

Introduction to Quantum Computing

Lecture notes for the summer term 2025

Jannik Hellenkamp

Stefan Stump

Dominique Unruh

2025-06-12

Table of contents

Welcome	2
1 Introduction	2
1.1 Double-slit experiment	2
1.2 What is a quantum computer?	4
2 Probabilistic systems	6
2.1 Deterministic possibilities	6
2.2 Probability distribution	6
2.3 Probabilistic processes	7
Applying a probabilistic process	8
3 Quantum systems	9
3.1 Classical possibilities	9
3.2 Quantum states	9
3.3 Unitary transformation	10
4 Observing probabilistic and measuring quantum systems	11
4.1 Observing a probabilistic system	12
4.2 Measuring a quantum system	13
4.3 Elitzur–Vaidman bomb tester	15
5 Partial observing and measuring systems	15
5.1 Partially observing a probabilistic system	15
5.2 Partially measuring a quantum system	18
6 Composite Systems	19
6.1 Constructing composite systems	20
6.2 Measuring composite systems	22
7 Quantum Circuits	23
7.1 Visual language	24
7.2 Important gates	24
7.2.1 Single qubit gates	24
7.2.2 Controlled-NOT gate	25
7.3 Teleportation	26
8 Ket Notation	29
8.1 Teleportation	30
9 Bernstein-Vazirani Algorithm	33

10 Shor's Algorithm	36
10.1 Discrete Fourier Transformation	37
10.2 Reducing factoring to period finding	38
10.3 The quantum algorithm for period finding	39
10.4 Post processing	40
10.5 Constructing the DFT	42
11 Grover's algorithm	45
11.1 Preparations	45
11.1.1 Constructing the oracle V_f	46
11.1.2 Constructing FLIP _*	46
11.2 The algorithm for searching	48
11.2.1 Understanding the algorithm for searching	48

Welcome

These are the lecture notes for the “Introduction to Quantum Computing” lecture held by Dominique Unruh at RWTH Aachen in the summer term 2025. They should be viewed as an addition to the handwritten notes and the lecture recordings.

The lecture notes will be updated progressively throughout the semester, following the pace of the lectures. Last year's lecture notes can be found [here](#), but note that they are incomplete and may differ from this years content.

If you spot an error, please report it to [Gitlab](#). Alternatively, you can send Stefan Stump an e-mail (stefan.stump@rwth-aachen.de). If you have a question of understanding, please ask it in the Moodle forum.

These lecture notes are released under the CC BY-NC 4.0 license, which can be found [here](#).

The Jupyter notebooks created during the lectures can be found in the [JupyterHub](#) in the course “[IQC] Introduction to Quantum Computing” or in Moodle. These will be added shortly after the lectures. If changes are made to the files, they can be easily reset with the usual git commands using “Git” and then “Open Git Repository in Terminal”.

1 Introduction

1.1 Double-slit experiment

We start by looking at one of the most famous quantum experiments to get an idea of the surprising nature of quantum behaviour.

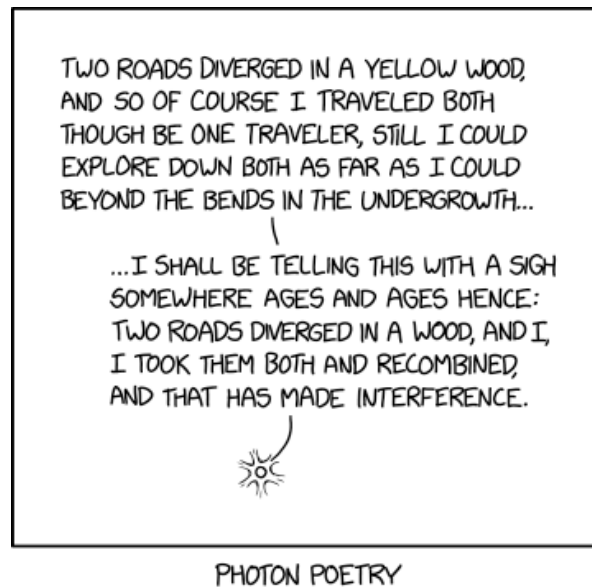


Figure 1.1: From [xkcd 3076](#)

In the double slit experiment, a light source is placed behind a wall with two narrow, closely spaced slits. On the other side, a photosensitive plate is positioned.

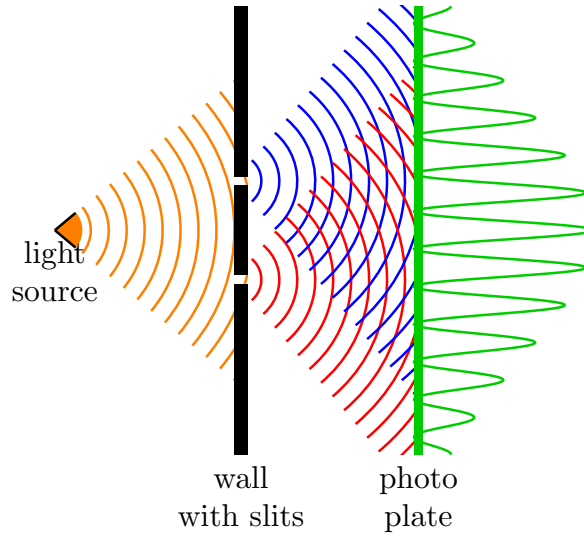


Figure 1.2: Double-slit experiment setup

An interference pattern appears on this photo plate, i.e. alternating light and dark stripes. This is due to the wave character of light. As the light waves pass through the slits, they overlap and interfere with each other. In some areas they have the same amplitude and reinforce each other, creating bright stripes. In other areas, the amplitudes have different signs and the waves cancel each other out, creating dark stripes.

This behaviour is to be expected. This is because the light travels different distances from the two slits to the same spot on the photo plate. If the difference is half a wavelength the two light waves cancel out at that spot.

Now we take individual photons. Here we would expect the interference pattern to disappear, as each photon can only pass through one of the slits. In that case, no interference can occur between photons coming from the two slits since they never meet. We would expect just two overlapping bright areas.

Surprisingly this behaviour does not occur. Even with single photons an interference pattern continues to appear. The photons do not decide to pass through one specific slit. They are in “superposition” between these two paths. This means that a single photon has two possibilities where it came from, both possibilities still happen at the same time and can cause interference with each other. For this reason, the amplitudes also add up or cancel out at the photo plate, resulting in the same interference pattern.

In later chapters, the mathematics shown will make this behaviour easier to understand.

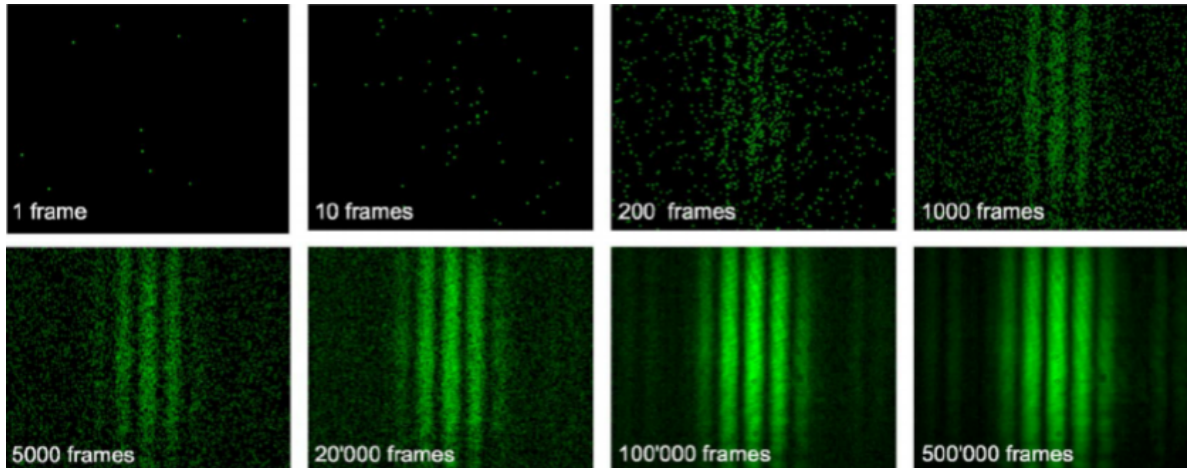


Figure 1.3: Resulting interference pattern when using single photons

1.2 What is a quantum computer?

To start into the topic of quantum computing and to understand the differences from classical computers, we first need to look at some of the basics of such classical computers.

In a classical computer the information is stored in *bits* which can either be in the state 0 or the state 1. These bits can be manipulated through different classical operations and we can look at these bits and read them, without interfering with the system or changing any states.

In a quantum computer the information is stored in a *qubit* which can be in a superposition *between* the state 0 and 1. Just as with classical computers, we can construct variables from these qubits to store bigger numbers. For example a 64-*qubit* integer would be described by 64 qubits which are in a superposition between 0 and $2^{64} - 1$. This can be imagined best as a variable where the universe has not yet decided on its value and therefore the variable has all possible values at the same time.

We can now use this superposition and manipulate it with different quantum operations. Contrary to a classical computer, in a quantum computer these operations are “applied” at all possible input values at the same time and the result is a superposition of all possible results of the operation. We call this effect *quantum parallelism*.

Example: Quantum parallelism

Let's say you have a quantum variable x in a superposition of numbers between 0 and $2^{64} - 1$ (all possible 64-bit values) and some function $f(x)$. You program a quantum computer to compute $f(x)$.

The quantum computer would compute $f(x)$ for $x = 0, x = 1, x = 2, \dots$ at the the same time and the result of this computation is a superposition of all possible values $f(x)$.

Reading this, one might be tempted to utilize quantum parallelism to run any algorithm on a quantum computer in order to optimize runtime. Unfortunately there is a big catch with quantum computers: If we try to look at the state of a qubit (also called *measuring*), the universe decides randomly on an outcome and therefore when measuring we only get the result of one computation and all the rest of the information is lost.

Example (continued): Quantum parallelism

After your quantum computer has calculated a superposition of all possible values $f(x)$, you want to get some information on the output and therefore you do a measurement on the resulting quantum state.

You will receive one random $f(x)$ and all the other possible solutions are lost.

Due to this restriction, naively running established algorithms on a quantum computer will not work. Fortunately there are some clever tricks to create some “interference” between different computations before measuring. This will give us useful information in some cases.

2 Probabilistic systems

To describe a quantum computer mathematically, we can do math similar to the known topic of probabilistic systems. We therefore first look into describing a probabilistic system.

2.1 Deterministic possibilities

At first we need to define all the different possible outcomes of our system. For example, for a coin flip this could be *heads* or *tails* and for a dice this could be the labels of the different sides. We call these possibilities *deterministic possibilities*. Note that we will only be using a *finite* number of possibilities.

