just one gate, the Hadamard gate H, acting on a qubit with initial state $|0\rangle$. However for the *not* gate we use a slightly different visual representation, as can be seen in Figure 3.2. As in Figure 3.3, a controlled

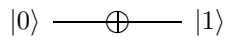

Figure 3.2: Example of a *not* gate acting on one qubit.

gate is shown with a filled black circle on the control qubit, and the ordinary symbol for the gate on the target qubit, with a line connecting the two. For a

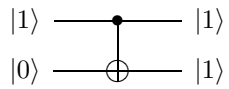

Figure 3.3: Example of a controlled-*not* gate.

measurement we will use a 'tab' symbol containing an M, and for qubits that are definitely in pure states, (or equivalently classical bits) we use a 'double

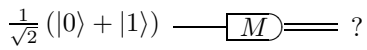

Figure 3.4: Example of a measurement. Note that for the input mixed state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, it is unknown what the result of the measurement will be. All that is known is that the result has equal probability of being $|0\rangle$ or $|1\rangle$.

line' as in Figure 3.4. We also have two important theorems relating to the construction of quantum circuits, which we leave unproved.

Theorem 3.4.1 ([1])**.** *All quantum circuits can be constructed using only controlled not and unary gates.*

Theorem 3.4.2 ([20])**.** *All quantum circuits can be constructed (in some approximated sense) using only Hadamard gates and Toffoli gates.*

A Toffoli gate is a three qubit gate that we will define in the next chapter.

Chapter 4

Boolean Circuits

We may wish to implement some classical operation using quantum gates that does not seem correspond to any unitary map. The map that the circuit in

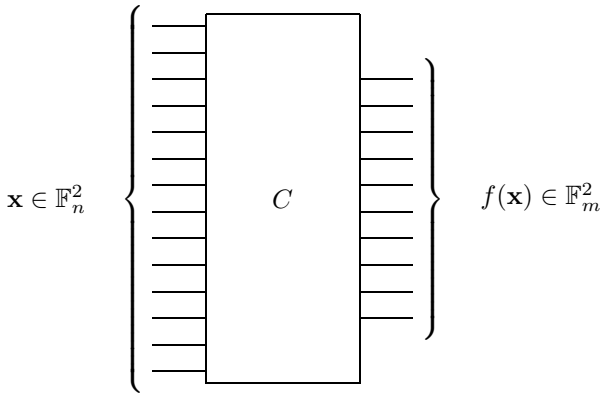

Figure 4.1: Boolean circuit performing function $f : \mathbb{F}_n^2 \to \mathbb{F}_m^2$.

Figure 4.1 performs cannot be a unitary operation (if we take the input and output to be some quantum registers in a pure state). The number of inputs do not equal the number of outputs, so the map is not invertible. This chapter shows how given any classical boolean circuit, it is in fact possible to construct a quantum circuit that performs the same function.

4.1 Boolean Circuits

For our purposes we say that a classical bit is essentially the same as a quantum bit, but that the state of a classical bit must always be a pure state, i.e. either $|0\rangle$ or $|1\rangle$. There are three boolean gates, i.e. 3 gates that act on the pure states $|0\rangle$ and $|1\rangle$, and output one of the pure states $|0\rangle$ or $|1\rangle$. They are the *and* ($\wedge$), *or* ($\vee$) and the *not* ($\neg$) gates.

A boolean circuit is some combination of these gates, and the *fanout* operation, where the output of one gate is essentially copied and used as the input to several other gates.

$\|x\rangle$	$\|y\rangle$	$\|x \wedge y\rangle$ (and)	$\|x \vee y\rangle$ (or)	$\|\neg x\rangle$ (not)
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$

Figure 4.2: Truth table for boolean gates *and*, *or* and *not*.

We consider $|0\rangle$ and $|1\rangle$ as representing members of a group, specifically $|0\rangle$ represents $0 \in \mathbb{F}_2$ and $|1\rangle$ represents $1 \in \mathbb{F}_2$ (cf Section 2.4). We also say that a basis vector $|x_0\rangle |x_1\rangle \dots |x_{m-1}\rangle$ in the compound system $H_2 \otimes \dots \otimes H_2$ (m times) represents the element $(x_0, x_1, \dots, x_{m-1}) \in \mathbb{F}_2^m$. Thus we can say that a boolean circuit actually performs a function $\mathbb{F}_2^n \to \mathbb{F}_2^m$.

Let us consider functions $\mathbb{F}_2^n \to \mathbb{F}_2^m$. In 1941 Emil Post [18][1] proved that any such function can be constructed using only the three boolean gates *and*, *or* and *not* (and *fanout*), and thus a boolean circuit can be constructed to perform this function. Thus if we can find unitary gates that perform these functions, we can show that given any function $\mathbb{F}_2^n \to \mathbb{F}_2^m$, a quantum circuit can be constructed that performs it.

The important gate that we will use is the Toffoli gate. It acts on three qubits and maps $|x\rangle |y\rangle |z\rangle \to |x\rangle |y\rangle |x \oplus (y \wedge z\rangle$ where $\wedge$ is *and* and $\oplus$ exclusive *or* (or in other words addition modulo two). It can itself be constructed using Hadamard gates, controlled phase flip gates and controlled-*not* gates as shown in Figure 4.4. The symbol for Toffoli gate is shown in Figure 4.3. We will

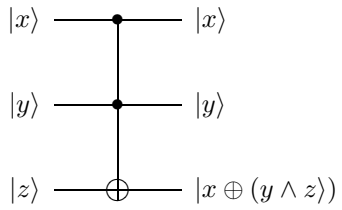

Figure 4.3: Toffoli gate.

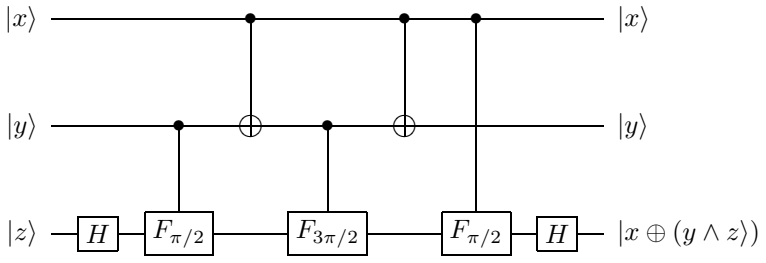

Figure 4.4: Decomposition of a Toffoli gate.

now describe how quantum gates can be used to perform the same function as boolean gates.

[1] According to [11]. Unfortunately a copy of [18] was not found.

1. **And** An *and* gate can be constructed using a Toffoli gate as in Figure 4.5. Note that an additional qubit is required in pure state $|0\rangle$, which we will call an *ancilla* qubit.

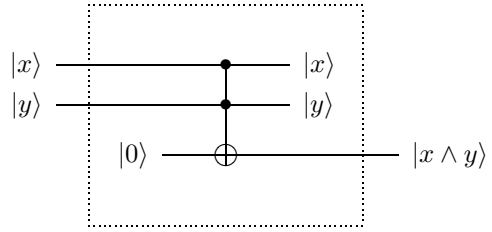

Figure 4.5: Toffoli gate as an *and* gate.

2. **Not** As shown in the previous chapter, a *not* gate is already unitary.

3. **Or** An *or* gate can be constructed using an *and* gate and three *not* gates. Thus it can be constructed with a Toffoli gate, three not gates and requires an ancilla qubit in pure state $|0\rangle$ as in Figure 4.6.

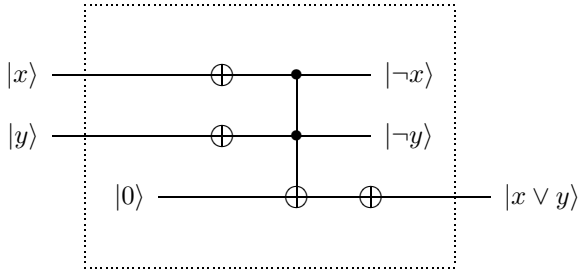

Figure 4.6: A Toffoli gate as an *or* gate.

4. **Fanout** A fanout can be constructed using a Toffoli gate and one *not* gate. Note we need two ancilla qubits to achieve this, both in pure state $|0\rangle$. As can be seen in Figure 4.7, technically we could omit the *not* gate and have one of the ancilla qubits initially in state $|1\rangle$, but it is easier to treat all ancilla qubits as having to be in state $|0\rangle$. We can see that this map actually copies basis states. We cannot have a unitary map that will copy all mixed states, as we will see in Chapter 6.

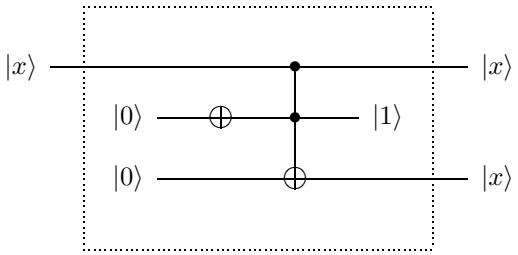

Figure 4.7: Toffoli gate as *fanout*.

Thus given any boolean circuit that computes a function $f : \mathbb{F}_n^2 \to \mathbb{F}_m^2$ with k gates we can construct a quantum circuit, R, that performs the same function. The quantum circuit uses $O(k)$ gates[2] and requires $q = O(k)$ ancilla qubits, all initially in pure state $|0\rangle$. Such a circuit would also output $m + n - q$ 'garbage qubits', and is shown in Figure 4.8.

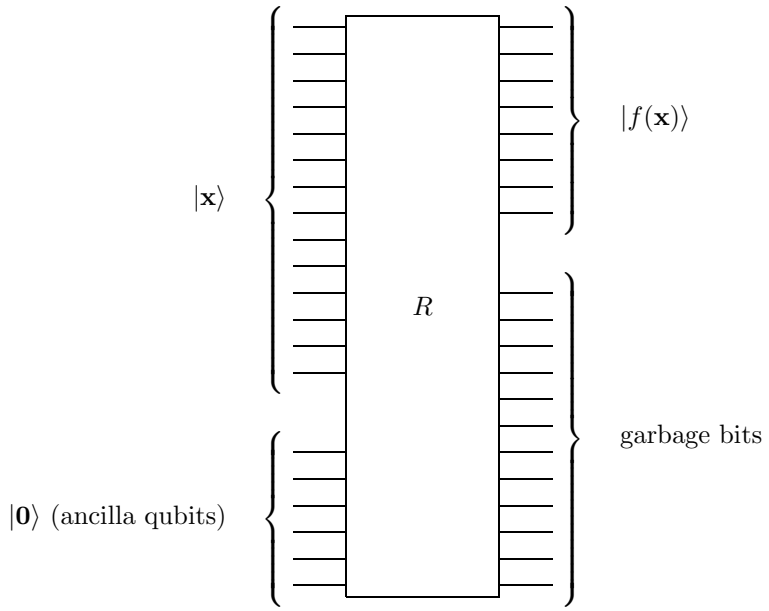

Figure 4.8: Quantum circuit emulating boolean circuit that performs function $f : \mathbb{F}_n^2 \to \mathbb{F}_m^2$.

However the garbage qubits may be undesirable. Since R is a quantum circuit it has an inverse R^{-1}. In fact, the Toffoli and *not* gates are self inverse, so we can just take the 'mirror image' of the circuit R for R^{-1}. Using this and a collection of controlled-*not* gates we can construct the following quantum circuit that requires the presence of ancilla bits, all initially in state $|0\rangle$, but outputs them all in state $|0\rangle$ (perhaps to be used again). One feature of this circuit is that the input is preserved, perhaps to be used in some other circuit. Such a circuit is shown in Figure 4.9.

[2]For functions $f, g : \mathbb{N} \to \mathbb{N}$ we say $f = O(g)$ if there are constants $c > 0$ and $n_0 \in \mathbb{N}$ such that $n > n_0$ implies $f(n) \leq cg(n)$.

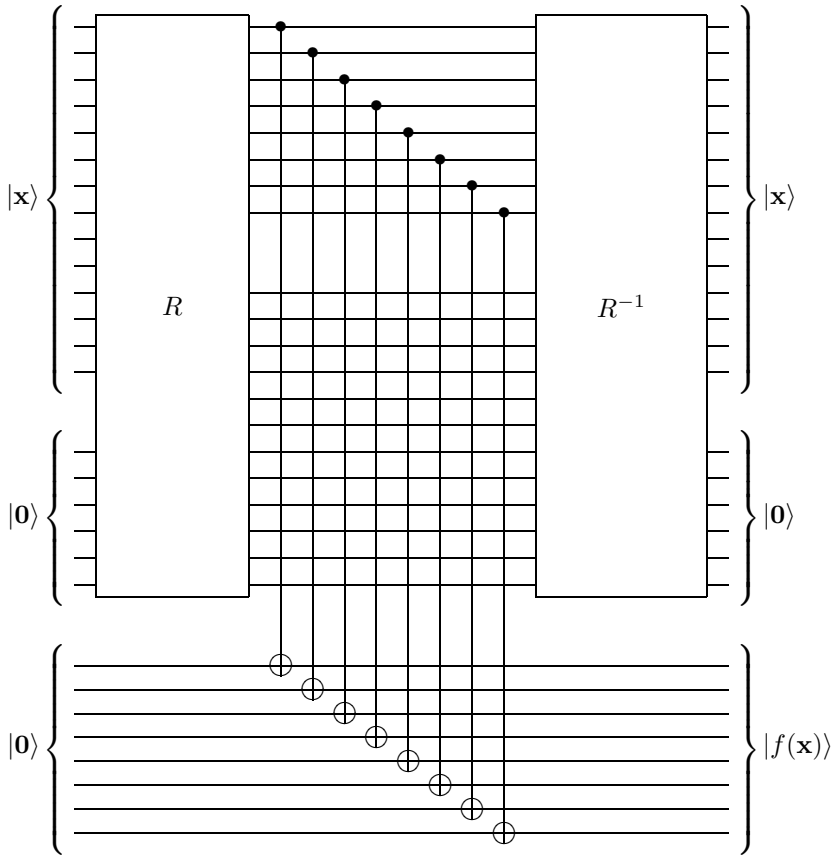

Figure 4.9: Quantum circuit emulating boolean circuit that performs function $f : \mathbb{F}_n^2 \to \mathbb{F}_m^2$. Note that this circuit preserves input and ancilla qubits.

Chapter 5

Superdense Coding

In this chapter we will describe a simple algorithm by which two classical bits of information can be encoded into one qubit, the qubit transmitted, and the original two classical bits recovered [4]. As is traditional, the sender is called Alice, and the receiver is called Bob.

For this, we assume that Alice and Bob have a quantum channel: some means of Alice to transmit a qubit to Bob. We also assume that Alice and Bob share an entangled pair of qubits in state

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

Alice's qubit is on the left in the above expression, which we will label A, and Bob's qubit is on the right, which we will label B. One could say that for Alice and Bob to share this EPR pair, they needed to have previously transmitted a qubit. However no transmission of the classical bits occurs at this point, as Alice may not have even decided what classical bits she will send to Bob when this initial transmission occurs.

We will label Alice's classical bits as a and b. Note that these are the same as quantum bits in pure states $|a\rangle$ and $|b\rangle$ respectively. The protocol follows.

1. If $a = 1$ Alice performs the phase flip F on her qubit A.

a	State after Alice's first operation
0	$\frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$
1	$\frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$

2. If b Alice also performs the *not* operation on her qubit A.

a	b	State after Alice's second operation
0	0	$\frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$
0	1	$\frac{1}{\sqrt{2}}(\|10\rangle + \|01\rangle)$
1	0	$\frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$
1	1	$\frac{1}{\sqrt{2}}(\|10\rangle - \|01\rangle)$

3. Alice transmits her qubit to Bob.

4. Bob performs the controlled-*not* on his qubit B using the qubit A as the control qubit.

a	b	State after Bob's first operation
0	0	$\frac{1}{\sqrt{2}}(\|00\rangle + \|10\rangle)$
0	1	$\frac{1}{\sqrt{2}}(\|11\rangle + \|01\rangle)$
1	0	$\frac{1}{\sqrt{2}}(\|00\rangle - \|10\rangle)$
1	1	$\frac{1}{\sqrt{2}}(\|11\rangle - \|01\rangle)$

5. Bob performs the Hadamard Transform H on A. At this point the qubits are guaranteed to be in state $|ab\rangle$.

6. Bob observes the two qubits A and B. He is certain of observing a and b.

Note that this process has *copied* the classical bits a and b. Both Alice and Bob each have a copy of a and b at the end of this protocol. Since we can view classical bits as quantum bits in pure states, this process has essentially copied two pure quantum states. It is impossible to copy a general mixed state using unitary maps, as will be proved in the next chapter.

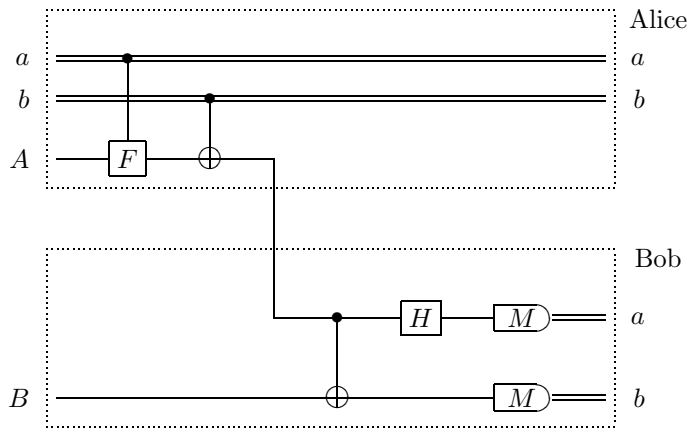

Figure 5.1: Superdense coding.

Chapter 6

Quantum Teleportation

In this chapter we will describe the converse to superdense coding: how the state of a qubit can somehow be 'encoded' into two classical bits, the bits transmitted, and the original state recovered. As in the previous chapter the sender with the original state to send is called Alice, and the receiver Bob. This protocol was introduced in [3].

For this we will assume that Alice and Bob have a classical channel: some means for Alice to send classical bits to Bob.

Remark. A classical channel is any traditional means of sending information: fax, e-mail - even carrier pigeon!

Remark. Quantum teleportation has been successfully carried out a number of times, teleporting states of atoms or photons. Most notably was a 600m teleportation of photons across the River Danube, Vienna Austria in 2004 [22]. We will not discuss the issue in any depth, but quantum teleportation is crucial in quantum communication schemes.

Also as before we will start with Alice and Bob sharing an EPR pair

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{6.1}$$

One could say that to get to this point that Alice and Bob had to share a quantum channel. However, this could have happened a long time before Alice had the qubit to teleport to Bob.

One very important point to make is that this process *teleports* the quantum state. The encoding process actually destroys the original quantum state. There is actually no way to copy a general quantum state using unitary mappings as we will see later in this chapter.

In the EPR pair (6.1), Alice has the qubit on the left which we will label A, and Bob has the qubit on the right which we will call B. Alice's qubit to be teleported we will label T. It is in (possibly) a mixed state which we will refer to as $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$ for some unknown $a, b \in \mathbb{C}$ where $|a|^2 + |b|^2 = 1$. Thus the compound state of the system $T \otimes A \otimes B$ is

$$\begin{aligned} & (a\,|0\rangle + b\,|1\rangle)\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ = \quad & \frac{a}{\sqrt{2}}\,|000\rangle + \frac{a}{\sqrt{2}}\,|011\rangle + \frac{b}{\sqrt{2}}\,|100\rangle + \frac{b}{\sqrt{2}}\,|111\rangle\,. \end{aligned}$$

The quantum teleportation protocol follows. It is also shown as a quantum circuit diagram in Figure 6.1.

1. Alice performs the controlled-*not* operation on her qubit A, using the qubit to be teleported T as the control qubit. Thus the compound state becomes

$$\frac{a}{\sqrt{2}}|000\rangle + \frac{a}{\sqrt{2}}|011\rangle + \frac{b}{\sqrt{2}}|110\rangle + \frac{a}{\sqrt{2}}|101\rangle$$

2. Alice performs the Hadamard transform H on the qubit to be teleported T. Thus the state becomes

$$\begin{aligned} & \frac{1}{2}|00\rangle(a|0\rangle + b|1\rangle) + \frac{1}{2}|01\rangle(a|1\rangle + b|0\rangle) \\ + \quad & \frac{1}{2}|10\rangle(a|0\rangle - b|1\rangle) + \frac{1}{2}|00\rangle(a|1\rangle - b|0\rangle). \end{aligned}$$

3. Alice measures both qubits A to get b_A, T to get b_T, where b_A and b_T are classical bits. There are 4 possible outcomes, each with equal probability of $\frac{1}{4}$.

b_A	b_T	State after Alice's observation
0	0	$\|00\rangle(a\|0\rangle + b\|1\rangle)$
0	1	$\|01\rangle(a\|1\rangle + b\|0\rangle)$
1	0	$\|10\rangle(a\|0\rangle - b\|1\rangle)$
1	1	$\|11\rangle(a\|1\rangle - b\|0\rangle)$

4. Alice transmits the classical bits to Bob

5. If $b_A = 1$, Bob performs the not operation on his qubit B.

b_T	b_A	State after Bob's first operation
0	0	$\|00\rangle(a\|0\rangle + b\|1\rangle)$
0	1	$\|01\rangle(a\|0\rangle + b\|1\rangle)$
1	0	$\|10\rangle(a\|0\rangle - b\|1\rangle)$
1	1	$\|11\rangle(a\|0\rangle - b\|1\rangle)$

6. If $b_T = 1$, Bob performs the phase flip operation F on his qubit B. Now Bob's qubit B is in state $a|0\rangle + b|1\rangle = |\psi\rangle$.

At the end of the above protocol we only have one qubit in state $|\psi\rangle$. The original qubit in state $|\psi\rangle$ was measured and so its state collapsed to a pure state. As we will see there is no way we can have an algorithm to copy any general state $|\psi\rangle$. We will now formalise the idea of copying, and show that it is impossible.

Let us consider the space H_n with orthonormal basis $\{|x_0\rangle, |x_1\rangle \ldots |x_{n-1}\rangle\}$.

Definition 6.0.1 (Quantum Copy Machine)**.** A unitary map U in $H_n \otimes H_n$ is called a quantum copy machine if for all $|\psi\rangle \in H_n$

$$U(|\psi\rangle|x_0\rangle) = |\psi\rangle|\psi\rangle.$$

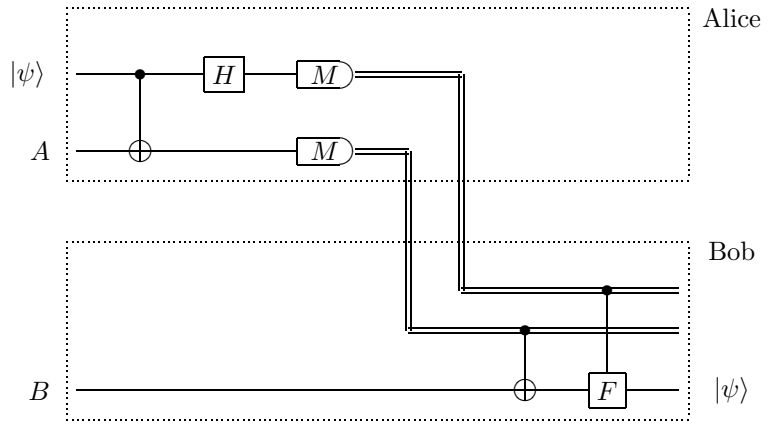

Figure 6.1: Quantum teleportation.

Using this definition we now give the following result [25].

Theorem 6.0.1 (No-Cloning Theorem)**.** *For $n > 1$ there does not exist a quantum copy machine.*

Proof. Let us assume that $n > 1$ and that a quantum copy machine, U, exists. Since $n > 1$ there must be two orthonormal basis vectors $|x_0\rangle$ and $|x_1\rangle$. Let $|\psi\rangle = \frac{1}{\sqrt{2}}(|x_0\rangle + |x_1\rangle)$. Thus by definition of quantum copy machine

$$\begin{aligned} U\left(|\psi\rangle|x_0\rangle\right) &= |\psi\rangle|\psi\rangle \\ &= \frac{1}{2}\left(|x_0\rangle|x_0\rangle + |x_0\rangle|x_1\rangle + |x_1\rangle|x_0\rangle + |x_1\rangle|x_1\rangle\right). \end{aligned}$$

However U is linear so

$$\begin{aligned} U(|\psi\rangle|x_0\rangle) &= U\left(\frac{1}{\sqrt{2}}\left(|x_0\rangle + |x_1\rangle\right)|x_0\rangle\right) \\ &= \frac{1}{\sqrt{2}}U(|x_0\rangle|x_0\rangle) + \frac{1}{\sqrt{2}}U(|x_1\rangle|x_0\rangle) \\ &= \frac{1}{\sqrt{2}}|x_0\rangle|x_0\rangle + \frac{1}{\sqrt{2}}|x_1\rangle|x_1\rangle \end{aligned}$$

This is a contradiction. □

Thus by the No-Cloning Theorem there does not exist any combination of unitary maps so that an arbitrary quantum state can be copied. Note however that the No-Cloning Theorem does not forbid a combination of unitary maps in $H_n \otimes H_n$ to copy basis states. In fact we have already seen two examples of this: superdense coding and the use of the Toffoli gate for fanout of classical bits (remember that a classical bit is essentially the same as quantum bit in a pure state).

Chapter 7

Quantum Fourier Transform

In this chapter we investigate an important map of quantum systems, the quantum Fourier transform. As we will see this map is a unitary linear map, and we shall find how to decompose it into a finite number of quantum gates. This map is used in all of the following chapters describing quantum algorithms.

7.1 Discrete Fourier Transform

We will in this section extremely briefly, and largely without proof, describe what we mean by *characters* and *discrete Fourier transform*. Throughout this chapter G will be a finite abelian group, and $n = |G|$. We will also say $G = \{g_1, g_2, \ldots, g_n\}$. The group operation in G will be written additively.

7.1.1 Characters of Finite Abelian Groups

Definition 7.1.1. A *character* of G is a homomorphism $\chi : G \to \mathbb{C}\backslash\{0\}$.

Note that for all $g \in G$ and characters χ of G, $\chi(g)$ is an nth root of unity. Since χ is a homomorphism and $ng = 1_G$, we have $\chi(g)^n = \chi(ng) = \chi(1_G) = 1$.

We can see that the characters themselves form an abelian group, with the product (we will use multiplicative notation) of two characters χ_α and χ_β is defined by requiring it to satisfy $\chi_\alpha\chi_\beta(g) = \chi_\alpha(g)\chi_\beta(g)$ for all g in G. We call this group either the *character group* or the *dual group* of G, and write it as $\widehat{G}$. We then have the following result, which we state without proof.

Lemma 7.1.1. *Let G be a finite abelian group. Then $G \cong \widehat{G}$.*

One thing the above lemma implies is that there are exactly the same number of characters as elements of the group, i.e. there is a one to one correspondence between the two. Let us consider the cyclic group $\mathbb{Z}_n$. For each fixed $y \in \mathbb{Z}_n$ define χ_y by

$$\chi_y(x) = e^{\frac{2\pi i x y}{n}}.$$

There is a one to one correspondence between Z_n and all the characters of $\mathbb{Z}_n$. Therefore each character of $\mathbb{Z}_n$ is equal to χ_y for some unique $y \in \mathbb{Z}_n$.