Example: Random 2-bit number

Imagine you have a random number generator, which outputs 2-bit numbers. The deterministic possibilities of this generator are 00, 01, 10 and 11.

We will always assume the deterministic possibilities to be ordered in some way (even if it is an arbitrary one). In the example above, the deterministic possibilities are 00, 01, 10, 11, not 00, 10, 01, 11. We will need this to know the order of entries in vectors and matrices later.

2.2 Probability distribution

Next, we need to assign each possibility a probability. We write this as $\Pr[x] = p$ where $p \in [0, 1]$ is the probability of the deterministic possibility x .

Example: Coin flip

For a coin flip the probability of heads would be $\Pr[\text{heads}] = \frac{1}{2}$ and the probability for tails would be $\Pr[\text{tails}] = \frac{1}{2}$.

If we combine all probabilities for all the possible outcomes and write them as a vector, we get a *probability distribution*. Here it comes in handy that we have a ordering on the deterministic possibilities. If the deterministic possibilities are x_1, \dots, x_n , their probabilities will be in the vector in that order.

Definition 2.1 (Probability distribution). A vector $d \in \mathbb{R}^n$ is a valid probability distribution iff $\sum d_i = 1$ and $\forall i \ d_i \geq 0$.

This vector has n entries, where each entry corresponds to a deterministic possibility x and the probability of x is $\Pr[x] = d_i$. The sum over all probabilities has to be 1 and each entry needs to be nonnegative in order to be a valid probability.

Example (continued): Coin flip

For a coin flip the probability distribution would be $d_{\text{coin}} \in \mathbb{R}^2$ with $d = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$.

Example (continued): Random 2-bit number

Recall your random 2-bit number generator from above. Imagine your generator outputs each deterministic possibility with equal probability, except for the possibility 00, which

is never generated. The corresponding probability distribution would be

$$d_{2\text{-bit}} = \begin{pmatrix} 0 \\ \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{pmatrix}.$$

2.3 Probabilistic processes

With a probability distribution, we can only describe the probabilities of possibilities without any knowledge of a prior state. We therefore add another element to our toolbox of probabilistic systems called a *probabilistic process*.

A probabilistic process is a collection of n probability distributions, where for each deterministic possibility i there is a probability distribution a_i . This means that if the system is in deterministic possibility i before the process is applied, the system will afterwards be distributed according to a_i . We can write this as a matrix, where each column is a probability distribution a_i .

Definition 2.2 (Probabilistic process). A matrix $A \in \mathbb{R}^{N \times N}$ is a valid probabilistic process iff for every column a of A , a is a valid probability distribution.

From Definition 2.1 we know that a valid probability distribution a has the properties $\sum a_i = 1$ and $\forall i \ a_i \geq 0$, therefore a matrix A is a probabilistic process iff $A \in \mathbb{R}^{N \times N}$ with $\sum a_i = 1$ and $\forall i \ a_i \geq 0$. Such a matrix is also called a *stochastic matrix*.

Example (continued): Random 2-bit number

Imagine a second device, which receives a 2-bit number as an input and flips both bits at the same time with a probability of $\frac{1}{3}$. The probability distributions for each of the deterministic possibility would then be

$$a_{00} = \begin{pmatrix} \frac{2}{3} \\ 0 \\ 0 \\ \frac{1}{3} \end{pmatrix}, a_{01} = \begin{pmatrix} 0 \\ \frac{2}{3} \\ \frac{1}{3} \\ 0 \end{pmatrix}, a_{10} = \begin{pmatrix} 0 \\ \frac{1}{3} \\ \frac{2}{3} \\ 0 \end{pmatrix} \text{ and } a_{11} = \begin{pmatrix} \frac{1}{3} \\ 0 \\ 0 \\ \frac{2}{3} \end{pmatrix}.$$

From this we can construct the process as a matrix from these processes as follows:

$$A_{\text{flip}} = \begin{pmatrix} a_{00} & a_{01} & a_{10} & a_{11} \end{pmatrix} = \begin{pmatrix} \frac{2}{3} & 0 & 0 & \frac{1}{3} \\ 0 & \frac{2}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & \frac{2}{3} & 0 \\ \frac{1}{3} & 0 & 0 & \frac{2}{3} \end{pmatrix}.$$

Applying a probabilistic process

Having defined probability distributions and probabilistic processes, we can now combine these two elements and apply a probabilistic process on a probability distribution.

Definition 2.3 (Applying a probabilistic process). Given an initial probability distribution $x \in \mathbb{R}^n$ and a probabilistic process $A \in \mathbb{R}^{n \times n}$, the result $y \in \mathbb{R}^n$ of applying the process A is defined as

$$y = Ax.$$

Example (continued): Random 2-bit number

Recall the 2-bit number generator and the bit flip from above. Imagine you would first draw a random 2-bit number from the generator and then run the bit flip device. We already know that the probability distribution of the generator is $d_{2\text{-bit}}$. Using A_{flip} we can calculate the final probability distribution:

$$A_{\text{flip}} \cdot d_{2\text{-bit}} = \begin{pmatrix} \frac{2}{3} & 0 & 0 & \frac{1}{3} \\ 0 & \frac{2}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & \frac{2}{3} & 0 \\ \frac{1}{3} & 0 & 0 & \frac{2}{3} \end{pmatrix} \begin{pmatrix} 0 \\ \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{pmatrix} = \begin{pmatrix} \frac{1}{9} \\ \frac{1}{3} \\ \frac{1}{3} \\ \frac{2}{9} \end{pmatrix}.$$

3 Quantum systems

With the basics for a probabilistic system defined, we now look into describing a quantum computer mathematically. In the following table you can see the analogy from the quantum world to the probabilistic world.

Probabilistic world	Quantum world
Probability distributions	Quantum states
Probabilities	Amplitudes
Deterministic possibilities	Classical possibilities
Stochastic matrix as process	Unitary matrix as process

3.1 Classical possibilities

Like in the probabilistic systems we need to define all outcomes for a quantum system. For example, a photon can be in the state *up* or *down*. We call these possibilities *classical possibilities*. Note that we will only be using a *finite* number of possibilities.

Example: Random bit

Imagine you have a random bit generator, which outputs one bit. The classical possibilities of this generator are 0 and 1.

We will always assume the classical possibilities to be ordered in some way (even if it is an arbitrary one), like the deterministic possibilities. In the example above, the classical possibilities are 0, 1 not 1, 0. We will need this to know the order of entries in vectors and matrices later.

3.2 Quantum states

One of the most important element of the quantum world is a quantum state. A quantum state describes the state of a quantum system as a vector. Each entry of the vector represents a *classical* possibility (similar to the deterministic possibilities in a probability distribution). The entries of a quantum state are called *amplitude*. In contrast to a probabilistic system, these entries can be negative and are also complex numbers.

These amplitudes tell us the probability of the quantum state being in the corresponding classical possibility. To calculate the probabilities from the amplitude, we can take the square of the absolute value of the amplitude.

This means that for the classical possibility x and a quantum state ψ the probability for x is $\Pr[x] = |\psi|^2$. To have valid probabilities, the sum of all probabilities need to sum up to 1. From this we get the formal definition of a quantum state:

Definition 3.1 (Quantum State). A quantum state is a vector $\psi \in \mathbb{C}^n$ with $\sqrt{\sum_{i=1}^n |\psi_i|^2} = 1$.

Example: Some Quantum states

The following vectors are valid quantum states with the classical possibilities 0 and 1:

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad |+\rangle := \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}, \quad |-\rangle := \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix}.$$

Note that the symbol $|\rangle$ is not yet introduced, so just understand it as some label at this point. The probabilities for each state can be calculated as follows:

$$\begin{aligned} |0\rangle : \quad & \Pr[0] = |1|^2 = 1 & \Pr[1] = |0|^2 = 0, \\ |1\rangle : \quad & \Pr[0] = |0|^2 = 0 & \Pr[1] = |1|^2 = 1, \\ |+\rangle : \quad & \Pr[0] = |\frac{1}{\sqrt{2}}|^2 = \frac{1}{2} & \Pr[1] = |\frac{1}{\sqrt{2}}|^2 = \frac{1}{2}, \\ |-\rangle : \quad & \Pr[0] = |\frac{1}{\sqrt{2}}|^2 = \frac{1}{2} & \Pr[1] = |-\frac{1}{\sqrt{2}}|^2 = \frac{1}{2}. \end{aligned}$$

We can see here that two different quantum states ($|+\rangle$ and $|-\rangle$) can have the same probabilities for all classical possibilities.

3.3 Unitary transformation

We now have defined quantum states and need a way to describe some processes, which we want to apply on the quantum states. In the probabilistic world, we have stochastic matrices for this, but unfortunately we can not use these matrices on quantum states, since the output of applying these on a quantum state is not guaranteed to be a quantum state again. We therefore look for a different property of a matrix for which the outcome of applying that matrix is guaranteed to be a quantum state. The following Lemma is therefore useful.

Lemma 3.1 (Unitary matrix). *For a square matrix U , the following are equivalent:*

- U maps every quantum state to a quantum state,
- $U^\dagger U = I$ and $U U^\dagger = I$,
- $U^\dagger U = I$,
- all columns are quantum states and mutually orthogonal.

Definition 3.2 (Unitary transformation). A matrix U is called *unitary* iff $U^\dagger U = I$ and $UU^\dagger = I$.

Then the evolution of a quantum state is always described by a unitary matrix. So if the current state is ψ and we apply the transformation matrix U , the state is $U\psi$ afterwards.

A unitary matrix is by definition invertible, therefore we can undo all unitary transformations by applying U^\dagger .

Example: Some Unitary transformations

The following matrices are examples for unitary transformations:

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

These matrices are called Pauli-matrices, we will get to know them later on.

As an example for applying a unitary on a quantum state, we apply the Pauli X matrix on the quantum state $|0\rangle$:

$$X|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle.$$

4 Observing probabilistic and measuring quantum systems

So far we only talked about the description of a probabilistic and a quantum system. We now look into observing/measuring those systems.

4.1 Observing a probabilistic system

Observing a probabilistic system is the process of learning the outcome from a probability distribution. If our probability distribution for example represents a coin flip, observing this distribution is equivalent to actually flipping the coin. In the probabilistic case, an observation is just about updating our knowledge or beliefs. This will be different in the quantum case.

Definition 4.1 (Observing a probabilistic system). Given a probability distribution $d \in \mathbb{R}^n$, we will get the outcome i with a probability d_i . The new distribution is then

$$e_i = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \leftarrow 1 \text{ at the } i\text{-th position.}$$

The intuition for the new distribution is that we know after observing that i is the deterministic possibility for sure.

When observing a probabilistic system, the observation is just a passive process with no impact on the system. This means that there is no difference to the end result, whether we observe during the process or not. We take a look at an example to further understand this.