We can also find the characters of $\mathbb{F}_2^m$. We know that

$$\mathbb{F}_2^m = \mathbb{F}_2 \oplus \mathbb{F}_2 \oplus \cdots \oplus \mathbb{F}_2 \ (m \text{ times}). \tag{7.1}$$

Lemma 7.1.1 shows that the character group of $\mathbb{F}_2^m$ must be isomorphic to $\mathbb{F}_2^m$ itself. Thus by equation (7.1) each character of $\mathbb{F}_2^m$ must be equal to the product of m characters of $\mathbb{F}_2$, and given m characters of $\mathbb{F}_2$ their product must be equal to a character of $\mathbb{F}_2^m$. Now note that $\mathbb{F}_2 \cong \mathbb{Z}_2$. Each character of $\mathbb{Z}_2$ is given by, for some $y \in \mathbb{Z}_2$,

$$\chi_y(x) = e^{\frac{2\pi i x y}{n}} = (-1)^{xy}.$$

The decomposition (7.1) means that for each $\mathbf{x} \in \mathbb{F}_2^m$ there are $x_i \in \mathbb{F}_2$ such that $\mathbf{x} = (x_1, x_2, \ldots, x_m)$. Therefore each character of $\mathbb{F}_2^m$ is determined by $\mathbf{y} \in \mathbb{F}_2^m$ so that

$$\chi_{\mathbf{y}}(\mathbf{x}) = (-1)^{\mathbf{x}.\mathbf{y}},$$

where . is the standard inner product.

7.1.2 Discrete Fourier Transform

Let us consider the vector space V of functions $f : G \to \mathbb{C}$. We assign to V an inner product defined by

$$\langle f_1 | f_2 \rangle = \sum_{k=1}^{n} f_1^*(g_k) f_2(g_k).$$

It can be shown that the set

$$\left\{ \frac{1}{\sqrt{n}}\chi_1, \frac{1}{\sqrt{n}}\chi_2, \ldots, \frac{1}{\sqrt{n}}\chi_n \right\},$$

is a basis of this space, which we will call the *character basis*. Therefore for each function $f : G \to \mathbb{C}$ there exist (unique) coefficients $\widehat{f}_i \in \mathbb{C}$, the *fourier coefficients*, such that

$$f = \widehat{f}_1 \frac{1}{\sqrt{n}}\chi_1 + \widehat{f}_2 \frac{1}{\sqrt{n}}\chi_2 + \ldots + \widehat{f}_n \frac{1}{\sqrt{n}}\chi_n.$$

Definition 7.1.2. Given a function $f : G \to \mathbb{C}$, the *discrete Fourier transform* of f is the function $\widehat{f} : G \to \mathbb{C}$ defined by

$$\widehat{f}(g_i) = \widehat{f}_i,$$

where $\widehat{f}_i$ is defined above.

Given the values of $f(g_i)$ and the character of the group χ_i there is an easy way to find the Fourier transform (which can be easily derived using the fact

that the characters form a basis of the space, and so must be orthogonal with respect to the inner product). We have

$$\widehat{f}(g_i) = \frac{1}{\sqrt{n}} \sum_{k=1}^{n} \chi_i^*(g_k) f(g_k).$$

Because we have found all of the characters of $\mathbb{F}_2^m$ and $\mathbb{Z}_n$, we can simply substitute them into the above equation to find the Fourier transform for any functions $\mathbb{F}_2^m \to \mathbb{C}$ and $\mathbb{Z}_n \to \mathbb{C}$. Let $f : \mathbb{Z}_n \to \mathbb{C}$. Then the Fourier transform of f is

$$\widehat{f}(x) = \frac{1}{\sqrt{n}} \sum_{y \in \mathbb{Z}_n} e^{-\frac{2\pi i x y}{n}} f(y).$$

Let $f : \mathbb{F}_2^m \to \mathbb{C}$. Then the Fourier transform of f is

$$\widehat{f}(\mathbf{x}) = \frac{1}{\sqrt{2^m}} \sum_{\mathbf{y} \in \mathbb{F}_n^2} (-1)^{\mathbf{x}.\mathbf{y}} f(\mathbf{y}).$$

There is also an inverse to the Fourier transform, which we now define.

Definition 7.1.3. Given a function $f : G \to \mathbb{C}$, the *inverse Fourier transform* of f is defined to be the function $\widetilde{f} : G \to \mathbb{C}$

$$\widetilde{f}(g_i) = \frac{1}{\sqrt{n}} \sum_{k=1}^{n} \chi_k(g_i) f(g_k).$$

If we consider both the forward and inverse transforms for all g_i and write them out as matrix vector equations, we can see that

$$\widehat{\widetilde{f}} = \widetilde{\widehat{f}}.$$

We can see that the inverse Fourier transform in $\mathbb{Z}_n$ looks very similar to the forward transform

$$\widehat{f}(x) = \frac{1}{\sqrt{n}} \sum_{y \in \mathbb{Z}_n} e^{\frac{2\pi i x y}{n}} f(y).$$

The inverse transform in $\mathbb{F}_2^m$ is exactly equal to the forward transform, i.e. it is self inverse.

We also have an identity that is very important. This will be used to show the quantum Fourier transform, that we will define later, is a unitary map. We define the norm on V by $\|f\| = \sqrt{\langle f|f\rangle}$. We leave the following theorem unproved.

Theorem 7.1.1 (Pareseval's Identity)**.**

$$\|f\| = \|\widehat{f}\|$$

It should be noted that Fourier transforms are known to be able to 'extract' the periods of functions. We use this fact in Chapter 9 to factor integers.

7.2 Quantum Fourier Transform

Let $G = \{g_1, g_2, \ldots, g_n\}$ be a finite abelian group, and let $\{\chi_1, \chi_2, \ldots, \chi_n\}$ be the set of characters of G. We will also let H_n be an n-dimensional quantum system. We label an orthogonal basis of H_n by $\{|g_1\rangle, |g_2\rangle, \ldots, |g_n\rangle\}$. Thus we say that H_n *represents* G (see section 2.4).

We also note that any quantum state

$$\sum_{i=1}^{n} \alpha_i |g_i\rangle \text{ where } \sum_{i=1}^{n} |\alpha_i|^2 = 1 \tag{7.2}$$

defines a map

$$f : G \to \mathbb{C} \text{ with } f(g_i) = \alpha_i \text{ and } \|f\| = 1. \tag{7.3}$$

We can see that given map of the form of (7.3) defines a quantum state of the form of (7.2). Thus a general quantum state can be written as

$$\sum_{i=1}^{n} f(g_i) |g_i\rangle \text{ where } f : G \to \mathbb{C} \text{ and } \|f\| = 1. \tag{7.4}$$

Definition 7.2.1. The *quantum Fourier transform* (QFT) in G is the map on H_n defined by

$$\sum_{i=1}^{n} f(g_i) |g_i\rangle \xmapsto{QFT_G} \sum_{i=1}^{n} \widehat{f}(g_i) |g_i\rangle , \tag{7.5}$$

where $\widehat{f}$ is the ordinary discrete Fourier transform of f.

Lemma 7.2.1. *The quantum Fourier transform is a unitary map on H_n.*

Proof. It is clear by the definition that the quantum Fourier transform is linear. By Parseval's identity 7.1.1 we have that $\|\widehat{f}\| = \|f\|$. Thus Fourier transform is a norm preserving linear map on a finite dimensional Hilbert space, and so must be unitary. □

Recall that the Fourier transform of f is

$$\widehat{f} = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \chi_i^*(g_i) f(g_i),$$

so for basis vectors $|g_i\rangle$ the QFT is

$$|g_i\rangle \xmapsto{QFT_G} \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \chi_i^*(g_i) |g_i\rangle . \tag{7.6}$$

Since QFT is linear if we find a linear map that performs the above operation on basis states $|g_i\rangle$, then it would perform the general QFT (7.5) on general mixed states.

It should be noted that the QFT depends on the group G. If we took that H_n represents some other group, then the Fourier transform of that group would be different, and so the quantum Fourier transform on H_n would be different. We use QFT when we take H_m to represent either $\mathbb{F}_2^m$ or $\mathbb{Z}_{2^m}$.

7.2.1 Quantum Fourier Transform in $\mathbb{F}_2^m$

From Section 7.1.1 we see that the characters of $\mathbb{F}_2^m$ are $\chi_y(\mathbf{x}) = (-1)^{\mathbf{x}\cdot\mathbf{y}}$. Thus if we label an orthonormal basis of H_{2^m} as $\{|\mathbf{x}\rangle : \mathbf{x} \in \mathbb{F}_2^m\}$ the QFT in $\mathbb{F}_2^m$ on H_{2^m} on a basis vector is

$$|\mathbf{x}\rangle \xmapsto{QFT_{\mathbb{F}_2^m}} \frac{1}{\sqrt{2^m}} \sum_{\mathbf{y}\in\mathbb{F}_2^m} (-1)^{\mathbf{x}\cdot\mathbf{y}} |\mathbf{y}\rangle . \tag{7.7}$$

We recall that H_{2^m} is a quantum register of length m. That is, the compound system of m qubits. We can show this in general circuit diagram Figure 7.1. Note that technically the diagram does *not* show describe a quantum circuit. A quantum circuit is defined to be some collection of quantum gates, and a quantum gate is defined to act on some finite number of qubits that is independent of the size of the register. However, we can, and will now, decompose this

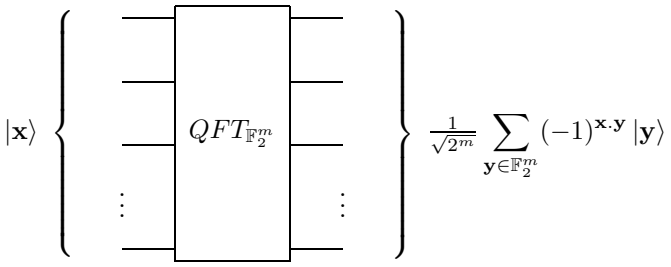

Figure 7.1: Quantum Fourier transform in $\mathbb{F}_2^m$.

transform into some finite number of quantum gates.

Let us first consider a more general case. Let G be a finite abelian group such that $G = U \oplus V$. Let $U = \{u_1, u_2, \ldots, u_s\}$ and $V = \{v_1, v_2, \ldots, v_s\}$. Let H_r represent U and H_s represent V. By this we mean that we label an orthonormal basis of H_r by $\{|u_1\rangle, |u_2\rangle, \ldots, |u_r\rangle\}$ and we label an orthonormal basis of H_s by $\{|v_1\rangle, |v_2\rangle, \ldots, |v_s\rangle\}$. Since $G = U \oplus V$ all members of G can be uniquely expressed as $g_{ij} = u_i + v_j$. Thus the vector $|u_i\rangle |v_j\rangle \in H_r \otimes H_s = H_{rs}$ can be thought of as representing the unique g_{ij} such that $g_{ij} = u_i + v_j$, so we define $|g_{ij}\rangle = |u_i\rangle |v_j\rangle$. Thus H_{rs} represents G.

By Lemma 7.1.1 we can know that $\widehat{G} = \widehat{U} \otimes \widehat{V}$. Thus any character of G can be uniquely written as $\chi_{kl}(g_{ij}) = \chi_k^U(u_i)\chi_l^V(v_j)$, for characters χ_k^U and χ_l^V of U and V respectively. -

Lemma 7.2.2. *Using the above notation, the map*

$$|g_{ij}\rangle \longmapsto \Big(\frac{1}{\sqrt{r}} \sum_{k=1}^{r} (\chi_k^U(u_i))^* |u_k\rangle\Big)\Big(\frac{1}{\sqrt{s}} \sum_{l=1}^{s} (\chi_l^V(v_j))^* |v_l\rangle\Big)$$

is the QFT in G on H_{rs}.

Proof. If we expand out the RHS using the definition of tensor product, we get

$$\begin{aligned}\text{RHS} &= \Big(\frac{1}{\sqrt{rs}}\sum_{k=1}^{r}\sum_{l=1}^{s}(\chi_k^U(u_i))^*(\chi_l^V(v_i))^*\,|u_k\rangle\,|v_l\rangle\Big)\\ &= \Big(\frac{1}{\sqrt{rs}}\sum_{k=1}^{r}\sum_{l=1}^{s}\chi_{kl}^*(u_i+u_j)\,|u_k+v_l\rangle\Big)\\ &= \Big(\frac{1}{\sqrt{rs}}\sum_{k=1}^{r}\sum_{l=1}^{s}\chi_{kl}^*(g_{ij})\,|g_{ij}\rangle\Big),\end{aligned}$$

Which is exactly the quantum Fourier transform of in G of $|g_{ij}\rangle$. □

We can use this lemma to decompose the QFT in $\mathbb{F}_2^m$. Recall that for $|\mathbf{x}\rangle \in H_{2^m}$ there are (unique) $|x_1\rangle, |x_2\rangle, \ldots, |x_m\rangle \in H_2$ such that

$$|\mathbf{x}\rangle = |x_1\rangle\,|x_2\rangle \ldots |x_m\rangle\,.$$

We see that the decomposition is in fact very simple.

Definition 7.2.2. The Hadamard transform is defined as $H_m = H \otimes H \otimes \ldots H$ (m times).

Thus using the above notation $H_m\,|\mathbf{x}\rangle = (H\,|x_1\rangle)(H\,|x_2\rangle)\ldots(H\,|x_m\rangle)$ (m times).

Remark. We have now defined H_m to be both the Hadamard transform on a Hilbert space of dimension m, or the Hilbert space of dimension m itself. In any particular case it should hopefully be clear from context whether we mean the Hilbert space or the map.