Example: Random 1-bit number

We use a random 1-bit number example similar to the random 2-bit example from Chapter 2. We have a distribution $d_{1\text{-bit}} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$ which represents the probability distribution of

generating a 1-bit number with equal probability. We also have a process $A_{\text{flip}} = \begin{pmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \end{pmatrix}$ which flips the bit with a probability of $\frac{1}{3}$.

We look at two different cases: For the first case, we observe only the final distribution and for the second case we observe after the generation of the 1-bit number and we also observe the final distribution.

Observing the final distribution

From Section 2.3 we know that the final distribution d is

$$d = A_{\text{flip}} \cdot d_{1\text{-bit}} = \begin{pmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \end{pmatrix} \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}.$$

We observe this distribution and will get outcome 0 and the new distribution $d = e_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ with a probability of $\Pr[0] = d_0 = \frac{1}{2}$. We get the outcome 1 and the new distribution

$d = e_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ with a probability of $\Pr[1] = d_1 = \frac{1}{2}$.

Observing after generation and the final distribution

We now observe the system after the generation of the 1-bit number and also observe the final distribution. After the generation, we will get outcome 0 and the new distribution $d = e_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ with a probability of $\Pr[0] = d_0 = \frac{1}{2}$. We get the outcome 1 and the new distribution $d = e_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ with a probability of $\Pr[1] = d_1 = \frac{1}{2}$.

We now apply in each case the matrix A_{flip} . This will give us the outcome $A_{\text{flip}} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{2}{3} \\ \frac{1}{3} \end{pmatrix}$ for the case of the outcome 0 and the outcome $A_{\text{flip}} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{3} \\ \frac{2}{3} \end{pmatrix}$ for the case of the outcome 1. If we observe the distribution $\begin{pmatrix} \frac{2}{3} \\ \frac{1}{3} \end{pmatrix}$, we will get the outcome 0 and the new distribution $d = e_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ with a probability of $\Pr[0] = \frac{2}{3}$ and the outcome 1 and the new distribution $d = e_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ with a probability of $\Pr[1] = \frac{1}{3}$. If we observe the distribution $\begin{pmatrix} \frac{1}{3} \\ \frac{2}{3} \end{pmatrix}$, we will get the outcome 0 and the new distribution $d = e_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ with a probability of $\Pr[0] = \frac{1}{3}$ and the outcome 1 and the new distribution $d = e_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ with a probability of $\Pr[1] = \frac{2}{3}$.

Combining these probabilities, we get the total probability $\Pr[0] = \frac{1}{2} \frac{2}{3} + \frac{1}{2} \frac{1}{3} = \frac{1}{2}$ for the outcome 0 and the probability $\Pr[1] = \frac{1}{2} \frac{1}{3} + \frac{1}{2} \frac{2}{3} = \frac{1}{2}$ for the outcome 1. This is the same as observing the final distribution.

4.2 Measuring a quantum system

Unlike in the probabilistic system, the “observation” of a quantum system is called *measuring*. The definition is similar to the observation of a probabilistic system, except that we need to take the absolute square of the amplitude to get the probability and that the state after measuring is called *post-measurement-state* (p.m.s.).

Definition 4.2 (Measuring a quantum system). Given a quantum State $\psi \in \mathbb{C}^n$, we will

get the outcome i with a probability $|\psi_i|^2$. The post-measurement-state (p.m.s.) is then

$$e_i = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \leftarrow 1 \text{ at the } i\text{-th position.}$$

This is called a complete measurement in the computational basis.

With this similarity to the probabilistic observation in the definition, one might assume that measuring a quantum state has also no impact on the system. **This is not the case, measuring a quantum state changes the system!** We can see this effect with an example:

Example: Measuring a quantum system

Let $\psi = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$ be a quantum state and $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ be a unitary transformation.

We look at two different cases: First we apply H immediately and then measure the system. As a second case, we do a measurement before the application of the H unitary and then a measurement after applying it.

Measure the final state

We first calculate the state after applying H :

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

Measuring this state will get the outcome 0 with probability $\Pr[0] = |\psi_0|^2 = 1$ and have the post-measurement-state $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$. The outcome 1 can never occur, i.e. $\Pr[1] = |\psi_1|^2 = 0$

Measure the initial and the final state

Measuring ψ with no further unitary matrices applied can have the outcome 0 or 1. We will look at the final measurement for each case:

The first measurement will have outcome 0 with probability $\Pr[0] = |\psi_0|^2 = \frac{1}{2}$ and the post-measurement-state will be $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$. H applied to this post-measurement-state will be

$H \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$. When measuring this state, we will get the outcome 0 with probability

$\Pr[0] = |\frac{1}{\sqrt{2}}|^2 = \frac{1}{2}$ and outcome 1 with probability $\Pr[1] = |\frac{1}{\sqrt{2}}|^2 = \frac{1}{2}$.
 The outcome 1 will appear at the initial state with probability $\Pr[1] = |\psi_1|^2 = \frac{1}{2}$ and
 the post-measurement-state will be $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$. H applied to this post-measurement-state will
 be $H \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix}$. When measuring this state, we will get the outcome 0 with
 probability $\Pr[0] = |\frac{1}{\sqrt{2}}|^2 = \frac{1}{2}$ and outcome 1 with probability $\Pr[1] = |-\frac{1}{\sqrt{2}}|^2 = \frac{1}{2}$.
 So independent of the outcome of the first measurement, at the second measurement
 the outcome 0 and 1 have a probability of $\frac{1}{2}$. This shows that when measuring before
 applying H , we will receive different probabilities for the second measurement, then when
 measuring only at the end. This proves that measurements can change the system.

4.3 Elitzur–Vaidman bomb tester

This section will be updated later on.

5 Partial observing and measuring systems

In the previous chapter, we looked into observing a probabilistic and measuring a quantum system. In this approach, we always looked at the full system. This means that we either have no measurement at all or we know the exact possibility, in which our system is.

For larger systems, this can become quite complicated, as we might not need the full measurement, but only some partial information. For example if we consider a dice throw, we might not need the final number of the dice, but we are only interested if it is an even or an odd number. To archive this, we can do a partial observation on a probabilistic system.

5.1 Partially observing a probabilistic system

To perform a partial observation on a probabilistic system, we first decide on which *alternatives* we want to distinguish. Each alternative is described by a set A of deterministic possibilities. By performing the partial observation, we will get for each alternative A the probability that the system is in a deterministic state in A .

Definition 5.1 (Partially observing a probabilistic system). Given a probabilistic system with deterministic possibilities $X = (x_1, \dots, x_n)$, a distribution $\mu \in \mathbb{R}^N$ and a family of alternatives A_1, \dots, A_m with $A_i \cap A_j = \emptyset$ and $\bigcup_i A_i = X$, the probability of observing the alternative k is given by

$$\Pr[\text{outcome} = k] = \sum_{x_i \in A_k} \mu_{(x_i)}.$$

The distribution v after the observation of the outcome k is given by the (normalized) conditional distribution:

$$v = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix} \text{ with } v_i := \begin{cases} \frac{\mu_i}{\Pr[\text{outcome}=k]} & \text{if } x_i \in A_k \\ 0 & \text{if } x_i \notin A_k \end{cases}.$$

Note: In this definition we were careful to distinguish between the names x_i of the deterministic possibilities and their number i (e.g. when writing μ_i). We will often be less precise and simply pretend the deterministic possibilities are the number $1, \dots, N$. That is, we would write the definition as follows and pretend it means the above:

Definition 5.2 (Partially observing a probabilistic system). Given a distribution $\mu \in \mathbb{R}^N$ and a family of alternatives $A_1, \dots, A_m \subseteq \{1, \dots, N\}$ with $A_i \cap A_j = \emptyset$ and $\bigcup_i A_i = \{1, \dots, N\}$, the probability of observing the alternative k is given by

$$\Pr[\text{outcome} = k] = \sum_{i \in A_k} \mu_i.$$

The distribution v after the observation of the outcome k is given by the conditional distribution:

$$v = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix} \text{ with } v_i := \begin{cases} \frac{\mu_i}{\Pr[\text{outcome}=k]} & \text{if } i \in A_k \\ 0 & \text{if } i \notin A_k \end{cases}.$$

Note that similar to the full observation of a probabilistic system a partial observation does not actually change the system. We only get some new knowledge. In particular, a third person can never notice whether we observed the system or not.

Example: Partially observing a probabilistic system

A fair dice was rolled and it is only known that it is not a 5. Thus the distribution μ is given by

$$\mu = \begin{pmatrix} \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \\ 0 \\ \frac{1}{5} \end{pmatrix}.$$

Now we want to observe whether the number is low (≤ 3) or high (≥ 4). This means we have two alternatives: $A_{\text{low}} = \{1, 2, 3\}$ and $A_{\text{high}} = \{4, 5, 6\}$. We therefore obtain the following probabilities for these two alternatives

$$\Pr[\text{outcome} = \text{low}] = \frac{1}{5} + \frac{1}{5} + \frac{1}{5} = \frac{3}{5},$$

$$\Pr[\text{outcome} = \text{high}] = \frac{1}{5} + 0 + \frac{1}{5} = \frac{2}{5}.$$

The conditional distribution after the outcome low is

$$\begin{pmatrix} \frac{1/5}{3/5} \\ \frac{1/5}{3/5} \\ \frac{1/5}{3/5} \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

That is, we know we have a uniformly random number from $\{1, 2, 3\}$, but don't know which. And after the outcome high the conditional distribution is

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{1/5}{2/5} \\ 0/2/5 \\ \frac{1/5}{2/5} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{2} \\ 0 \\ \frac{1}{2} \end{pmatrix}.$$

5.2 Partially measuring a quantum system

Similar to the partial observation of a probabilistic system, we can perform a partial measurement on a quantum system.

Definition 5.3 (Partially measuring a quantum system). Given a quantum system with classical possibilities $X = (x_1, \dots, x_n)$, a quantum state $\mu \in \mathbb{C}^N$ and a family of alternatives A_1, \dots, A_m with $A_i \cap A_j = \emptyset$ and $\bigcup_i A_i = X$, the probability of observing the alternative k is given by

$$\Pr[\text{outcome} = k] = \sum_{x_i \in A_k} |\psi_{(x_i)}|^2.$$

The post-measurement-state of the outcome k is computed as follows:

1. Computing the non-normalized post-measurement-state $\phi^{(k)}$ denoted by $\phi^{(k)} := (\phi_1, \dots, \phi_N)$ with

$$\phi_i := \begin{cases} \psi_i & \text{if } x_i \in A_k \\ 0 & \text{if } x_i \notin A_k \end{cases}.$$

2. Computing the normalized post-measurement-state by calculating:

$$\text{post-measurement-state} := \frac{\phi^{(k)}}{\|\phi^{(k)}\|} = \frac{\phi^{(k)}}{\sqrt{\Pr[\text{outcome} = k]}}.$$

As in Definition 5.1, we were precise about the difference between the classical possibility x_i and their numbers but will not always be so precise in the future.

As with the complete measurement for quantum systems, the measurement can change the system. Note that there exist other types of definitions for a measurement e.g. projective measurements, generalized measurements, POVMs, ... The variant above can best be described as a “projective measurement in the computational basis”.

There is a slight difference between this definition and Definition 4.2, namely if you compute the post-measurement-state, you may get a different result. The two post-measurement-states can differ by a factor $c \in \mathbb{C}$ with $|c| = 1$, called a “global phase”. Such a global phase makes no observable physical difference, so this “contradiction” is not a problem.

Example: Partially measuring a quantum system

A photon is in superposition between the 4 paths left, right, top and bottom:

$$\psi = \begin{pmatrix} \frac{1}{10} \\ -\frac{3}{10} \\ \frac{9}{10}i \\ \frac{3}{10} \end{pmatrix}.$$

There are two alternatives: $A_{\text{horizontal}}$, so that the photon is in the left or right path, and A_{vertical} , so that the photon is in the top or bottom path. We therefore obtain the

following probabilities for these two alternatives

$$\Pr[A_{\text{horizontal}}] = \left| \frac{1}{10} \right|^2 + \left| -\frac{3}{10} \right|^2 = \frac{1}{100} + \frac{9}{100} = \frac{1}{10},$$

$$\Pr[A_{\text{vertical}}] = \left| \frac{9}{10}i \right|^2 + \left| \frac{3}{10} \right|^2 = \frac{81}{100} + \frac{9}{100} = \frac{9}{10}.$$

The normalized post-measurement-state for the alternative $A_{\text{horizontal}}$ is

$$\begin{pmatrix} \frac{1}{10} \\ -\frac{3}{10} \\ 0 \\ 0 \end{pmatrix} / \sqrt{\frac{1}{10}} = \begin{pmatrix} \frac{1}{\sqrt{10}} \\ -\frac{3}{\sqrt{10}} \\ 0 \\ 0 \end{pmatrix}$$

For the alternative A_{vertical} the normalized post-measurement-state is

$$\begin{pmatrix} 0 \\ 0 \\ \frac{9}{10}i \\ \frac{3}{10} \end{pmatrix} / \sqrt{\frac{9}{10}} = \begin{pmatrix} 0 \\ 0 \\ \frac{3}{\sqrt{10}}i \\ \frac{1}{\sqrt{10}} \end{pmatrix}.$$

6 Composite Systems

So far our probabilistic and quantum systems consist of only one single distribution/state. In the real world, quantum computers often have several different registers (variables).

In theory, we could use a single very big distribution/state to model multiple qubits. For example a 10 qubit system could be modeled with the classical possibilities 0000000000, 0000000001, ..., 1111111110, 1111111111.

Unfortunately the vector for these states gets really big, for 10 qubits, the vector would have the dimension of 1024. Since this is very inconvenient to write down, we need to look at a different solution. For this, we compose different probabilistic or quantum systems with each other.

6.1 Constructing composite systems

Definition 6.1 (Composite systems / Tensor product). Given two probabilistic or quantum systems A and B with the possibilities of A given by x_1, \dots, x_N and a distribution/state μ_A and with the possibilities of B given by y_1, \dots, y_M and a distribution/state μ_B , the *composite* system called AB has the possibilities

$$x_1y_1, x_1y_2, \dots, x_1y_M, x_2y_1, x_2y_2, \dots, x_2y_M, \dots, x_Ny_1, x_Ny_2, \dots, x_Ny_M$$

and the distribution/state μ_{AB} of AB is given by the *tensor product*

$$\mu_{AB} := \mu_A \otimes \mu_B = \begin{pmatrix} (\mu_A)_1 \cdot \mu_b \\ \vdots \\ (\mu_A)_N \cdot \mu_b \end{pmatrix}.$$

This vector has the size NM . Here $(\mu_A)_i$ stands for the i -th entry of μ_A .

The definition of combining a probabilistic and a quantum system are the same.

Notice that the entry corresponding to the possibility x_iy_j in the composite system is then $(\mu_{AB})_{x_iy_j} = (\mu_A)_{x_i}(\mu_B)_{y_j}$. Here we identify the classical possibility x_i and y_j with the indices $1, \dots, N$ and $1, \dots, M$, respectively the classical possibility x_iy_j with the indices $1, \dots, NM$. (So $(\mu_{AB})_{x_iy_j}$ has just one index, namely $x_iy_j \in \{1, \dots, NM\}$.)

Example: Composite system

Let the distributions for the system A with the possibilities $1, 2$ and the system B with the possibilities a, b, c be given by

$$\mu_A = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}, \quad \mu_B = \begin{pmatrix} \frac{1}{2} \\ 0 \\ \frac{\sqrt{3}}{2} \end{pmatrix}.$$

Then the composite system AB has the possibilities $1a, 1b, 1c, 2a, 2b, 2c$ and the distribution

$$\mu_{AB} = \mu_A \otimes \mu_B = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \otimes \begin{pmatrix} \frac{1}{2} \\ 0 \\ \frac{\sqrt{3}}{2} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \cdot \frac{1}{2} \\ \frac{1}{\sqrt{2}} \cdot 0 \\ \frac{1}{\sqrt{2}} \cdot \frac{\sqrt{3}}{2} \\ -\frac{1}{\sqrt{2}} \cdot \frac{1}{2} \\ -\frac{1}{\sqrt{2}} \cdot 0 \\ -\frac{1}{\sqrt{2}} \cdot \frac{\sqrt{3}}{2} \end{pmatrix} = \begin{pmatrix} \frac{1}{2\sqrt{2}} \\ 0 \\ \frac{\sqrt{3}}{2\sqrt{2}} \\ -\frac{1}{2\sqrt{2}} \\ 0 \\ -\frac{\sqrt{3}}{2\sqrt{2}} \end{pmatrix}.$$

We now need a way to apply operators on these combined systems. For this we can also construct the tensor product of either two probabilistic processes or two unitary transformations by using the tensor product of two matrices.

Definition 6.2 (Composite matrices / Tensor product). Given two matrices S and T with S of the size $N \times N$ and T of the size $M \times M$. The tensor product $S \otimes T$ of is given by

$$S \otimes T = \begin{pmatrix} S_{11}T & \dots & S_{1N}T \\ \vdots & \ddots & \vdots \\ S_{N1}T & \dots & S_{NN}T \end{pmatrix}.$$

Overall we can say: If we apply S to the system A and T to the system B , we apply $S \otimes T$ to the composite system AB .

If the distribution d_{AB} of a given probabilistic system AB can be written as a composite of two distributions d_A and d_B , we know that A and B are independent of each other. If we cannot write d_{AB} as two separate distributions, the probabilities are depended on each other.

If the quantum state ψ_{AB} of a given quantum system AB can be written as a composite of two different quantum states ψ_A and ψ_B , the quantum states of A and B are independent of each other. If we can not write ψ_{AB} as a tensor product of two quantum systems, the quantum states depend on each other. We call this *entangled*.

Lemma 6.1. For the (unitary) matrices A, B, C and D , the vectors (quantum states) ψ, ϕ and χ and the constant c , the following applies

- $(A \otimes B)(\psi \otimes \phi) = A\psi \otimes B\phi$,
- $(A \otimes B)(C \otimes D) = AC \otimes BD$,
- $\psi \otimes (\phi + \chi) = (\psi \otimes \phi) + (\psi \otimes \chi)$,
- $(\psi + \phi) \otimes \chi = (\psi \otimes \chi) + (\phi \otimes \chi)$,
- $A \otimes (B + C) = (A \otimes B) + (A \otimes C)$,
- $(A + B) \otimes C = (A \otimes C) + (B \otimes C)$,
- $c\phi \otimes \psi = c(\phi \otimes \psi)$,
- $\phi \otimes c\psi = c(\phi \otimes \psi)$,
- $A \otimes cB = c(A \otimes B)$ and
- $cA \otimes B = c(A \otimes B)$.

These rules only apply if the dimensions of the matrices and vectors match.

6.2 Measuring composite systems

To perform a (partial) observation or (partial) measurement on a composite system AB , we can compose two separate measurements on the systems A and B similar as we constructed the tensor product.

Definition 6.3 (Composite measurements). Given two systems A and B with possibilities $1, \dots, N$ and $1, \dots, M$ and two partial measurements M_A and M_B on systems A and B with alternatives $A_1, \dots, A_N \subseteq \{x_1, \dots, x_n\}$ and $B_1, \dots, B_M \subseteq \{y_1, \dots, y_M\}$. The measurement $M_A \otimes M_B$ on AB is a measurement with the alternatives $C_{11}, C_{12}, \dots, C_{NM}$ where $C_{ij} = A_i \times B_j$.

If we only have a set of alternatives for system A , we can do a measurement $M_A \otimes I$ with alternatives $C_1, \dots, C_N := A \otimes \{y_1, \dots, y_M\}$.

Example: Composite measurement (quantum)

Let A be a system with the states $1, 2, 3$ and μ_A a measurement with the two alternatives $A_{\text{low}} = \{1, 2\}$, $A_{\text{high}} = \{3\}$. The quantum state is

$$\psi_A = \begin{pmatrix} \frac{2}{3} \\ \frac{1}{3} \\ -\frac{2}{3} \end{pmatrix}.$$

Another system B has the states a, b, c . The measurement μ_B the two alternatives $B_{\text{vocal}} = \{a\}$, $B_{\text{consonant}} = \{b, c\}$. The quantum state is

$$\psi_B = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2}i \\ \frac{1}{\sqrt{2}} \end{pmatrix}.$$

So the composite system $C = AB$ has the classical possibilities $1a, 1b, 1c, 2a, 2b, 2c, 3a, 3b$ and $3c$. The measurement $\mu_C := \mu_A \otimes \mu_B$ has the alternatives

$$\begin{aligned} C_{\text{low, vocal}} &= \{1a, 2a\}, & C_{\text{low, consonant}} &= \{1b, 1c, 2b, 2c\}, \\ C_{\text{high, vocal}} &= \{3a\}, & C_{\text{high, consonant}} &= \{3b, 3c\}. \end{aligned}$$

The quantum state is

$$\psi_C = \psi_A \otimes \psi_B = \begin{pmatrix} \frac{2}{3} \\ \frac{1}{3} \\ -\frac{2}{3} \end{pmatrix} \otimes \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2}i \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} \frac{2}{6} \\ \frac{2}{6}i \\ \frac{2}{3\sqrt{2}} \\ \frac{1}{6} \\ \frac{1}{6}i \\ \frac{1}{3\sqrt{2}} \\ -\frac{2}{6} \\ -\frac{2}{6}i \\ -\frac{2}{3\sqrt{2}} \end{pmatrix}.$$

We get for the alternative $C_{\text{low, vocal}}$ the probability

$$\left|\frac{2}{6}\right|^2 + \left|\frac{1}{6}\right|^2 = \frac{4}{36} + \frac{1}{36} = \frac{5}{36}$$

and thus the post-measurement-state is

$$\begin{pmatrix} \frac{2}{6} \\ 0 \\ 0 \\ \frac{1}{6} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} / \sqrt{\frac{5}{36}} = \begin{pmatrix} \frac{2}{6} \\ 0 \\ 0 \\ \frac{1}{6} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} / \frac{\sqrt{5}}{6} = \begin{pmatrix} \frac{2}{\sqrt{5}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{5}} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

7 Quantum Circuits

In the previous chapters, we learned the basics on how to construct a quantum computer. We will now start constructing quantum circuits from these. Note that we will no longer look into probabilistic systems.