Theorem 7.2.1. *The QFT in $\mathbb{F}_2^m$ on H_{2^m} is the map*

$$|\mathbf{x}\rangle \longmapsto H_m\,|\mathbf{x}\rangle\,.$$

Proof. We have that $\mathbb{F}_2^m = \mathbb{F}_2 \oplus \mathbb{F}_2 \oplus \ldots \oplus \mathbb{F}_2$ (m times). Thus by Lemma 7.2.2 if we can show that for map

$$|x_i\rangle \longmapsto H\,|x_i\rangle$$

is the QFT in $\mathbb{F}_2$ on H_2 we are done. However, this is immediate by taking $m = 1$ in (7.7). □

Note that have shown that for $|\mathbf{x}\rangle \in H_{2^m}$, we have

$$H_m\,|\mathbf{x}\rangle = \frac{1}{\sqrt{2^m}}\sum_{\mathbf{y}\in\mathbb{F}_2^m}(-1)^{\mathbf{x}\cdot\mathbf{y}}\,|\mathbf{y}\rangle\,.$$

Thus if we take $|\mathbf{x}\rangle = |\mathbf{0}\rangle$,

$$H_m\,|\mathbf{0}\rangle = \frac{1}{\sqrt{2^m}}\sum_{\mathbf{y}\in\mathbb{F}_2^m}|\mathbf{y}\rangle\,,$$

which is an equal superposition of all the basis states of H_{2^m}. This is a frequently used map. We will use it in the chapters on random numbers, factoring and searching.

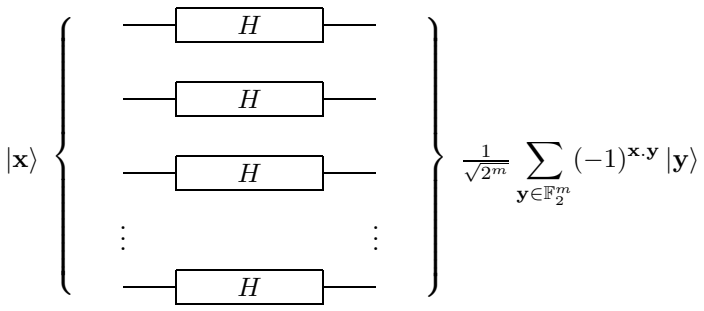

Figure 7.2: Decomposition of QFT in $\mathbb{F}_2^m$.

7.2.2 Quantum Fourier Transform in $\mathbb{Z}_n$

From Section 7.1.1 we know that the characters of $\mathbb{Z}_n$ are of the form $\chi_y(x) = e^{\frac{2\pi xy}{n}}$. Recall also from section 2.4 that if $n = 2^m$ then $\mathbb{Z}_n$ can be very naturally represented by H_{2^m} (i.e. m qubits). That is, given $x \in \mathbb{Z}_{2^m}$, we see that there are $x_i \in \{0, 1\}$ such that

$$x = 2^{m-1}x_{m-1} + 2^{m-2}x_{m-2} + \ldots + 2x_1 + x_0.$$

Thus we define $|x\rangle$ to be $|x_{m-1}\rangle |x_{m-2}\rangle \ldots |x_0\rangle$ and say that $|x\rangle$ represents x. We will actually be working with the *inverse* quantum Fourier transform. We do this for two reasons. Firstly in the next chapter, Shor's algorithm uses the inverse quantum Fourier transform. Secondly it is ever so slightly less confusing working with the inverse transform (as there are less '$-$'s in our expressions). However, the expressions are all very similar for the forward transform - it should not be too difficult to do though all the stages of the decomposition, putting in '$-$'s where appropriate to get the decomposition for the forward quantum Fourier transform.

By Section 7.1.2 we know that the inverse quantum Fourier transform in $\mathbb{Z}_{2^m}$ on H_{2^m} is the map

$$|x\rangle \xmapsto{QFT^{-1}_{\mathbb{Z}_n}} \frac{1}{\sqrt{2^m}} \sum_{y=0}^{2^m-1} e^{\frac{2\pi xy}{2^m}} |y\rangle$$

We know that this is linear, and unitary, but we would like to be able to decompose it to some finite number of quantum gates. We will use the following two lemmata

Lemma 7.2.3.

$$\sum_{y=0}^{2^m-1} e^{\frac{2\pi xy}{2^n}} |y\rangle = (|0\rangle + e^{\frac{\pi ix}{2^0}} |1\rangle)(|0\rangle + e^{\frac{\pi ix}{2^1}} |1\rangle) \ldots e^{\frac{\pi ix}{2^1}} |m-1\rangle) \tag{7.8}$$

Proof. Note for each $y \in \mathbb{Z}_{2^m}$ there are $y_i \in \{0, 1\}$ such that

$$y = 2^{m-1}y_{m-1} + 2^{m-2}y_{m-2} + \ldots + 2y_1 + y_0.$$

For each $y \in \mathbb{Z}_{2^m}$ let $y' \in \mathbb{Z}_{2^{m-1}}$ such that

$$y' = 2^{m-2}y_{m-1} + 2^{m-3}y_{m-2} + \ldots + 1y_1,$$

so $y = 2y' + y_0$. Thus we can split up the sum

$$\begin{aligned}\sum_{y=0}^{2^m-1} e^{\frac{2\pi xy}{2^n}} |y\rangle &= \sum_{y'=0}^{2^{m-1}-1} e^{\frac{2\pi ix(2y'+0)}{2^m}} |2y'+0\rangle + \sum_{y'=0}^{2^{m-1}-1} e^{\frac{2\pi ix(2y'+1)}{2^m}} |2y'+1\rangle \\ &= \sum_{y'=0}^{2^{m-1}-1} e^{\frac{2\pi ix2y'}{2^m}} |y'\rangle |0\rangle + \sum_{y'=0}^{2^{m-1}-1} e^{\frac{2\pi ix2y'}{2^m}} e^{\frac{2\pi ix}{2^m}} |y'\rangle |1\rangle \\ &= \sum_{y'=0}^{2^{m-1}-1} e^{\frac{2\pi ixy')}{2^{m-1}}} |y'\rangle (|0\rangle + e^{\frac{\pi ix}{2^{m-1}}} |1\rangle)\end{aligned}$$

We can continue applying the same method to the sum on the RHS. □

Lemma 7.2.4. *Let $x \in \mathbb{Z}_{2^m}$, with $x_i \in \{0,1\}$ such that*

$$x = 2^{m-1}x_{m-1} + 2^{m-2}x_{m-2} + \ldots + 2x_1 + x_0.$$

Then

$$\exp(\frac{\pi ix}{2^{l-1}}) = (-1)^{x_{l-1}} \exp(\frac{\pi ix_{l-2}}{2^1}) \ldots \exp(\frac{\pi ix_1}{2^{l-2}}) \exp(\frac{\pi ix_0}{2^{l-1}})$$

Proof.

$$\begin{aligned}\exp(\frac{\pi ix}{2^{l-1}}) &= \exp(\frac{\pi i(2^{m-1}x_{m-1} + 2^{m-2}x_{m-2} + \ldots + 2x_1 + x_0)}{2^{l-1}}) \\ &= \exp(\frac{\pi i(2^{l-1}x_{l-1} + 2^{l-2}x_{l-2} + \ldots + 2x_1 + x_0)}{2^{l-1}}) \\ &\quad \text{(since exp is } 2\pi\text{-periodic)} \\ &= (-1)^{x_{l-1}} \exp(\frac{\pi ix_{l-2}}{2^1}) \ldots \exp(\frac{\pi ix_1}{2^{l-2}}) \exp(\frac{\pi ix_0}{2^{l-1}})\end{aligned}$$

□

Using these two lemmata we can show how to compute the quantum Fourier transform in $\mathbb{Z}_{2^m}$ on H_{2^m}. Firstly we swap the order of the qubits, so

$$|x_{m-1}\rangle |x_{m-2}\rangle \ldots |x_1\rangle |x_0\rangle \longmapsto |x_0\rangle |x_1\rangle |x_{m-2}\rangle |x_{m-1}\rangle$$

A swap of the state of two qubits can be achieved by three controlled-*not* gates, as in Figure 7.3. Now what we want to do is to apply unitary maps on the

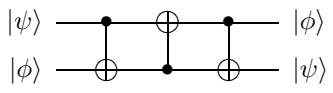

Figure 7.3: Swap of two qubits.

qubits so that the expression has the form given by equation (7.8), multiplied by a factor $1/\sqrt{2^m}$. This means that for the lth qubit from the left we want the coefficient of $|0\rangle$ to be $1/\sqrt{2^m}$ and the coefficient of $|1\rangle$ to be $(1/\sqrt{2^m})e^{\frac{\pi ix}{2^{l-1}}}$. Let us label the qubits from left to right as $A_0, A_1, \ldots, A_{m-1}$. We perform the following steps for $l = m-1 \ldots 1$ (in this order)

1. Notice qubit A_l is in state $|x_l\rangle$.

2. We apply the Hadamard Walsh gate to A_l, so

$$|x_l\rangle \longmapsto \frac{1}{\sqrt{2}}(|0\rangle + ((-1)^{x_l}\,|1\rangle)$$

3. For each $k = l \ldots 0$ apply to A_l the phase flip ϕ_{lk} defined as

$$\phi_{lk} = F_{\frac{\pi}{2^{l-k}}} = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{\pi i}{2^{l-k}}} \end{pmatrix}$$

but *controlled* with the control qubit A_k.

4. By Lemma (7.2.4) qubit A_l is now in state.

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{\pi i x}{2^{l-1}}}\,|1\rangle)$$

Once all these steps are completed, by Lemma (7.8) the QFT in $\mathbb{Z}_{2^m}$ has been performed. A circuit diagram for these steps is shown in Figure 7.4 (the swapping of the bits has been omitted, as have the subscripts for ϕ).

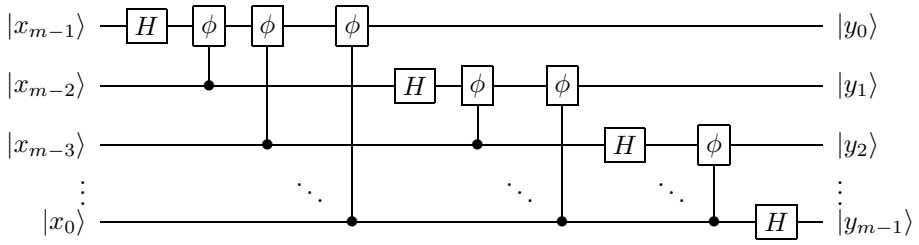

Figure 7.4: Decomposition of QFT in $\mathbb{Z}_{2^m}$.

Chapter 8

Random Numbers

Random numbers are quite useful in various computer calculations. For example they can assist is finding large multidimensional integrals, or in optimization processes using Monte Carlo methods. Random numbers produced on a standard computer are not really random, they are *pseudorandom*. They are deterministic - you put in the same input, i.e. the seed, and the generator will return the same output every time. This means that there is an underlying pattern to the numbers produced. This pattern may be subtle, but it still adversely affects the algorithms that require random numbers.

There are commercially available devices to generate random numbers that exploit quantum effects, such as the chance that a photon will go through a semi-silvered mirror, or the decay time for some radioactive isotope. However it is interesting to see how a quantum circuit could be constructed to generate random numbers. We will see that we already have already discussed all the required tools to do so.

Let us take H_m to represent $\mathbb{Z}_{2^m}$. We start with m qubits in state

$$|0\rangle$$

and then apply the Hadamard transform H_m to get

$$\frac{1}{\sqrt{2^m}} \sum_{i=0}^{2^m-1} |i\rangle \,.$$

We then measure the system. Each integer in the range $0 \ldots 2^m - 1$ is measured with equal probability. Specifically, each is measured with probability $1/\sqrt{2^m}$.

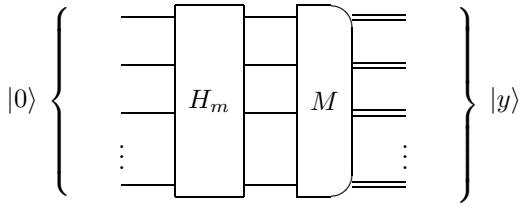

Figure 8.1: Circuit that generates random numbers.

Chapter 9

Factoring

In this chapter we show that quantum circuits offer us a way to factor large integers, a problem that was previously thought to be more or less unfeasible.

9.1 One Way Functions

Although not proved, multiplication is believed to be a *one way function*. Roughly speaking, a one way function is one that *easy* to compute but *hard* to compute the inverse.

By *easy* we generally mean that there is an algorithm that perform a number of operations that is equal to (or less than) some polynomial function of the input size. If n is the input size, then we say the number of operations is equal to $O(n^k)$ for some fixed k. We also will call such functions *efficient*. By *hard* we can mean that an *efficient* (classical) algorithm doesn't exist. That *any* one way functions actually exist is an open problem.

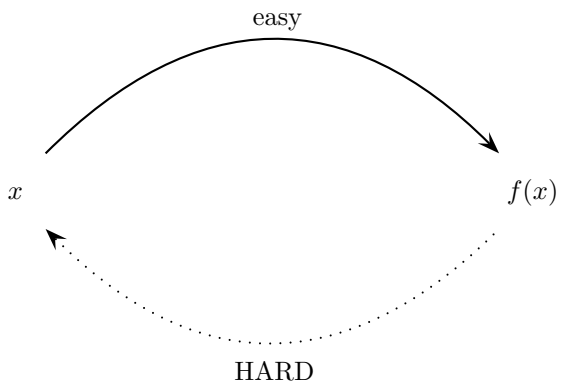

Figure 9.1: A one-way function

It is generally believed that multiplication is a one way function, i.e. there does not exist an efficient (classical) algorithm that factors integers. The fastest algorithms known scale in complexity (i.e. the number of operations required) with respect to some exponential function of the input size. Current popular

encryption algorithms such as RSA are seen as secure as it is assumed that multiplication is a one way function. Roughly speaking authorized viewers data can decrypt data because they know the correct factors of some number to multiply together to obtain the decryption key. Unauthorized eavesdroppers do not know all the factors, and so would have to calculate them themselves. For large numbers this would take a very long time on current computers. For example current classical computers would take several million years to factor a 256 digit number.