The quantum systems which we consider in the following sections consist of qubits, unless specified otherwise. A qubit is a quantum state ψ with $\psi \in \mathbb{C}^2$.

7.1 Visual language

So far we have only seen the elements of quantum computers in a mathematical form (i.e., as formulas). When constructing quantum circuits, this can get very unreadable very fast. Therefore we can draw quantum circuits as a picture, which also helps us to get a better intuition for these circuits. You can see a very simple example here:

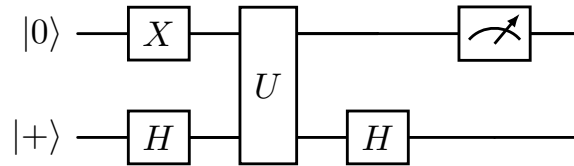


Figure 7.1: A basic quantum circuit

In this circuit we have two qubits $|0\rangle$ and $|+\rangle$, which are drawn as separate wires. Note that the symbol $|\rangle$ is introduced in the next chapter, so just understand it as a name for some state at this point. We first apply the unitary X on the top wire to $|0\rangle$ and at the same time we apply the unitary H at the bottom wire to $|+\rangle$. Mathematically this can be written as $(X \otimes H)(|0\rangle \otimes |+\rangle)$. Next we apply the unitary U , which operates on both qubits. After this, we apply a unitary H on the bottom wire. Since we do not apply anything on the top wire, we can write this mathematically as $I \otimes H$. Finally we measure the top qubit. This means a complete measurement in the computational basis of the qubit as described in Section 4.2. The meaning of the unitaries used is explained in the next section. A wire can contain multiple qubits, depending on the context.

7.2 Important gates

When working with quantum computers, we encounter some of the same unitaries very often. We distinguish between single qubit gates (unitary transformations $\in \mathbb{C}^{2 \times 2}$) and gates on multiple qubits.

7.2.1 Single qubit gates

The following gates are relevant single qubit gates:

Definition 7.1 (Identity matrix). The identity matrix I is defined as

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

This matrix is for example useful if a qubit/wire is to remain unchanged. The identity matrix also exists in other sizes.

Definition 7.2 (Pauli matrices). The Pauli matrices X, Y and Z are defined as

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

Note that X is also called bit-flip.

Definition 7.3 (Hadamard gate). The Hadamard gate H is defined as

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

The Hadamard gate is useful for introducing superpositions as it takes a classical bit $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and transforms it into a superposition $\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$.

7.2.2 Controlled-NOT gate

The gates introduced above only operate on a single qubit. To connect two different qubits, we need gates which operate on multiple qubits. For this we introduce the controlled-not:

Definition 7.4 (Controlled-NOT gate). The controlled-NOT gate $\text{CNOT} \in \mathbb{C}^{4 \times 4}$ is defined as

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

The CNOT gate flips the qubit of the second qubit if the first qubit is 1. We call the first wire the controlling wire and the second wire the target wire. It can be drawn in a quantum circuit as follows:

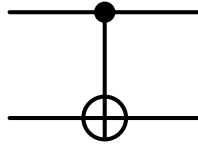


Figure 7.2: Controlled-NOT in a quantum circuit

If the second qubit should be the controlling wire and the first qubit the target wire, we can use CNOT' denoted as

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

Accordingly, the quantum circuit is drawn the other way round.

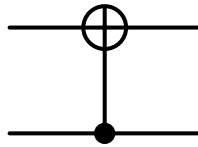


Figure 7.3: Controlled-NOT' in a quantum circuit

7.3 Teleportation

We are now looking at an example quantum circuit.

Example: Teleportation

Assumed: Alice has a qubit ψ and wants to send it to Bob. But only classic communication is possible, no quantum communication. However, they can share a state β_{00} beforehand.

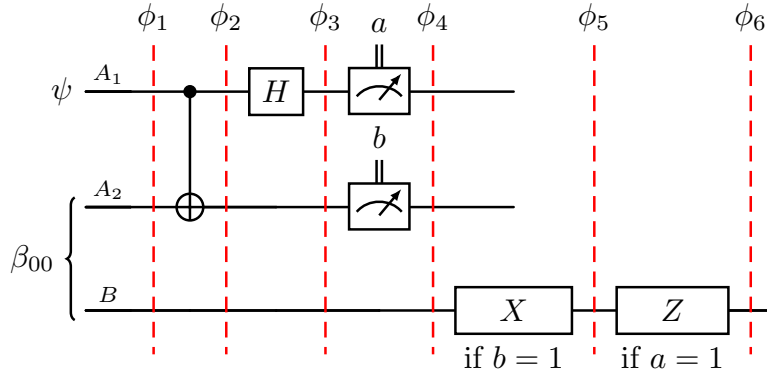


Figure 7.4: Circuit for qubit teleportation

1. Alice has the state $\psi = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ and the shared state is $\beta_{00} = \begin{pmatrix} 1/\sqrt{2} \\ 0 \\ 0 \\ 1/\sqrt{2} \end{pmatrix}$. This means that the entire state is

$$\phi_1 = \psi \otimes \beta_{00} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} / \sqrt{2} = \begin{pmatrix} \alpha \\ 0 \\ 0 \\ \alpha \\ \beta \\ 0 \\ 0 \\ \beta \end{pmatrix} / \sqrt{2}.$$

2. The CNOT can be extended to $\text{CNOT} \otimes \text{I}_2$ using the identity matrix:

$$\phi_2 = (\text{CNOT} \otimes \text{I}_2) \phi_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ 0 \\ 0 \\ \alpha \\ \beta \\ 0 \\ 0 \\ \beta \end{pmatrix} / \sqrt{2} = \begin{pmatrix} \alpha \\ 0 \\ 0 \\ \alpha \\ 0 \\ \beta \\ \beta \\ 0 \end{pmatrix} / \sqrt{2}.$$

3. Identical to step 2, the Hadamard gate can be extended with the identity matrix:

$$\phi_3 = (H \otimes I_4)\phi_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} \alpha \\ 0 \\ 0 \\ \alpha \\ 0 \\ \beta \\ \beta \\ 0 \end{pmatrix} \frac{1}{\sqrt{2}} = \frac{1}{2} \begin{pmatrix} \alpha \\ \beta \\ \beta \\ \alpha \\ \alpha \\ -\beta \\ -\beta \\ \alpha \end{pmatrix}.$$

4. Here we assume that $a = 0$ and $b = 1$. It applies $|\alpha|^2 + |\beta|^2 = 1$ because ψ is a quantum state. Therefore the probability for this is

$$\left| \frac{\beta}{2} \right|^2 + \left| \frac{\alpha}{2} \right|^2 = \frac{|\alpha|^2 + |\beta|^2}{4} = \frac{1}{4}$$

and the post-measurement-state

$$\phi_4 = \frac{1}{2} \begin{pmatrix} 0 \\ 0 \\ \beta \\ \alpha \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \frac{1}{\sqrt{\frac{1}{4}}} = \begin{pmatrix} 0 \\ 0 \\ \beta \\ \alpha \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

5. Since $b = 1$, the Pauli-matrix X is used:

$$\phi_5 = (I_4 \otimes X)\phi_4 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \beta \\ \alpha \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \alpha \\ \beta \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

6. Since $a = 0$, nothing happens in this step:

$$\phi_6 = \phi_5 = \begin{pmatrix} 0 \\ 0 \\ \alpha \\ \beta \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \otimes \psi.$$

So now Bob is aware of ψ and Alice has the now useless state $(0 \ 1 \ 0 \ 0)^T$. Note that Bob would also have ψ for all other results of a and b .

8 Ket Notation

So far we have only seen vectors as a way to mathematically describe a quantum state. This can get quite inconvenient if the vector get bigger and also often contains not that much useful information (e.g. a lot of 0 entries). We therefore introduce a new form of writing quantum states called the *ket* notation.

The idea works as follows: We can rewrite a quantum state ψ in the following way

$$\psi = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_N \end{pmatrix} = \psi_1 \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \psi_2 \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} + \dots + \psi_N \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} = \sum_{i=1}^N \psi_i \cdot e_i.$$

The vector e_i denotes the vector with 0 entries at every position except the i -th position, where the entry is 1.

From this notation we already get an advantage, since we can drop out all 0-entries. But we still have no intuitive mapping from the vector e_i to the classical possibility represented by e_i . For example, e_{123} can represent the classical possibility “red,4,top”. For this we use a $| \rangle$ symbol. More precise this means for a classical possibility x , which is the i -th possibility and

is represented by e_i , we write

$$|x\rangle := e_i = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \leftarrow 1 \text{ at the } i\text{-th position.}$$

In the example, we would therefore write $|\text{red},4,\text{top}\rangle$ for e_{123} .

Example: Ket notation

Given a quantum system with the classical possibilities 00, 01, 10 and 11, the quantum state $\psi = \left(\frac{1}{\sqrt{2}} \ 0 \ 0 \ \frac{1}{\sqrt{2}}\right)^T$ can be written as

$$\psi = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \frac{1}{\sqrt{2}} |00\rangle + \frac{1}{\sqrt{2}} |11\rangle.$$

Written like this, we can see at first glance that this is a superposition of the classical possibilities 00 and 11. Writing $\psi = \frac{1}{\sqrt{2}}e_1 + \frac{1}{\sqrt{2}}e_4$ would be less obvious.

Note that the ket notation can also be used in a few other ways. We can use it as described above to the state $|x\rangle$ corresponding to the classical possibility x , but we also use it to emphasize that ψ is a quantum state by writing $|\psi\rangle$ (here ψ is not a classical possibility). We also have two special cases $|+\rangle$ and $|-\rangle$ which are defined as follows:

$$\begin{aligned} |+\rangle &:= \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle, \\ |-\rangle &:= \frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle. \end{aligned}$$

Which of the meanings of the symbol $|\rangle$ is meant has to be deduced from the context.

8.1 Teleportation

We take another look at the example from the last chapter with ket notation.

Example: Teleportation

Once again, Alice has the qubit ψ and Alice and Bob have shared the state β_{00} .