Shor's algorithm [21] gives us a way to factor these integers using quantum circuits. This algorithm is not deterministic, and will not output a factor every time, but it will output a factor with high enough probability so that if it is run enough times, it would find a factor on average with fewer operations than a current classical computer.

Remark 9.1.1. Shor's algorithm has been implemented in an NMR quantum computer, but the most only to factor 15 into 3 and 5 using a 7 qubit [23].

Remark. This chapter contains several purely number theoretical results. The proofs to these are often omitted. Unless otherwise stated, the proofs can be found in [11].

9.2 Factors from Order

In this section will show that factoring a number n can be reduced to finding the order of an element in $a \in \mathbb{Z}_n^*$. Recall that $\mathbb{Z}_n^*$ is the group of integers modulo n with the group operation multiplication (and so must have a multiplicative inverse).

Lemma 9.2.1. *$\mathbb{Z}_n^*$ contains exactly all those a in $a \in \mathbb{Z}_n$ such that* $\gcd(a, n) = 1$.

For the sake of brevity throughout this chapter a will be a member of $\mathbb{Z}_n^*$ and r will be the order of a in $\mathbb{Z}_n^*$. Thus r is the smallest integer such that

$$a^r \equiv 1 \mod n. \tag{9.1}$$

Note for any finite group G, and for any $g \in G$ we have $g^{|G|} = 1_G$. Thus $r \leq n$. Also note that if n is a power of a single prime, there do exist efficient classical algorithms that can determine whether this is the case and to find the prime Thus we will focus on the case when n has at least two distinct prime factors.

Theorem 9.2.1. *Let n be odd with at least two distinct prime factors. Assume*

i. $a \neq 1$

ii. r *is even*

iii. $a^{\frac{r}{2}} \not\equiv -1 \mod n$.

Then both $\gcd(a^{\frac{r}{2}} + 1, n)$ *and* $\gcd(a^{\frac{r}{2}} - 1, n)$ *are non-trivial factors of n.*

Proof. We assume r is even and $a \neq 1$. Thus we can factor $a^r - 1$ into the product

$$a^r - 1 = (a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1). \tag{9.2}$$

By definition of order $a^r \equiv 1 \mod n$, which implies $a^r - 1 \equiv 0 \mod n$. This in turn implies that n must divide $a^r - 1$. Thus by (9.2) n must share some factor ($\neq 1$) with either $(a^{\frac{r}{2}} - 1)$ or with $(a^{\frac{r}{2}} + 1)$, or with both.

Assume that n divides $(a^{\frac{r}{2}} - 1)$. Then $a^{\frac{r}{2}} \equiv 1 \mod n$, which contradicts the fact that r is the smallest integer such that $a^r \equiv 1 \mod n$. Assume that n divides $(a^{\frac{r}{2}} + 1)$. Then $a^{\frac{r}{2}} \equiv -1 \mod n$, which is false by assumption.

We have found the n shares some factor with $(a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1)$, but divides neither of $(a^{\frac{r}{2}} - 1)$ or $(a^{\frac{r}{2}} + 1)$. Therefore $\gcd(a^{\frac{r}{2}} + 1, n)$ and $\gcd(a^{\frac{r}{2}} - 1, n)$ must be non-trivial factors of n. □

Say we are given a such that the assumptions in the above theorem are satisfied. Say also that for this a we can find it's order r in $\mathbb{Z}_n^*$ (this is not an easy task, but for the time being assume that we can). Then the above theorem says we can use Euclid's algorithm to calculate $\gcd(a^{\frac{r}{2}}+1, n)$ and $\gcd(a^{\frac{r}{2}}-1, n)$, and thus find non trivial factors of n. Euclid's algorithm is known to be efficient.

Example 9.2.1. Let $n = 15$, $d_+ = \gcd(a^{\frac{r}{2}} + 1, 15)$ and $d_- = \gcd(a^{\frac{r}{2}} - 1, 15)$.

$\mathbb{Z}_{15}^* = \{1, 2, 4, 7, 8, 11, 13, 14\}$

a	r	r even?	$a \neq 1$	$a^{\frac{r}{2}} \not\equiv -1$	d_+	d_-
1	0	✓	×	✓	1	15
2	4	✓	✓	✓	5	3
4	2	✓	✓	✓	5	3
7	4	✓	✓	✓	5	3
8	4	✓	✓	✓	5	3
11	2	✓	✓	✓	3	5
13	4	✓	✓	✓	5	3
14	2	✓	✓	×	15	1

We see that for each case where $a \in \mathbb{Z}_{15}^*$, $a \neq 1$, r is even and $a^{\frac{r}{2}} \not\equiv -1 \mod 15$ we have that $\gcd(a^{\frac{r}{2}} + 1, 15)$ and $\gcd(a^{\frac{r}{2}} - 1, 15)$ are non-trivial factors of 15. We also see that for the only $a \in \mathbb{Z}_{15}^*$ such that $a^{\frac{r}{2}} \equiv -1 \mod 15$, i.e. $a = 14$, $\gcd(a^{\frac{r}{2}} + 1, 15)$ and $\gcd(a^{\frac{r}{2}} - 1, 15)$ are trivial factors of 15.

Lemma 9.2.2. *Let n be odd with at least two distinct prime factors. Choose $a \in \mathbb{Z}_n^* \backslash \{1\}$ randomly with uniform distribution. The probability that the order r of a in $\mathbb{Z}_n^*$ is even and $a^{\frac{r}{2}} \not\equiv -1 \mod n$ is at least $\frac{1}{2}$.*

Example 9.2.2. Let us choose a from $\mathbb{Z}_{15}^* \backslash \{1\}$ at random using a uniform distribution (cf. Example 9.2.1). Then $\gcd(a^{\frac{r}{2}} + 1, 15)$ and $\gcd(a^{\frac{r}{2}} - 1, 15)$ would be non-trivial factors with a probability of $\frac{6}{7} > \frac{1}{2}$.

Therefore if we can find an algorithm to find the order of an element chosen from $\mathbb{Z}_n^* \backslash \{1\}$ with uniform distribution, then we can find a non trivial factor of n with a probability of at least $\frac{1}{2}$.

9.3 Shor's Algorithm

For a given $a \in \mathbb{Z}_n^*$, its order r is the period of the function $f : \mathbb{Z} \to \mathbb{Z}_n$

$$f(k) = a^k \mod n.$$

We will describe in this section an algorithm that uses the inverse QFT in $\mathbb{Z}_n$ to find the period of this function (with some probability), and then uses the method from the previous section to find non trivial factors . We will give the algorithm first, which if unfamiliar may seem to come out of the blue somewhat. We will then discuss the probability of it succeeding. Firstly we present a small discussion about continued fractions.

Consider $\alpha \in \mathbb{Q}_{>0}$. It should be clear that α has a finite continued fraction expansion, i.e. there are $\alpha_0, \alpha_1, \ldots, \alpha_n \in \mathbb{N}$ such that

$$\alpha = \alpha_0 + \cfrac{1}{\alpha_1 + \cfrac{1}{\alpha_2 + \cfrac{1}{\ddots + \cfrac{1}{\alpha_n}}}}.$$

In such a case we write $\alpha = [\alpha_0, \alpha_1, \ldots, \alpha_n]$. The ith convergent of the continued fraction expansion for α is defined to be

$$\frac{p_i}{q_i} = [\alpha_0, \alpha_1, \ldots, \alpha_i].$$

Note that all the convergents can be found efficiently using Euclid's algorithm (i.e. all the p_i and q_i). We will use the fact that the q_i can be found efficiently in the following algorithm, due to Shor [21].

The algorithm to find the order non trivial factors of n is as follows.

1. Pick $a \in \mathbb{Z}_n \backslash \{1\}$ randomly using a uniform distribution. A quantum circuit as discussed in Chapter 8 could be used.

2. Calculate $\gcd(a, n)$ using Euclid's algorithm. If $\gcd(a, n) > 0$, this is a non-trivial factor of n and stop. Otherwise we know that $a \in \mathbb{Z}_n^* \backslash \{1\}$.

3. Check if $r < 19$, simply by multiplying a together 19 times. If $r < 19$ then skip to Step 11. This rather bizarre step is related to the probabilities of the algorithm succeeding and is explained more later.

4. Pick m such that $m = 2^i$ for some integer i and $n^2 \leq m^2 < 2n^2$.

5. Prepare two quantum registers in compound state $|0\rangle |0\rangle$. Let one be large enough to represent $\mathbb{Z}_m$ Thus it must be of length (at least) i. Let the other be large enough to represent $\mathbb{Z}_n$ - thus it must be of length at least j where $2^j \geq n$. The register that represents $\mathbb{Z}_m$ is written on the left, and the register that represents $\mathbb{Z}_n$ is written on the right. Thus for $x \in \mathbb{Z}_m$ and $y \in \mathbb{Z}_n$ a typical pure state is written as $|x\rangle |y\rangle$.

6. Apply the Hadamard transform H_m. The resulting state is

$$\frac{1}{\sqrt{m}} \sum_{k=0}^{m-1} |k\rangle |0\rangle .$$

7. Apply a map defined by $|k\rangle |0\rangle \mapsto |k\rangle |a^k \mod n\rangle$. This results in state

$$\frac{1}{\sqrt{m}} \sum_{k=0}^{m-1} |k\rangle |a^k\rangle .$$

Note that the function $k \mapsto a^k \mod n$ had period r (which is unknown). Thus we can write the above superposition as

$$\frac{1}{\sqrt{m}} \sum_{l=0}^{r-1} \sum_{q=0}^{s_l} |qr + l\rangle |a^l\rangle ,$$

where s_l is the largest integer such that $s_l r + l < m$. We note that each $a_l < r$. Thus roughly speaking in the above superposition, in the register on the right, only the members of $\mathbb{Z}_n$ that are less than the order r are present.

We also note that the left multiplier of each $|a^l\rangle$ is

$$\sum_{q=0}^{s_l} |qr + l\rangle .$$

We can see that this superposition also contains information about r. It is a superposition of basis states such that an $|x\rangle$ is in the superposition if and only if $x = qr + l$ for some q. This is essentially a sequence with period r. We use the inverse QFT to 'extract' this period.

8. Apply the inverse QFT in $\mathbb{Z}_m$ to the register on the left. Thus the state is

$$\frac{1}{\sqrt{m}} \sum_{l=0}^{r-1} \sum_{q=0}^{s_l} \frac{1}{\sqrt{m}} \sum_{p=0}^{m-1} e^{\frac{2\pi i p(qr+l)}{m}} |p\rangle |a^l\rangle . \tag{9.3}$$

Don't worry if this looks a bit unwieldy - the proof that shows this state when measured returns a 'good' result with high enough probability is omitted. We won't directly use this state again.

9. Observe the register on the left. This results is some $p \in \mathbb{Z}_m$.

10. Apply Euclid's algorithm to find the convergents $\frac{p_i}{q_i}$ of $\frac{p}{m}$, and (try to) find the smallest q_i such that $a^{q_i} = 1 \mod n$. If such a q_i is found then q_i is the order of a.

11. Use Euclid's algorithm to calculate $\gcd(a^{\frac{r}{2}} + 1, n)$ and $\gcd(a^{\frac{r}{2}} - 1, n)$. As we will show, the probability that these are non trivial factors of n is at least $1/20 \log\log n$.

Remark 9.3.1. As already mentioned in Remark 9.1.1, Shor's algorithm has been implemented to factor 15 using a 7 qubit quantum computer. Using the notation from the algorithm above, $n = 15$. We need to have a register capable of representing $\mathbb{Z}_n$, so we must have a register of length $\lceil \log_2(15) \rceil = 4$, i.e. 4 qubits. We also need to have a register capable of representing $\mathbb{Z}_m$, where m is chosen such that $n^2 <= m < 2n^2$, and so the smallest such a register could be is $\lceil \log_2(15^2) \rceil = 8$ qubits. Thus we need at least 12 qubits. The method used in [23] was some 'cut down' version of the algorithm.

We will now discuss the probability of this algorithm succeeding. The final step has already been discussed in the previous section. If we have managed to find the order of the uniformly chosen element a, then the probability that the final step outputs non trivial factor(s) of n is at least $\frac{1}{2}$.

However the previous steps do not guarantee that we can find this order. We will now discuss how likely it is that we do. We need the following theorem, which we will not prove.

Theorem 9.3.1. *If for $a \in \mathbb{Q}_{>0}$ and $p, q \in \mathbb{N}$, $\gcd p, q = 1$ and*

$$0 < \left| a - \frac{p}{q} \right| \leq \frac{1}{2q^2}$$

then $\frac{p}{q}$ is a convergent of the continued fraction expansion of a.

This theorem is easily applied to our situation.

Lemma 9.3.1. *If for $p, r, d, m, n \in \mathbb{N}$ we have $\gcd(d, r) = 1$, $n^2 \leq m < 2n^2$, $r < n$ and*

$$\left| p - d\frac{m}{r} \right| \leq \frac{1}{2}, \tag{9.4}$$

then $\frac{d}{r}$ is a convergent of $\frac{p}{m}$.

Proof. Clearly we have

$$\begin{aligned} \left| \frac{p}{m} - \frac{d}{r} \right| =& \frac{1}{m} \left| p - d\frac{m}{r} \right| \leq \frac{1}{2m} \leq \frac{1}{2n^2} \\ <& \frac{1}{2r^2}. \end{aligned}$$

Therefore by Theorem 9.3.1 we have that $\frac{d}{r}$ is a convergent of $\frac{p}{m}$. □

The above lemma shows that if we have measured (9.3) to find a p, such that it is at most $\frac{1}{2}$ away from some integer multiple, d, of $\frac{m}{r}$, such that d satisfies $\gcd(d, r) = 1$, then step 10 will result in the correct order of a. A specific case is shown in Figure 9.2 that appears to show that it is likely that these requirements are satisfied. We present the following lemmata for more general cases.

Lemma 9.3.2. *For $n > 100$, observing (9.3) will result in p such that $\left| p - d\frac{m}{r} \right| \leq \frac{1}{2}$ for some integer d with a probability of at least $\frac{2}{5}$.*

There is a one to one correspondence between p that satisfy (9.4), and $d \in \{0, 1, \ldots, r-1\}$. Therefore observing p that satisfies equation (9.4) for some d essentially means that some $d \in \{0, 1, \ldots, r-1\}$ is chosen (although we may not know what it is). We would like to know, given a p that satisfies (9.4) for some d, what the probability of that for this d satisfies $\gcd(d, r) = 1$, which is the additional requirement to invoke Lemma 9.3.1.

Lemma 9.3.3. *For $r > 19$ and $r < n$ the probability that a d chosen as above satisfies $\gcd(d, r) = 1$ is as least $1/4 \log \log n$*

This previous lemma explains why the algorithm checks if the order of a is less than 19.

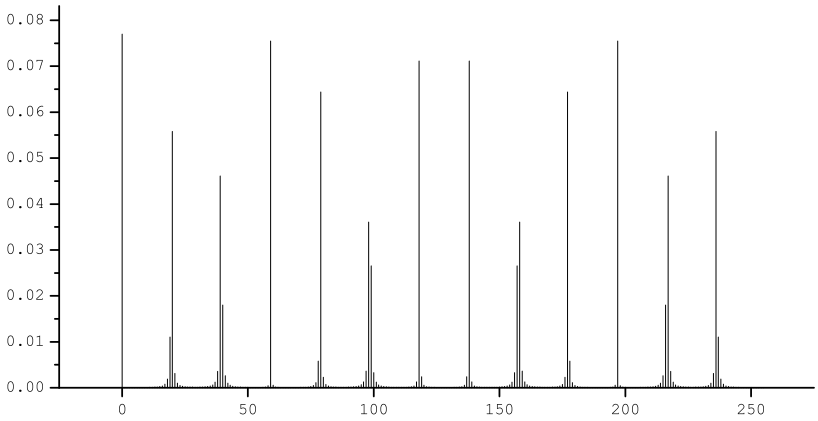

Figure 9.2: Example of the probability distribution of state (9.3) with respect to p for the case $r = 13$ and $m = 256$. We see that p is likely to be observed close to some integer multiple d of $\frac{m}{r}$. We also see that $d \in \{0, 1, \ldots, 12\}$. The only value of d such that $gcd(d, r) \neq 1$ is 0.

Theorem 9.3.2. *The overall probability that the factoring algorithm finds non trivial factors of* $n > 100$ *is at least*

$$\frac{1}{20 \log \log n}.$$

Proof. Notice that to find the factors, all of the following must be true of the uniformly chosen a:

1. The order r of a must be even and $a^{\frac{r}{2}} \not\equiv -1 \mod n$.
2. p must be observed such that $\left| p - d\frac{m}{r} \right| \leq \frac{1}{2}$ for some integer d.
3. For this integer $\gcd(d, r) = 1$.

The probabilities of each these are at least $\frac{1}{2}$, $\frac{2}{5}$ and $\frac{1}{4 \log \log n}$ respectively. □

It can be shown that the entire algorithm requires $O(l(n)^3)$ operations, where $l(n)$ is the number of digits required to represent n.

Because the probability of success of a single iteration of the algorithm is $1/20 \log \log(n)$, we can see that on average we would have to run it $20 \log \log n$ times to find a factor. In total this would take an average of $O(l(n)^3 \log \log(n))$ operations. This is much better than any classical algorithm that scales in complexity exponentially with respect to the input size $l(n)$.

Chapter 10

Searching

In this chapter we discuss a formalization of *searching*, show a few ways in which a quantum computer could (and couldn't) search unsorted lists faster than conventional computer.

10.1 Blackbox Functions

Let us define a *blackbox function* as a function

$$f : \mathbb{F}_2^m \longrightarrow \mathbb{F}_2$$

that returns its value instantly in one computational step, where the inner workings of f are unknown to us. We say our problem of searching some unsorted list is equivalent to trying to find some $\mathbf{x} \in \mathbb{F}_2^m$ such that $f(\mathbf{x}) = 1$. We call such $\mathbf{x}$ *solutions* to the searching problem.

We also define for some *unknown* $\mathbf{y} \in \mathbb{F}_2^m$, a blackbox function $f_\mathbf{y}$ such that

$$f_\mathbf{y}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} = \mathbf{y} \\ 0 & \text{if } \mathbf{x} \neq \mathbf{y}. \end{cases}$$

We say that searching in this case is trying to find $\mathbf{y}$, when all we can do is query f for some $\mathbf{x}$. In this case there is only one solution, $\mathbf{y}$.

Remark. Sometime a blackbox function is called an *oracle*, as it just 'magically' outputs the result.

Example. We can order the elements of $\mathbb{F}_2^m$. Notice that for $\mathbf{x} \in \mathbb{F}_2^m$ there are $\mathbf{x}_i \in \mathbb{F}_2$ such that $\mathbf{x} = (x_0, x_1, \ldots, x_{m-1})$. This is a binary representation of some number in Z_{2^m}. Thus we have a natural order on $\mathbb{F}_2^m$, and we have a *list*. We can say that each place on the list, i.e. each element of $\mathbb{F}_2^m$, actually represents an entry of a phone book. A phone book is unsorted with respect to the telephone numbers - if we are trying to find an entry with a given telephone number, we have no idea where in the phone book the listing may be. The only way is for us to check each line, see if the number given is equal to the one we want. Our current place in the phone book which we are checking is $\mathbf{x}$, the place in the phone book with the number we want is the unknown $\mathbf{y}$, the phone book itself is $f_\mathbf{y}$ and the checking of each line is calling $f_\mathbf{y}(\mathbf{x})$, which results in 1 if the line has the right number, and 0 if it hasn't.

Remark. One may think that the idea of a *blackbox function* is a bit artificial if we are trying to relate this to some real world scenario - real world operations are not instant. However it is a reasonable concept for working out probability of algorithms succeeding, and giving lower bounds on the number of operations used.

If $\mathbf{y}$ is unknown, picking $x_1, x_2, \ldots, x_k$ disjoint with uniform probability, and calling $f(\mathbf{x}_i)$ for each one would result in finding $\mathbf{y}$ with probability of $\frac{k}{2^m}$.

Example. Consider again the phone book example, with the phone book containing n entries. Picking k entries in the phone book at random with uniform distribution would find a solution with probability of $\frac{k}{n}$.

We can extend the idea of blackbox function to a *quantum* blackbox function. Given a traditional blackbox function f as above, define the quantum black box function, or *query operator*, Q_f to be the linear map such that

$$Q_f |\mathbf{x}\rangle |b\rangle = Q_f |\mathbf{x}\rangle |b \oplus f(\mathbf{x})\rangle ,$$

where $\mathbf{x} \in \mathbb{F}_2^m$, $b \in \mathbb{F}_2$, and $\oplus$ is addition modulo 2 (the group operation in $\mathbb{F}_2 = \mathbb{Z}_2$). Thus $|\mathbf{x}\rangle$ is the state of an m-length quantum register, known as the known as the *source* register, and $|b\rangle$ is the state of a single qubit, the *target* qubit. The map Q_f is a permutation on an $m+1$ dimensional Hilbert space, and so it is indeed unitary.

10.2 How Not To Search

We will describe in this section a perhaps naive way of using the query operator $Q_{f_\mathbf{y}}$ that does not do any better than just guessing $\mathbf{y}$. The steps are

1. Start with a source register of length m in state $|\mathbf{0}\rangle$, and a single qubit, the target qubit in state $|0\rangle$. Thus the compound state is $|\mathbf{0}\rangle |0\rangle$.

2. Apply the Hadamard transform H_m to the source register. This results in state

$$\frac{1}{\sqrt{2^m}} \sum_{\mathbf{x} \in \mathbb{F}_2^m} |\mathbf{x}\rangle |0\rangle .$$

3. Apply the query operator $Q_{f_\mathbf{y}}$ to result in state

$$\frac{1}{\sqrt{2^m}} \sum_{\mathbf{x} \in \mathbb{F}_2^m} |\mathbf{x}\rangle |f_\mathbf{y}(\mathbf{x})\rangle .$$

4. Measure the register of length m.

Since the state of the register before measurement was just an equal weighting of all possible states $|\mathbf{x}\rangle$, they probability that we find $\mathbf{y}$ using this method is the same as if we just picked $\mathbf{x}$ at random with uniform distribution.

10.3 Single Query Without (Grover's) Amplification

We improve the earlier method using Hadamard transforms. Firstly let us define the *modified query operator*. Notice that the query operator Q_f acting on state

$$|\mathbf{x}\rangle \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

results in state

$$(-1)^{f(\mathbf{x})} |\mathbf{x}\rangle \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle).$$

Notice that in this case the application of the query operator has flipped the amplitude of $|\mathbf{x}\rangle$ if and only if it is a solution to the searching problem, and has left the state of the target qubit as it was. We will no longer refer to the target qubit, and treat it just as an ancilla qubit. The modified query operator is thus defined to be the map

$$V_f |\mathbf{x}\rangle = (-1)^{f(\mathbf{x})} |\mathbf{x}\rangle .$$

We will also require another map. Let $f_{\mathbf{0}} : \mathbb{F}_2^m \to \mathbb{F}_2$ be defined to be

$$f_{\mathbf{0}}(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} = \mathbf{0} \\ 0 & \text{otherwise.} \end{cases}$$

By Chapter 4, this is can be constructed using a boolean circuit, and so it can be constructed using some quantum circuit, using some ancilla qubits. The circuit would perform the function

$$F_{\mathbf{0}} |\mathbf{x}\rangle |0\rangle = |\mathbf{x}\rangle |f_{\mathbf{0}}(\mathbf{x})\rangle ,$$

where we omit writing the ancilla qubits. Note $f_{\mathbf{0}}$ is *not* a blackbox function since $\mathbf{0}$ is known.

We will also use the fact that the fourier transform in $\mathbb{F}_2^m$ is self inverse. This means that the QFT in $\mathbb{F}_2^m$ is self inverse, i.e. the Hadamard transform H_m is self inverse. Thus we can apply the map H_m to either side of the following expression to result in the other side

$$|\mathbf{y}\rangle \xleftrightarrow{H_m} \frac{1}{\sqrt{2^m}} \sum_{\mathbf{x} \in \mathbb{F}_2^m} (-1)^{\mathbf{x}.\mathbf{y}} |\mathbf{x}\rangle , \tag{10.1}$$

or to take the specific case of $\mathbf{y} = \mathbf{0}$ we get

$$|\mathbf{0}\rangle \xleftrightarrow{H_m} \frac{1}{\sqrt{2^m}} \sum_{\mathbf{x} \in \mathbb{F}_2^m} |\mathbf{x}\rangle . \tag{10.2}$$

Using these we are ready to describe a better algorithm for searching.

1. Start, as before, with a source register of length m in state $|\mathbf{0}\rangle$.

2. Apply the Hadamard transform H_m to the source register. This results in state

$$\frac{1}{\sqrt{2^m}} \sum_{\mathbf{x} \in \mathbb{F}_2^m} |\mathbf{x}\rangle .$$

3. Apply the modified query operator $V_{f_\mathbf{y}}$ to result in state

$$\frac{1}{\sqrt{2^m}} \sum_{\mathbf{x} \in \mathbb{F}_2^m} (-1)^{f_\mathbf{y}(\mathbf{x})} |\mathbf{x}\rangle .$$

Remember that we are using an ancilla qubit that we are not writing. Notice also that this is an equal superposition of all states $|\mathbf{x}\rangle$ where $\mathbf{x} \in \mathbb{F}_2^m$, but with the sign of state $|\mathbf{y}\rangle$ inverted. If we were to measure the system at this point, we would only find the solution $\mathbf{y}$ with the same probability as if we chose $\mathbf{x}$ at random using a uniform distribution.

4. Apply the Hadamard transform H_m to result in the map

$$\begin{aligned}\frac{1}{\sqrt{2^m}} \sum_{\mathbf{x} \in \mathbb{F}_2^m} (-1)^{f_\mathbf{y}(\mathbf{x})} |\mathbf{x}\rangle =& \frac{1}{\sqrt{2^m}} \Big((\sum_{\mathbf{x} \in \mathbb{F}_2^m} |\mathbf{x}\rangle) - 2 |\mathbf{y}\rangle \Big) \\ \xmapsto{H_m} & |0\rangle - \frac{2}{\sqrt{2^m}} \sum_{\mathbf{x} \in \mathbb{F}_2^n} (-1)^{\mathbf{x}.\mathbf{y}} |\mathbf{x}\rangle \\ =& (1 - \frac{2}{\sqrt{2^m}}) |0\rangle - \frac{2}{\sqrt{2^m}} \sum_{\mathbf{x} \neq \mathbf{0}} (-1)^{\mathbf{x}.\mathbf{y}} |\mathbf{x}\rangle\end{aligned}$$

We use (10.1) and (10.2) to calculate the above.

5. We then apply $F_\mathbf{0}$ to the compound system of the register and an extra qubit (in state $|0\rangle$) to get

$$\begin{aligned}& (1 - \frac{2}{\sqrt{2^m}}) |\mathbf{0}\rangle |0\rangle - \frac{2}{\sqrt{2^m}} \sum_{\mathbf{x} \neq \mathbf{0}} (-1)^{\mathbf{x}.\mathbf{y}} |\mathbf{x}\rangle |0\rangle \\ \xmapsto{F_\mathbf{0}} & (1 - \frac{2}{\sqrt{2^m}}) |\mathbf{0}\rangle |1\rangle - \frac{2}{\sqrt{2^m}} \sum_{\mathbf{x} \neq \mathbf{0}} (-1)^{\mathbf{x}.\mathbf{y}} |\mathbf{x}\rangle |0\rangle\end{aligned}$$

Note that the resulting state is entangled.