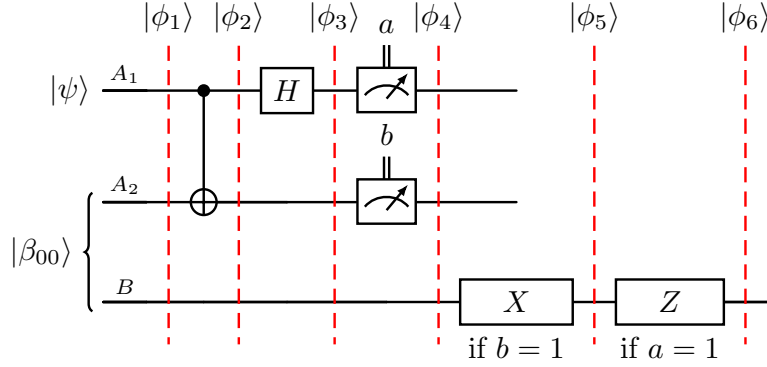


Figure 8.1: Circuit for qubit teleportation

1. Alice has the state $|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha|0\rangle + \beta|1\rangle$ and the shared state is $|\beta_{00}\rangle = \left(\frac{1}{\sqrt{2}} \ 0 \ 0 \ \frac{1}{\sqrt{2}}\right)^T = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$. This means that the entire state is

$$\begin{aligned} |\phi_1\rangle &= |\psi\rangle \otimes |\beta_{00}\rangle \\ &= (\alpha|0\rangle + \beta|1\rangle) \otimes \left(\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle\right) \\ &= \frac{\alpha}{\sqrt{2}}|000\rangle + \frac{\alpha}{\sqrt{2}}|011\rangle + \frac{\beta}{\sqrt{2}}|100\rangle + \frac{\beta}{\sqrt{2}}|111\rangle. \end{aligned}$$

2. We can now translate each ket notation individually and get the result much simpler:

$$\begin{aligned} |\phi_2\rangle &= (\text{CNOT} \otimes I_2) |\phi_1\rangle \\ &= (\text{CNOT} \otimes I_2) \left(\frac{\alpha}{\sqrt{2}}|000\rangle + \frac{\alpha}{\sqrt{2}}|011\rangle + \frac{\beta}{\sqrt{2}}|100\rangle + \frac{\beta}{\sqrt{2}}|111\rangle \right) \\ &= \frac{\alpha}{\sqrt{2}}|000\rangle + \frac{\alpha}{\sqrt{2}}|011\rangle + \frac{\beta}{\sqrt{2}}|110\rangle + \frac{\beta}{\sqrt{2}}|101\rangle. \end{aligned}$$

3. Identical to step 2, we can look at each ket notation individually:

$$\begin{aligned}
|\phi_3\rangle &= (H \otimes I_4) |\phi_2\rangle \\
&= (H \otimes I_4) \left(\frac{\alpha}{\sqrt{2}} |000\rangle + \frac{\alpha}{\sqrt{2}} |011\rangle + \frac{\beta}{\sqrt{2}} |101\rangle + \frac{\beta}{\sqrt{2}} |110\rangle \right) \\
&= \frac{\alpha}{\sqrt{2}} (H |0\rangle \otimes |00\rangle) + \frac{\alpha}{\sqrt{2}} (H |0\rangle \otimes |11\rangle) + \frac{\beta}{\sqrt{2}} (H |1\rangle \otimes |01\rangle) + \frac{\beta}{\sqrt{2}} (H |1\rangle \otimes |10\rangle) \\
&= \frac{\alpha}{\sqrt{2}} \left(\left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) \otimes |00\rangle \right) + \frac{\alpha}{\sqrt{2}} \left(\left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) \otimes |11\rangle \right) + \\
&\quad \frac{\beta}{\sqrt{2}} \left(\left(\frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle \right) \otimes |01\rangle \right) + \frac{\beta}{\sqrt{2}} \left(\left(\frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle \right) \otimes |10\rangle \right) \\
&= \frac{\alpha}{2} |000\rangle + \frac{\alpha}{2} |100\rangle + \frac{\alpha}{2} |011\rangle + \frac{\alpha}{2} |111\rangle + \frac{\beta}{2} |001\rangle - \frac{\beta}{2} |101\rangle + \frac{\beta}{2} |010\rangle - \frac{\beta}{2} |110\rangle.
\end{aligned}$$

4. We again assume that $a = 0$ and $b = 1$ and therefore only $\frac{\alpha}{2} |011\rangle$ and $\frac{\beta}{2} |010\rangle$ are relevant. It applies $|\alpha|^2 + |\beta|^2 = 1$ because ψ is a quantum state. Therefore the probability for this is

$$\left| \frac{\beta}{2} \right|^2 + \left| \frac{\alpha}{2} \right|^2 = \frac{|\alpha|^2 + |\beta|^2}{4} = \frac{1}{4}$$

and the post-measurement-state

$$|\phi_4\rangle = \frac{\frac{\alpha}{2} |011\rangle}{\sqrt{\frac{1}{4}}} + \frac{\frac{\beta}{2} |010\rangle}{\sqrt{\frac{1}{4}}} = \alpha |011\rangle + \beta |010\rangle = |01\rangle \otimes (\alpha |1\rangle + \beta |0\rangle).$$

5. Since $b = 1$, the Pauli-matrix X is used:

$$\begin{aligned}
|\phi_5\rangle &= (I_4 \otimes X) |\phi_4\rangle \\
&= (I_4 \otimes X) \left(|01\rangle \otimes (\alpha |1\rangle + \beta |0\rangle) \right) \\
&= I_4 |01\rangle \otimes (X(\alpha |1\rangle + \beta |0\rangle)) \\
&= |01\rangle \otimes (\alpha X |1\rangle + \beta X |0\rangle) \\
&= |01\rangle \otimes (\alpha |0\rangle + \beta |1\rangle).
\end{aligned}$$

6. Since $a = 0$, nothing happens in this step:

$$|\phi_6\rangle = |\phi_5\rangle = |01\rangle \otimes (\alpha |0\rangle + \beta |1\rangle) = |01\rangle \otimes |\psi\rangle.$$

As expected, we get the same result as in the previous chapter.

The ket notation can save a lot of work and sources of error. (Keep in mind that the example here got a bit lengthy because we wrote out a lot of intermediate steps.)

9 Bernstein-Vazirani Algorithm

With all the quantum basics from the previous chapters, we now can start with the first quantum algorithm. This algorithm is called the Bernstein-Vazirani algorithm.

This algorithm tackles the following problem: Given a secret $s \in \{0,1\}^n$ and the function $f : \{0,1\}^n \rightarrow \{0,1\}$, defined as $f(x) := x \cdot s$. \cdot denotes the inner product of two bitstrings here. This means that for bitstrings x and y of length n , the inner product is $x \cdot y = x_1y_1 + \dots + x_ny_n \bmod 2$.

The goal is to find the secret s using as little queries of f as possible. By “query” we mean an evaluation of f . The word query stems from the fact that we often think of the algorithm having access to a so-called “oracle” which we can “query” to get $f(x)$.

Classically we will need at least n queries to f to get s definitely. A classical algorithm with only $m \leq n$ queries will get s with a probability of 2^{m-n} if s is uniformly random.

We will now look at a quantum algorithm which will find s with only one evaluation of f . This algorithm is sketched in the following circuit:

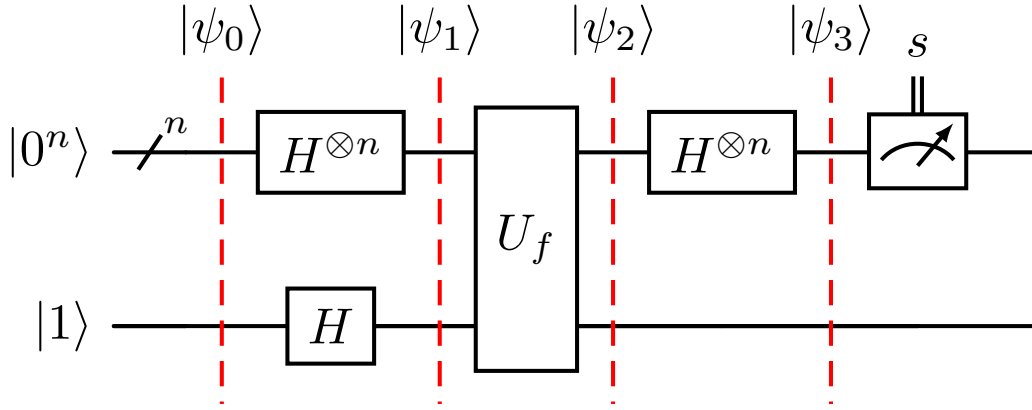


Figure 9.1: The quantum circuit for Bernstein-Vazirani

Note that U_f is defined with the explanation below.

We start with n qubits on the top wire. All of these qubits are in the state $|0\rangle$, which we write $|0\rangle^n = |0\rangle \otimes \dots \otimes |0\rangle$. The bottom wire is in the state $|1\rangle$. Both wires composed together can be written as $|\psi_0\rangle = |0^n 1\rangle = |0\rangle^n \otimes |1\rangle$, which is the overall starting state of our algorithm. We now perform the following steps

1. First we apply a Hadamard gate on all qubits. This is denoted for the first n qubits by the $H^{\otimes n}$ gate and for the last qubit by the H gate on the bottom wire. The resulting quantum state is calculated as follows:

$$\begin{aligned}
|\psi_1\rangle &= (H^{\otimes n} \otimes H) (|\psi_0\rangle) \\
&= (H^{\otimes n} \otimes H) (|0\rangle^n \otimes |1\rangle) \\
&= (H^{\otimes n} |0\rangle^n) \otimes H |1\rangle \\
&= |+\rangle^{\otimes n} \otimes |-\rangle \\
&= \left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right)^{\otimes n} \otimes |-\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \otimes |-\rangle
\end{aligned}$$

Roughly speaking, we are now in the superposition over all classical possibilities on the top wire and in $|-\rangle$ on the bottom wire.