6. Measure the extra qubit (the one on the right). From now on we ignore this qubit (the state was entangled so the act of measurement has affected the source register).

State after measurement	Probability
$\|\mathbf{0}\rangle$	$(1 - \frac{2}{2^m})^2$
$\frac{1}{\sqrt{2^m - 1}} \sum_{\mathbf{x} \neq \mathbf{0}} (-1)^{\mathbf{x}.\mathbf{y}} \|\mathbf{x}\rangle$	$1 - (1 - \frac{2}{2^m})^2$

7. Apply the Hadamard transform H_m to the source register. Note that

$$\sum_{\mathbf{x} \neq \mathbf{0}} (-1)^{\mathbf{x}.\mathbf{y}} |\mathbf{x}\rangle = \Big(\sum_{\mathbf{x} \in \mathbb{F}_2^n} (-1)^{\mathbf{x}.\mathbf{y}} |\mathbf{x}\rangle \Big) - |\mathbf{0}\rangle .$$

State after Hadamard transform H_m	Probability
$\frac{1}{\sqrt{2^m}} \sum_{\mathbf{x} \in \mathbb{F}_2^m} \lvert \mathbf{x} \rangle$	$(1 - \frac{2}{2^m})^2$
$\frac{1}{\sqrt{(2^m-1)(2^m)}} \left((2^m - 1) \lvert \mathbf{y} \rangle - \sum_{\mathbf{x} \neq \mathbf{y}} \lvert \mathbf{x} \rangle \right)$	$1 - (1 - \frac{2}{2^m})^2$

8. Measure the source register. If the state was that of the top row in the above table, the probability that this measurement results in $\mathbf{y}$ is $\frac{1}{2^m}$. If the state was that of the bottom row in the above table, the probability of that the measurement results in $\mathbf{y}$ is $\frac{2^m-1}{2^m}$. Thus taking the probabilities of being in those states to start with into account, the overall probability the this final measurement results in $\mathbf{y}$ is

$$\begin{aligned} & \frac{1}{2^m}(1 - \frac{2}{2^m})^2 + (\frac{2^m - 1}{2^m})(1 - (1 - \frac{2}{2^m})^2) \\ =& \frac{5}{2^m} + \frac{12}{2^{2m}} - \frac{8}{2^{3m}} \\ \approx& \frac{5}{2^m} \text{ for large } m. \end{aligned}$$

This method makes a single query to the modified query operator. It returns $\mathbf{y}$ with a probability of about $\frac{5}{2^m}$ for large m. This is about 5 times better than making a single query by picking a single $\mathbf{x}$ randomly using a uniform distribution.

10.4 Amplitude Amplification

In this section we describe a method of using unitary operators that, given a superposition of uniform amplitudes

$$\frac{1}{\sqrt{2}} \sum_{\mathbf{x} \in \mathbb{F}_2^n} \lvert \mathbf{x} \rangle ,$$

will amplify the amplitude of those $\lvert \mathbf{x} \rangle$ such that $f(\mathbf{x}) = 1$. Recall that a quantum state must be of unit length, so this amplification is only relative. Once the operator has amplified these amplitudes, observation is more likely to lead to a solution.

As earlier we will make use of the modified query operator V_f. We will also need another map R_m such that

$$R_m \lvert \mathbf{x} \rangle = \begin{cases} - \lvert \mathbf{x} \rangle & \mathbf{x} = 0 \\ \lvert \mathbf{x} \rangle & \text{otherwise.} \end{cases}$$

If we index $2^m \times 2^m$ matrices by $\mathbb{F}_2^m$ (taking the natural order defined by taking the elements of $\mathbb{F}_2^m$ as binary representations of integers), we get that

$$R_m = \begin{pmatrix} -1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix} .$$

We will omit how to decompose this map into a finite number of quantum gates for the sake of brevity.

Definition 10.4.1. Using the notation above, define G_m to be the unitary map such that

$$G_m = -H_m R_m H_m V_f.$$

We say the map $-H_m R_m H_m$ is the *inversion about average* operator, for reasons that will become clear.

To get an idea of what $H_m R_m H_m$ does, we try to find what its matrix looks like. We order the basis elements of $\mathbb{F}_2^m$ by taking the elements of $\mathbb{F}_2^m$ to be a binary representation of some integer. Firstly recall that

$$H_m = \sum_{\mathbf{x}\in\mathbb{F}_2^m} (-1)^{\mathbf{x}.\mathbf{y}} |\mathbf{x}\rangle .$$

Therefore $(H_m)_{\mathbf{xy}} = \frac{1}{\sqrt{2^m}}(-1)^{\mathbf{x}.\mathbf{y}}$. Thus (we omit the working again for brevity)

$$\begin{aligned}(H_m R_m H_m)_{\mathbf{xy}} &= \dots \\ &= \begin{cases} -\frac{2}{2^m} & \mathbf{x} \neq \mathbf{y} \\ 1 - \frac{2}{2^m} & \mathbf{x} = \mathbf{y} \end{cases}.\end{aligned}$$

Therefore

$$\begin{aligned}H_m R_m H_m &= \begin{pmatrix} 1-\frac{2}{2^m} & -\frac{2}{2^m} & -\frac{2}{2^m} & \dots & -\frac{2}{2^m} \\ -\frac{2}{2^m} & 1-\frac{2}{2^m} & -\frac{2}{2^m} & \dots & -\frac{2}{2^m} \\ -\frac{2}{2^m} & -\frac{2}{2^m} & 1-\frac{2}{2^m} & \dots & -\frac{2}{2^m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{2}{2^m} & -\frac{2}{2^m} & -\frac{2}{2^m} & \dots & 1-\frac{2}{2^m} \end{pmatrix} \\ &= I - 2P,\end{aligned}$$

where P is the projection onto the subspace spanned by

$$|\psi\rangle = \frac{1}{\sqrt{2}} \sum_{\mathbf{x}\in\mathbb{F}_2^n} |\mathbf{x}\rangle .$$

We would like to know the effect $-H_m R_m H_m$ has on a general state

$$\sum_{\mathbf{x}\in\mathbb{F}_2^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle .$$

We define the average of the amplitudes as

$$A = \frac{1}{\sqrt{2}} \sum_{\mathbf{x}\in\mathbb{F}_2^n} \alpha_{\mathbf{x}},$$

and so we can see that

$$\sum_{\mathbf{x}\in\mathbb{F}_2^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle = A \sum_{\mathbf{x}\in\mathbb{F}_2^n} |\mathbf{x}\rangle + \sum_{\mathbf{x}\in\mathbb{F}_2^n} (\alpha_{\mathbf{x}} - A) |\mathbf{x}\rangle .$$

The first term in the equation above is clearly part of the subspace spanned by $|\psi\rangle$. The second term is part of the orthogonal complement of that subspace,

as can be easily checked (by checking that the inner product of the two terms is equal to 0). Thus

$$-H_m R_m H_m \sum_{\mathbf{x}\in\mathbb{F}_2^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle = (2P - I) \sum_{\mathbf{x}\in\mathbb{F}_2^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$$
$$= \sum_{\mathbf{x}\in\mathbb{F}_2^n} (2A - \alpha_{\mathbf{x}}) |\mathbf{x}\rangle .$$

If some given $|\mathbf{x}\rangle$ has amplitude $\alpha_{\mathbf{x}} \approx A$, then the map $-H_m R_m H_m$ would not change the amplitude of $|\mathbf{x}\rangle$ by very much. Conversely if some $|\mathbf{x}'\rangle$ has amplitude of $\alpha_{\mathbf{x}'} \approx -A$, then the map $-H_m R_m H_m$ would approximately invert this amplitude and multiply it by 3. This is why $-H_m R_m H_m$ is called inversion about average.

10.4.1 Single Query, Single Solution

We can now describe Grover's search algorithm [10] for the case when there is a single solution $\mathbf{y}$.

1. Start with a quantum register of length m in state $|\mathbf{0}\rangle$.
2. Apply the Hadamard transform H_m to get
$$\frac{1}{\sqrt{2^m}} \sum_{\mathbf{x}\in\mathbb{F}_2^n} |\mathbf{x}\rangle .$$
3. Apply the modified query operator $V_{f_{\mathbf{y}}}$. This results in state
$$\frac{1}{\sqrt{2^m}} \sum_{\mathbf{x}\in\mathbb{F}_2^m} (-1)^{f_{\mathbf{y}}(\mathbf{x})} |\mathbf{x}\rangle .$$
Note that $V_{f_{\mathbf{y}}}$ has flipped the amplitude of $|\mathbf{y}\rangle$ (although $\mathbf{y}$ is still unknown).
4. Apply the map $-H_m R_m H_m$. The average of the amplitudes (before applying the map) was
$$A = \frac{1}{2^m}\Big((2^m - 1)\frac{1}{\sqrt{2^m}} + (1)(\frac{-1}{\sqrt{2^m}}\Big) = \frac{1}{\sqrt{2^m}}\Big(1 - \frac{2}{2^m}\Big).$$
Thus $-H_m R_m H_m$ performs the map
$$\frac{1}{\sqrt{2^m}} \sum_{\mathbf{x}\in\mathbb{F}_2^m} (-1)^{f_{\mathbf{y}}(\mathbf{x})} |\mathbf{x}\rangle = \frac{1}{\sqrt{2^m}} \sum_{\mathbf{x}\neq\mathbf{y}} |\mathbf{x}\rangle + (-\frac{1}{\sqrt{2^m}}) |\mathbf{y}\rangle$$
$$\xrightarrow{-H_m R_m H_m} (2A - \frac{1}{\sqrt{2^m}}) \sum_{\mathbf{x}\neq\mathbf{y}} |\mathbf{x}\rangle + (2A + \frac{1}{\sqrt{2^m}}) |\mathbf{y}\rangle$$
$$= \frac{1}{\sqrt{2^m}}\Big(1 - \frac{4}{2^m}\Big) \sum_{\mathbf{x}\neq\mathbf{y}} |\mathbf{x}\rangle + \frac{1}{\sqrt{2^m}}\Big(3 - \frac{4}{2^m}\Big) |\mathbf{y}\rangle .$$

5. Measure the system. This results in $\mathbf{y}$ with probability $\frac{1}{2^m}(3-\frac{4}{2^m})^2 \approx \frac{9}{2^m}$ for large m. This is approximately 9 times better than choosing any $\mathbf{x}$ at random using a uniform distribution.

Remark 10.4.1. Grover's search algorithm has been carried out on a two qubit NMR computer [12].

10.4.2 Known Number of Solutions

Let us now assume that a blackbox function $f : \mathbb{F}_m^2 \to \mathbb{F}^2$ has k solutions. Let T be the set of solutions and F be the rest of $\mathbb{F}_m^2$. Thus $|T| = k$. From the previous section we can see that given a uniform superposition of all the representatives of $\mathbb{F}_m^2$,

$$\frac{1}{\sqrt{2^m}} \sum_{\mathbf{x}\in\mathbb{F}_2^m} |\mathbf{x}\rangle \,, \tag{10.3}$$

the operator G_m keeps the amplitudes of $|\mathbf{x}\rangle$ such that $\mathbf{x} \in T$ all the same, and it keeps all the amplitudes of $|\mathbf{x}\rangle$ such that $\mathbf{x} \in F$ the same. Thus after r applications of G_m to (10.3) we know that the state of the system is

$$t_r \sum_{\mathbf{x}\in T} |\mathbf{x}\rangle + f_r \sum_{\mathbf{x}\in F} |\mathbf{x}\rangle \,, \tag{10.4}$$

for some $t_r, f_r \in \mathbb{C}$ dependant on r. Our aim is to try to find r such that t_r is maximised. To do this we can, and will, find t_r and f_r explicitly in terms of k and m. To achieve that, we will create a recurrence relation, i.e. find t_{r+1} and f_{r+1} in terms of t_r and f_r, and then solve it using the initial conditions $t_0 = f_0 = \frac{1}{\sqrt{2^m}}$.

Lemma 10.4.1. *Let all notation be as above.*

$$\begin{pmatrix} t_{r+1} \\ f_{r+1} \end{pmatrix} = \begin{pmatrix} 1-\frac{2k}{2^m} & 2-\frac{2k}{2^m} \\ -\frac{2k}{2^m} & 1-\frac{2k}{2^m} \end{pmatrix} \begin{pmatrix} t_r \\ f_r \end{pmatrix} \tag{10.5}$$

Proof. Apply the modified query operator V_f to (10.4) to result in

$$-t_r \sum_{\mathbf{x}\in T} |\mathbf{x}\rangle + f_r \sum_{\mathbf{x}\in F} |\mathbf{x}\rangle \,.$$

The average of the amplitudes in the above expression is

$$A = \frac{1}{2^m}\Big(-t_r k + f_r(2^m - k)\Big).$$

Application of the operator $-H_m R_m H_m$ results in

$$(2A + t_r) \sum_{\mathbf{x}\in T} |\mathbf{x}\rangle + (2A - f_r) \sum_{\mathbf{x}\in F} |\mathbf{x}\rangle \,.$$

We have now applied G_m $r+1$ times, so in fact $t_{r+1} = 2A + t_r$, and $f_{r+1} = 2A - t_r$. Expanding out these expressions gives us the result. □

Lemma 10.4.2. *Let all notation be as above. Then the solution to (10.5) with the initial conditions* $t_0 = f_0 = \frac{1}{\sqrt{2^m}}$ *is*

$$t_r = \frac{1}{\sqrt{k}} \sin\Big((2r+1)\theta_0\Big)$$
$$f_r = \frac{1}{\sqrt{2^m - k}} \cos\Big((2r+1)\theta_0\Big),$$

where $\theta_0 \in [0, \frac{\pi}{2}]$ *is chosen such that* $\sin^2\theta_0 = \frac{k}{2^m}$.

Proof. This is an exercise in recurrence relations and so will be omitted. See [11] for a proof. □

Lemma 10.4.3. *The probability of seeing a solution after* r *applications of* G_m *is* $\sin^2((2r+1)\theta_0)$.