2. Next, we apply the unitary U_f on both wires. This unitary is defined as

$$U_f |x, y\rangle = |x, y \oplus f(x)\rangle$$

This unitary represents the function f and combines the output of $f(x)$ with the bottom wire y . For our quantum states, this means that the state after U_f can be calculated as follows:

$$\begin{aligned}
|\psi_2\rangle &= U_f |\psi_1\rangle \\
&= U_f \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \otimes |-\rangle \\
&= U_f \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \otimes |-\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} U_f (|x\rangle \otimes |-\rangle) \\
&\stackrel{*}{=} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle \otimes |-\rangle \\
&= \left(\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle \right) \otimes |-\rangle
\end{aligned}$$

Note that the $*$ holds since we can rewrite $U_f(|x\rangle \otimes |-\rangle)$ as

$$\begin{aligned}
U_f(|x\rangle \otimes |-\rangle) &= \frac{1}{\sqrt{2}}U_f|x, 0\rangle - \frac{1}{\sqrt{2}}U_f|x, 1\rangle \\
&= \frac{1}{\sqrt{2}}|x, f(x)\rangle - \frac{1}{\sqrt{2}}|x, \overline{f(x)}\rangle \\
&= \begin{cases} \frac{1}{\sqrt{2}}|x, 0\rangle - \frac{1}{\sqrt{2}}|x, 1\rangle & f(x) = 0 \\ \frac{1}{\sqrt{2}}|x, 1\rangle - \frac{1}{\sqrt{2}}|x, 0\rangle & f(x) = 1 \end{cases} \\
&= \begin{cases} |x\rangle \otimes |-\rangle & f(x) = 0 \\ -|x\rangle \otimes |-\rangle & f(x) = 1 \end{cases} \\
&= (-1)^{f(x)} |x\rangle \otimes |-\rangle
\end{aligned}$$

The bottom wire has not changed and is still $|-\rangle$. But on the top wire, we now have $f(x)$ somehow encoded into our quantum state. The phenomenon that the output of f is encoded as a -1 in the input register is called phase kickback. Measuring this quantum state would not give us any advantage, since we would just get one random x . We therefore perform one final step before measuring.

3. As the final unitary, we perform another $H^{\otimes n}$ on the top wire. We hope that the result of this unitary transformation is the state $|\psi_3\rangle = |s\rangle \otimes |-\rangle$. To check, whether our hopes become reality, we can calculate $(H^{\otimes n})^\dagger |s\rangle \otimes |-\rangle$ and check if it is equal to $|\psi_2\rangle$. We do

it in this direction, since these calculations are a bit simpler:

$$\begin{aligned}
\left((H^{\otimes n})^\dagger \otimes I\right) |\psi_3\rangle &= (H^{\otimes n})^\dagger |s\rangle \otimes |-\rangle \\
&= H^{\otimes n} |s\rangle \otimes |-\rangle \\
&= H^{\otimes n} (|s_1\rangle \otimes \cdots \otimes |s_n\rangle) \otimes |-\rangle \\
&= H |s_1\rangle \otimes \cdots \otimes H |s_n\rangle \otimes |-\rangle \\
&= \bigotimes_{i=1}^n \left(\frac{1}{\sqrt{2}} |0\rangle + (-1)^{s_i} \frac{1}{\sqrt{2}} |1\rangle \right) \otimes |-\rangle \\
&= \frac{1}{\sqrt{2^n}} \bigotimes_{i=1}^n (|0\rangle + (-1)^{s_i} |1\rangle) \otimes |-\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} \left((-1)^{x_1 s_1} (-1)^{x_2 s_2} \cdots (-1)^{x_n s_n} |x\rangle \right) \otimes |-\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} \left((-1)^{\sum_i x_i s_i \bmod 2} |x\rangle \right) \otimes |-\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{s \cdot x} |x\rangle \otimes |-\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle \otimes |-\rangle \\
&= |\psi_2\rangle
\end{aligned}$$

This calculation shows that we have the quantum state $|s\rangle \otimes |-\rangle$ before the measurement.

4. We now perform a measurement on the top wire and measure s as a result.

This concludes the Bernstein-Vazirani algorithm.

10 Shor's Algorithm

One of the best known quantum algorithm is Shor's algorithm for finding the prime factors of an integer. It was developed by Peter Shor in 1994.

10.1 Discrete Fourier Transformation

One of the tools required for Shor's algorithm is the Discrete Fourier Transformation (DFT). Generally, a Fourier transformation is a mathematical technique that decomposes a function into its constituent frequencies. We use the DFT to find the period of a vector.

The DFT is defined as follows:

Definition 10.1 (Discrete Fourier Transformation (DFT)). The discrete Fourier transform (DFT) is a linear transformation on \mathbb{C}^N represented by the matrix

$$\text{DFT}_N = \frac{1}{\sqrt{N}}(\omega^{kl})_{kl=0,\dots,N-1} \in \mathbb{C}^{N \times N}$$

with $\omega = e^{2i\pi/N}$, which is the N -th root of unity.

It applies $\omega^N = 1$ and $\omega^M \neq 1$ for all $0 < M < N$.

This transformation is best imagined as a process, which takes a periodic vector as an input and outputs the period of that vector. The DFT has some important properties, which help us later on.

Theorem 10.1 (Properties of the DFT). *Here are some properties of the DFT which can be used without further proof.*

1. The DFT_N is unitary.
2. $\omega^t = \omega^{t \bmod N}$ for all $t \in \mathbb{Z}$.
3. Let $t \mid N$ and s be fix variables and the quantum state $\psi \in \mathbb{C}^N$ given by

$$|\psi_i| = \begin{cases} \sqrt{\frac{t}{N}} & \text{if } i = at + s \text{ for some } a \\ 0 & \text{else.} \end{cases}$$

In other words ψ is t -periodic. Then for $\phi := \text{DFT}_N$ it applies

$$|\phi_i| = \begin{cases} \frac{1}{\sqrt{t}} & \text{if } \frac{N}{t} \mid i \\ 0 & \text{else.} \end{cases}$$

The first peak of ϕ is at $\frac{N}{t}$.

10.2 Reducing factoring to period finding

With the DFT, we have seen, that we can use a unitary to find the period of a quantum state. We now look into using period finding to factor integers. We first look at the definition of some problems:

Definition 10.2 (Factoring problem). Given integer N with two prime factors $p, q > 2$ such that $pq = N$ and $p \neq q$, find p and q .

Note that this definition of the factoring problem is a simplified version of the factoring problem, where N has only 2 prime factors.

Definition 10.3 (Period finding problem). Given $f : \mathbb{Z} \rightarrow X$ with $f(x) = f(y)$ iff $x \equiv y \pmod{r}$ for some fixed secret r , find r .
We call r the *period* of f .

To start the reduction, we need a special case of the period finding problem called order finding:

Definition 10.4 (Order finding problem). For known a and N which are relatively prime, find the period r of $f(i) = a^i \pmod{N}$. We call r the order of a written $r = \text{ord } a$. (This is similar to finding the smallest $i > 0$ with $f(i) = a^i \pmod{N} = 1$).

Since the order finding problem is just the period finding problem for a specific $f(x)$, we know that if we can solve the period finding problem within reasonable runtime, we can also solve the order finding problem within reasonable runtime. We now reduce the factoring problem to the order finding problem:

We have an integer N as an input for the factoring problem.

1. Pick an $a \in \{2, \dots, N-1\}$ relatively prime to N .
2. Compute the order of a , so that $r := \text{ord mod } a$ (using one solver for the order finding problem).
3. If the order r is odd, restart at 1.
4. Calculate $x := a^{\frac{r}{2}} + 1 \pmod{N}$ and $y := a^{\frac{r}{2}} - 1 \pmod{N}$.
5. If $\gcd(x, N) \in \{1, N\}$, we restart at 1.
6. Return $p = \gcd(x, N)$ and $q = \frac{N}{\gcd(y, N)}$.

The output of the reduction are p, q , such that $pq = N$. This holds, since

$$xy = (a^{\frac{r}{2}} + 1)(a^{\frac{r}{2}} - 1) = a^r - 1 \equiv 1 - 1 = 0 \pmod{N}.$$

This means that $N \mid xy$ and therefore $p \mid xy$ and $q \mid xy$. From this it can be concluded that $p \mid x$, $q \mid y$, $p \mid y$ and $q \nmid x$ which leads to $\gcd(x, N) = p$. Or p and q swapped.

Theorem 10.2 (Probability of an abort). *If N has at least two different prime factors and N is odd, then the probability to restart is $\leq \frac{1}{2}$.*

All in all this reduction shows, that if we have an oracle which can solve the period finding problem within reasonable runtime, we can also solve the factoring problem within reasonable runtime (since all other operations are classically fast to compute).

10.3 The quantum algorithm for period finding

We now look into an quantum algorithm that solves the period finding problem within reasonable runtime.

For the quantum circuit we need an $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ which is r -periodic with $r < 2^n$.

The quantum algorithm for period finding is shown in this figure:

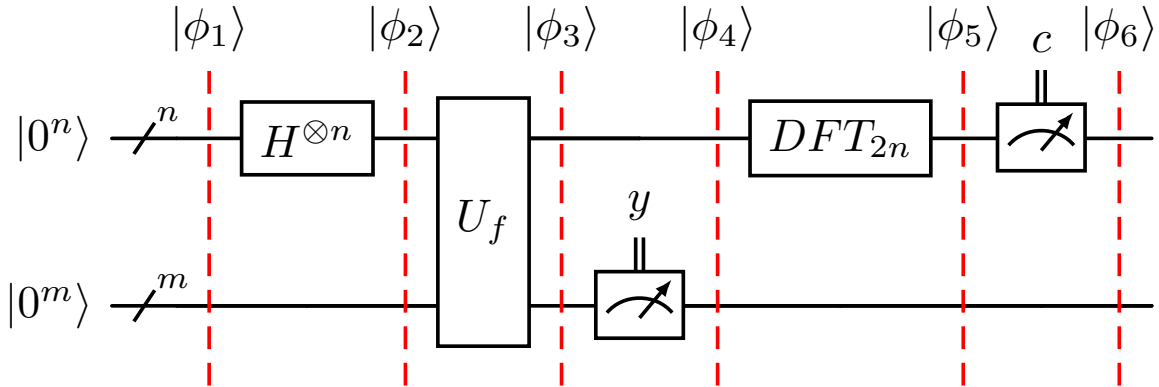


Figure 10.1: Shor's algorithm (quantum part)

The algorithm works as follows:

1. We start with $|\psi_1\rangle = |0^n\rangle \otimes |0^m\rangle$.
2. We bring the top wire into the superposition over all entries. The quantum state is then $|\psi_2\rangle = G \cdot \sum_{x \in \{0, \dots, 2^n-1\}} |x\rangle \otimes |0^m\rangle$ with the constant $G := 2^{-\frac{n}{2}}$.
3. We apply U_f , which is the unitary of $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$. This calculates the superposition over all possible values $f(x)$ on the bottom wire. The resulting quantum state is $|\psi_3\rangle = G \cdot \sum_{x \in \{0, \dots, 2^n-1\}} |x, f(x)\rangle$.
4. To understand the algorithm better, we measure the bottom wire at this point. This will give us one random value $y = f(x_0)$ for some x_0 . The top wire will then contain a superposition over all values x where $f(x) = f(x_0)$. Since f is known to be r -periodic, we

know, that $f(x) = f(x_0)$ iff $x \equiv x_0 \pmod{r}$. This means, that the resulting quantum state on the top wire is periodic. So the complete quantum state is $|\psi_4\rangle = C \cdot \sum_{x \equiv x_0 \pmod{r}} |x\rangle \otimes |f(x_0)\rangle$ with the constant $C = \frac{\sqrt{r}}{\sqrt{2^n}}$.