Proof. After r iterations the state of the system is

$$t_r \sum_{\mathbf{x}\in T} |\mathbf{x}\rangle + f_r \sum_{\mathbf{x}\in F} |\mathbf{x}\rangle \,. \tag{10.6}$$

Thus the probability of observing any solution, i.e. any $\mathbf{x} \in T$, is equal to $|T|t_r^2 = kt_r^2$. By Lemma 10.4.2 we have that this must be equal to $\sin^2((2r+1)\theta_0)$ □

This leads to a surprising, albeit not always that useful, result.

Lemma 10.4.4. *Let* $k = \frac{1}{4}.2^m$. *Applying* G_m once *and then measuring the system would result in a solution with* certainty.

Proof. If $k = \frac{1}{4}.2^m$ then $\sin^2\theta_0 = \frac{1}{4}$ which implies $\theta_0 = \frac{\pi}{6}$. Then by 10.4.3 the probability of finding a solution after one application of G_m is $\sin^2(\frac{\pi}{2}) = 1$. □

However there are clearly cases when $k \neq \frac{1}{4}.2^m$.

Theorem 10.4.1. *Let* $f : \mathbb{F}_m^2 \to \mathbb{F}^2$ *be a blackbox function with* k *unknown solutions in* $\mathbb{F}_m^2$. *Assume that* $0 < k \leq \frac{3}{4}.2^m$. *Let* $\theta_0 \in [0, \frac{\pi}{3})$ *be such that* $\sin^2\theta_0 = \frac{k}{2^m}$. *Then applying* G_m *exactly* $\lfloor \frac{\pi}{4\theta_0} \rfloor$ *times to (10.3) and then measuring the system would result in a solution with a probability of at least* $\frac{1}{4}$.

Proof. By Lemma 10.4.3 the probability of seeing a solution after $\lfloor \frac{\pi}{4\theta_0} \rfloor$ iterations is $\sin^2((2\lfloor \frac{\pi}{4\theta_0} \rfloor + 1)\theta_0)$. Thus we try to give a lower bound for this probability. For some $|\delta| \leq \frac{1}{2}$

$$\left\lfloor \frac{\pi}{4\theta_0} \right\rfloor = -\frac{1}{2} + \frac{\pi}{4\theta_0} + \delta,$$

so

$$\left(2\left\lfloor \frac{\pi}{4\theta_0} \right\rfloor + 1\right)\theta_0 = \frac{\pi}{2} + 2\delta\theta_0.$$

Therefore

$$\left|\left(2\left\lfloor \frac{\pi}{4\theta_0} \right\rfloor + 1\right)\theta_0 - \frac{\pi}{2}\right| = |2\delta\theta_0| \leq \frac{\pi}{3}.$$

Since the (absolute value of the) derivative of $\sin^2$ is always less than or equal to 1, we must have that

$$\sin^2((2\left\lfloor\frac{\pi}{4\theta_0}\right\rfloor+1)\theta_0) \geq \sin^2(\frac{\pi}{2}-\frac{\pi}{3}) = \frac{1}{4}. \qquad \square$$

Now it should be more or less clear what the algorithm is. Remember k, the number of solutions, is known.

1. If $k > \frac{3}{4}.2^m$, pick $\mathbf{x}$ at random with uniform probability (perhaps using the random number generator from Chapter 8), and stop. In this case the probability of $\mathbf{x}$ being a solution is clearly at least $\frac{3}{4}$.
2. Otherwise start with a quantum register of length m in state $|\mathbf{0}\rangle$.
3. Apply the Hadamard transform H_m to get a uniform superposition of all the basis states
$$\frac{1}{\sqrt{2^m}}\sum_{\mathbf{x}\in\mathbb{F}_2^m}|\mathbf{x}\rangle\,.$$
4. Apply the operator G_m exactly $\lfloor\frac{\pi}{4\theta_0}\rfloor$ times.
5. Measure the system. By Theorem 10.4.1 the probability of finding a solution is at least $\frac{1}{4}$.

We can easily compare to the classical case when $k = 1$ and n is large. Roughly speaking, for small θ, $\sin\theta \approx \theta$. Thus for large n, $\frac{1}{2^m}$ is small, so $\theta_0 \approx \frac{1}{\sqrt{2^m}}$, which in turn implies $\lfloor\frac{\pi}{4\theta_0}\rfloor \approx \frac{\pi}{4}\sqrt{2^m}$. Thus the above algorithm performs $O(\sqrt{2^m})$ queries to find a solution with a probability of $\frac{1}{4}$. A search picking $\mathbf{x}$ at random with uniform probability would have to make $\frac{1}{4}2^m = O(2^m)$ queries to find a solution with probability of $\frac{1}{4}$.

10.4.3 Unknown Number of Solutions

Often we may not know how many solutions there are previous to a search, i.e. k is unknown. The algorithm in the previous section chose the number of the of times to apply G_m using k. The following theorem, which we will not prove, gives us a way to pick a number of times to apply G_m without previous knowledge of k.

Theorem 10.4.2. *Let $f : \mathbb{F}_m^2 \to \mathbb{F}^2$ be a blackbox function with k solutions in $\mathbb{F}_m^2$. Assume that $0 < k \leq \frac{3}{4}.2^m$. Let d be an integer such that $d \geq \sqrt{2^m}$. Let r be a uniformly chosen $r \in \{0, 1, \ldots, d-1\}$. Then applying G_m exactly r times to (10.3) would result in a solution with probability of at least $\frac{1}{4}$.*

Thus we present the following modified version of Grover's algorithm, which is in fact a simplified version of [6]. Remember that the number of solutions k is unknown.

1. Pick an $\mathbf{x} \in \mathbb{F}_2^m$ randomly with uniform distribution. If $f(\mathbf{x}) = 1$, then stop.

2. Let $d = \lfloor \sqrt{2^m} \rfloor + 1$.

3. Choose an integer $r \in \{0, 1, \ldots, d-1\}$ randomly with uniform distribution, perhaps using the quantum circuit describe in Chapter 8.

4. Apply the Hadamard transform H_m on a quantum register of length at m in state $|\mathbf{0}\rangle$ to get a uniform superposition of all the basis states

$$\frac{1}{\sqrt{2^m}} \sum_{\mathbf{x} \in \mathbb{F}_2^m} |\mathbf{x}\rangle .$$

5. Apply the operator G_m exactly r times.

6. Measure the system.

We know that there must be k solutions to the searching problem, it's just that we don't know the value of k. If $k > \frac{3}{4}.2^m$, then step 1 has output a solution with a probability of at least $\frac{3}{4}$. If $k \leq \frac{3}{4}.2^m$, then Theorem 10.4.2 tells us that the probability of getting a solution from step 6 is at least $\frac{1}{4}$. In either case, the probability of getting a solution is at least $\frac{1}{4}$, performing (at most) $O(\sqrt{2^m})$ queries. Thus finding a solution would, on average, require the algorithm to be run 4 times, which would still take $O(\sqrt{2^m})$ queries. A classical algorithm that would on average require $O(2^m)$ queries.

Chapter 11

Conclusion

We have merely skimmed the surface of what is *theoretically* possible using the effects of superposition of states as described in Chapter 2. By the results in Chapter 4 we have shown the unitary operations, which are the allowable operations on states, can be put together to make a circuit that will perform the same task as any classical boolean circuit. There would only be little point if a quantum computer could only perform those operations.

Note we say *little* point and not *no* point. We haven't yet mentioned it, but the speed of current classical processors, essentially very complicated boolean circuits, is limited largely by the fact that each operation produces heat, which is difficult to disperse. If current processors get too hot they essentially melt and become useless. As discussed in Chapter 3, operations in a quantum circuit are unitary maps, and are thus invertible. We will not get into the physics to explain why, but this means that quantum operations would produce no heat, and so the speed of a quantum circuit should not be limited by heat production.

Also we have shown that there is a way to send two classical bits using one quantum bit in Chapter 5. We have shown that it is possible to *teleport* the state of a quantum bit, but not to copy a general state in Chapter 6. Both of these protocols use entangled particles, as described in Chapter 2, these are particles that are somehow linked together, where the measurement of one affects the other.

Chapter 10 showed methods by which quantum circuits could be used to search large unsorted databases taking fewer queries than a classical computer, and said that a very simple case had been carried out (cf. Remark 10.4.1).

In Chapter 9 we have described Shor's algorithm, part of which can be run on a classical computer, and part of which must be run on a quantum computer, that factors integers in polynomial time. As explained in Remarks 9.1.1 and 9.3.1, this has only been carried out using a 7 qubit NMR quantum computer. However, it is difficult to scale up NMR computers to have enough qubits to factor large numbers. We quote Shor himself,

> "*If you're going to factor a 200-digit number, you're going to need thousands of qubits. The road from 7 to thousands is a long one.*"

Factoring large integers could theoretically crack current encryption schemes such as RSA. This leads to the rather obvious question of whether, if quantum computers will be able to factor large integers, is it possible to have some sort of

encryption schemes that are *not* crackable by quantum computers? The answer to this question is *yes*. There are encryption schemes, some of which are already available commercially, that use quantum properties to encrypt and send data. These use similar methods to the quantum teleportation protocol in Chapter 6. Schemes that detect if someone is eavesdropping on a transmission are also known.

Another field of quantum algorithms, that has not been mentioned, is the field of quantum error correction algorithms. These are schemes that correct some errors introduced during computation due to *decoherence* (cf. Remark 2.0.2). However, if the level of error is too high, which at the moment it is, these schemes offer little help.

In summary quantum computers have theoretically great potential, however decoherence is a problem that is still a long way from being solved.

Bibliography

[1] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin and H. Weinfurter: *Elementary gates for quantum computation*, Physical Review A 52:5, 3457-3467 (1995).

[2] J.S. Bell: *On the Einstein Podolsky Rosen paradox*, Physics 1:3, 195-200 (1964).

[3] Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K. Wootters: *Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels*, Physical Review Letters 70, 1895-1899 (1993).

[4] Charles H. Bennet and Stephen J. Wiesner: *Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states*, Physical Review Letters 69:20, 2881-2884 (1992).

[5] G.P. Berman, G.D. Doolen, R. Mainieri and V. I. Tsifrinovich: *Introduction to quantum computers*, Word Scientific, (1998).

[6] Michel Boyer, Gilles Brassard, Peter Hoeyer, Alain Tapp: *Tight bounds on quantum searching*, Fourth Workshop on Physics and Computation, 36-43 (1996).

[7] W. Clearwater: *Ultimate zero and one*, Copernicus (2000).

[8] A. Einstein, B. Podolsky and N. Rosen: *Can quantum-mechanical description of physical reality be considered complete?* Physical Review 41, 777-780 (1935).

[9] Oded Goldreich: *Foundations of cryptography: basic tools*, Cambridge University Press (2001).

[10] L. K. Grover: *A fast quantum mechanical algorithm for database search*, Proceedings of 28th Annual ACM Symposium on the Theory of Computing, 212-219 (1996).

[11] Mika Hirvensalo: *Quantum Computing: Second Edition*, Springer (2003).

[12] J. A. Jones, M. Mosca and R. H. Hansen: *Implementation of a quantum search algorithm on a nuclear magnetic resonance quantum computer*, Nature 393, 344-346 (1998).

[13] Laszlo B. Kish: *End of Moores law: thermal (noise) death of integration in micro and nano electronics*, Physics Letters A 305, 144149 (2002).

[14] C. Lavor, L. R. U. Manssur and R. Portugal: *Shor's algorithm for factoring large integers*, Laboratório Nacional de Computaçáo Científica (2004).

[15] I. Marcikic, H. de Riedmatten, W. Tittel, H. Zbinden, M. Legré, and N. Gisin: *Distribution of time-bin entangled qubits over 50 km of optical fiber*, Physical Review Letters 93:18, 1805021-1805024 (2004).

[16] M. Nielsen and I. Chuang: *Quantum computation and quantum information*, Cambridge University Press (2000).

[17] J. L. O'Brien, G. J. Pryde, A. G. White, T. C. Ralph amd D. Branning: *Demonstration of an all-optical quantum controlled-NOT gate* Nature 426, 264-267 (2003).

[18] Emil Post: *The two-valued iterative systems of mathematical logic*, Annals of Mathematics Studies, Princeton University Press (1941).

[19] K. J. Resch, M. Lindenthal, B. Blauensteiner, H. R. Bhm, A. Fedrizzi, C. Kurtsiefer, A. Poppe, T. Schmitt-Manderbach, M. Taraba, R. Ursin, P. Walther, H. Weier, H. Weinfurter, and A. Zeilinger: *Distributing entanglement and single photons through an intra-city, free-space quantum channel*, Optics Express 13, 202-209 (2005).

[20] Yaoyun Shi: *Both Toffoli and controlled-NOT need little help to do universal quantum computation*, Quantum Information and Computation, 3:1, 84-92 (2003).

[21] Peter W. Shor: *Algorithms for quantum computation: discrete logarithms and factoring*, Proceedings of the 35th annual symposium on foundations of computer science, 124134 (1994).

[22] Rupert Ursin, Thomas Jennewein, Markus Aspelmeyer, Rainer Kaltenbaek, Michael Lindenthal, Philip Walther and Anton Zeilinger: *Communications: Quantum teleportation across the Danube* Nature 430, 849-849 (2004).

[23] M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. Sherwood, and I. L. Chuang: *Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance*, Nature 414, 883-887 (2001).

[24] John Waltrous: *Introduction to quantum computation: lecture notes*, University of Calgary (2005).

[25] W. K. Wootters and W. H. Zurek: *A single quantum cannot be cloned* Nature 299, 802-803 (1982).

[26] T. Yamamoto, Yu. A. Pashkin, O. Astafiev, Y. Nakamura and J. S. Tsai: *Demonstration of conditional gate operation using superconducting charge qubits*, Nature 425, 941-944 (2003).