5. We apply the Discrete Fourier Transform on the top wire. This will “analyze” the top wire for the period and output a vector with entries at multiples of $\frac{2^n}{r}$ as seen in Theorem 10.1.
6. We measure the top wire and get one random multiple of $\frac{2^n}{r}$, which we can denote as $c = a \cdot \frac{2^n}{r}$ for some a .

Since we get a multiple of $\frac{2^n}{r}$ on each run, we can simply run the algorithm multiple times to get different multiples and then compute $\frac{2^n}{r}$ by taking the gcd of those multiples. From that we compute r . Unfortunately this only works because we assumed $r \mid 2^n$. Since this does usually not hold, we only get approximate multiples of $\frac{2^n}{r}$ (which is not even an integer) and thus we need a post processing.

10.4 Post processing

So far we have seen the DFT to analyze the period of a quantum state, we have seen a way to reduce the factoring problem to the period finding and we have seen a quantum algorithm for finding an approximate multiple of such a period of a function. We just need one final step to find r . For this we start with a theorem:

Theorem 10.3. *Iff $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ is r -periodic, the following holds with probability $\Omega(\frac{1}{\log \log r})$:*

$$\frac{-r}{2} \leq rc \pmod{2^n} \leq \frac{r}{2}$$

where c is the output of the second measurement of the quantum circuit described in Section 10.3 and n is the number of qubits on the upper wire of the quantum circuit.

We assume that the theorem holds for our outcome c of the second measurement (if that is not the case, our result will be wrong and we can just run the quantum algorithm again to get a different outcome) and R is an upper bound on r :

Then exists an integer d such that:

$$\begin{aligned} |rc - d2^n| &\leq \frac{r}{2} \quad || \text{ divide by } r \cdot 2^n \\ \Leftrightarrow \left| \frac{c}{2^n} - \frac{d}{r} \right| &\leq \frac{1}{2^{n+1}} \end{aligned}$$

The fraction $\varphi := \frac{c}{2^n}$ is known, but the fraction $\frac{d}{r}$ is unknown. All we know is that it is a fraction and that denominator is $\leq R$. The goal is to find a fraction $\frac{1}{2^{n+1}}$ -close to $\frac{c}{2^n}$.

For this we use another theorem:

Theorem 10.4. *Under the conditions $r^2 \leq 2^n$ and 2^n is in the order of R^2 , the fraction $\frac{d}{r}$ is a convergent of the continued fraction expansion of $\frac{c}{2^n}$, where c, d, r and n are the variables defined above.*

This theorem uses the convergent of a continued fraction expansion. A continued fraction expansion of a number t is the number rewritten as a fraction in the form

$$t = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \dots}}}$$

where a_i always has to be the biggest possible integer. We call $[a_0, a_1, a_2, a_3, \dots]$ the continued expansion of t . The expansion is finite iff t is rational. For a given continued expansion, a prefix $[a_0, \dots, a_i]$ is called a convergent. Writing this convergent as a normal fraction will give us an approximation of the number t .

Example: Continued expansion of a fraction

The number 2.3 can be written as

$$2.3 = 2 + \frac{1}{3 + \frac{1}{3+0}}$$

and the continued fraction expansion of 2.3 is $[2, 3, 3]$. The expansions $[2]$ and $[2, 3]$ are convergents of the expansion of 2.3 and written as a fraction will give us the approximations 2 and $2 + \frac{1}{3} = 2.\bar{3}$.

The number 0.99 can be written as

$$0.99 = 0 + \frac{1}{1 + \frac{1}{99+0}}$$

and the continued fraction expansion of 0.99 is $[0, 1, 99]$. The expansions $[0]$ and $[0, 1]$ are convergents of the expansion of 0.99 and written as a fraction will give us the approximations 0 and $0 + \frac{1}{1} = 1$.

Using Theorem 10.4 (with $\varphi := \frac{c}{2^n}$ and $q := 2^n$) we can find $\frac{d}{r}$ and from this r which is the period of our function using the following steps:

For each convergent γ of φ do the following:

1. Compute γ as fraction $\frac{d}{r}$.
2. Stop if $r \leq 2^n$ and this $\frac{1}{2^{n+1}}$ -close to $\frac{c}{2^n}$ and return r .

Note: It can happen, that the resulting fraction does not have the right r in the denominator, because $\frac{d}{r}$ was simplified (if numerator and denominator shared a common factor). But the probability of this happening is sufficiently small and already included in the probability in Theorem 10.3.

This completes the postprocessing of Shor's algorithm.

10.5 Constructing the DFT

So far we have described everything necessary for Shor's algorithm, but only described the matrix representation of the DFT_N . We will now take a closer look into implementing the DFT_M as a quantum circuit. Since we only use the DFT_N for Shor's algorithm so far, we will only look at $N = 2^n$, which is the DFT applied on n qubits.

To start the circuit, we recall the definition of the DFT_N from Definition 10.1: $\text{DFT}_N := \frac{1}{\sqrt{2^n}}(\omega^{kl})_{kl}$ with $\omega := e^{2\pi i/2^n}$. To apply the DFT_N to a quantum state $|j\rangle$ we calculate

$$\begin{aligned}
\text{DFT}_N |j\rangle &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{jk} |k\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k / N} |k\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{k_1 \in \{0,1\}} \dots \sum_{k_n \in \{0,1\}} e^{2\pi i j (\sum_{l=1}^n k_l 2^{-l})} |k_1 \dots k_n\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{k_1 \in \{0,1\}} \dots \sum_{k_n \in \{0,1\}} \bigotimes_{l=1}^n e^{2\pi i j k_l 2^{-l}} |k_l\rangle \\
&= \frac{1}{\sqrt{N}} \bigotimes_{l=1}^n \sum_{k_l \in \{0,1\}} e^{2\pi i j k_l 2^{-l}} |k_l\rangle \\
&= \frac{1}{\sqrt{N}} \bigotimes_{l=1}^n (|0\rangle + e^{2\pi i j 2^{-l}} |1\rangle) \\
&= \frac{1}{\sqrt{N}} \bigotimes_{l=1}^n (|0\rangle + e^{2\pi i 0 \cdot j_{n-l+1} \dots j_n} |1\rangle).
\end{aligned}$$

The expression $0.j$ expresses a binary fraction (e.g. $0.101 = \frac{1}{2} + \frac{1}{8} = \frac{5}{8}$).

With this we have shown, that we can write $\text{DFT}_N |j\rangle$ as the following tensor product of quantum states

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0 \cdot j_n} |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0 \cdot j_{n-1} j_n} |1\rangle) \otimes \dots \otimes \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0 \cdot j_1 \dots j_n} |1\rangle).$$

From this rewritten tensor product, we can get an idea on how to construct the quantum circuit for the DFT_N . Namely, we can construct a quantum circuit for each element of the tensor product and from this build the general circuit.

For this, we segment the tensor product into different elements ψ as follows:

$$\underbrace{\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0 \cdot j_n} |1\rangle)}_{\psi_1} \otimes \underbrace{\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0 \cdot j_{n-1} j_n} |1\rangle)}_{\psi_2} \otimes \dots \otimes \underbrace{\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0 \cdot j_1 \dots j_n} |1\rangle)}_{\psi_n}.$$

We also introduce a new gate R_k which is defined by the following matrix:

$$R_k := \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i / 2^k} \end{pmatrix}.$$

To understand the construction of the circuit from these elements, we will look at an example for $n = 3$ first:

Example: Construction of the DFT circuit for $n = 3$

We start by building the tensor product for $n = 3$. The input for the DFT circuit is $|j\rangle = |j_1 j_2 j_3\rangle$. Using the formula from above, we can write the result of $\text{DFT}_{2^3} |j\rangle$ as the following tensor product:

$$\underbrace{\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0 \cdot j_3} |1\rangle)}_{\psi_1} \otimes \underbrace{\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0 \cdot j_2 j_3} |1\rangle)}_{\psi_2} \otimes \underbrace{\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0 \cdot j_1 j_2 j_3} |1\rangle)}_{\psi_3}.$$

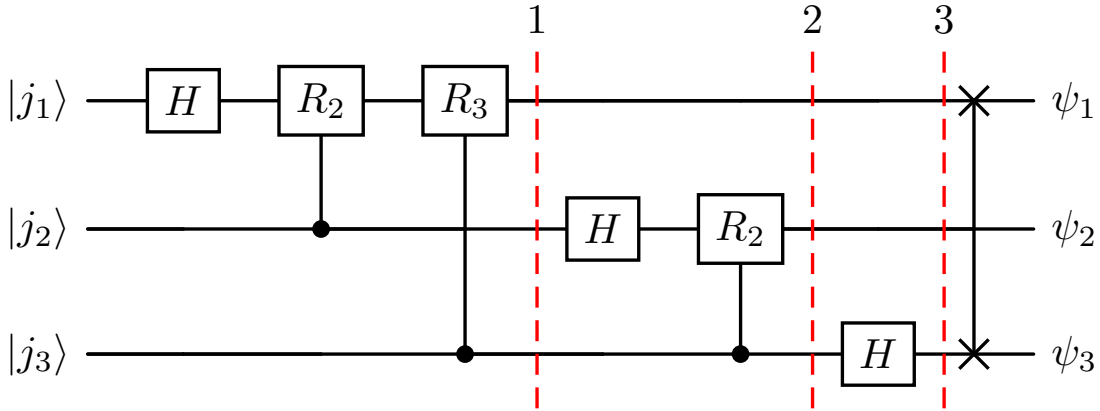


Figure 10.2: The DTF for three qubits

1. First we construct the ψ_3 element. Contrary to the intuition, we use the top wire containing $|j_1\rangle$ for this. We use a Hadamard-gate to bring $|j_1\rangle$ into the superposition $\frac{1}{\sqrt{2}}(|0\rangle + (-1)^{j_1} |1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0 \cdot j_1} |1\rangle)$. This looks close to ψ_3 already, we now need to add the last two decimal places $j_2 j_3$ to the state. For this we use R_2 and R_3 . We apply R_2 controlled by the wire j_2 and R_3 controlled by the wire j_3 . This means, that we only apply the R -gate, if the corresponding wire contains a 1. You can see this written as a quantum circuit at the figure below. After applying R_2 we have the state $\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0 \cdot j_1 j_2} |1\rangle)$ and after applying R_3 we have the state $\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0 \cdot j_1 j_2 j_3} |1\rangle)$ on the top wire. This is the same as ψ_3 , so we are done on the first wire (We are at the first slice in the figure).
2. The next step is to construct the ψ_2 state on the middle wire. We again use a Hadamard-gate to bring $|j_2\rangle$ into the superposition $\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0 \cdot j_2} |1\rangle)$. We now need to include the last decimal point j_3 , for which we use R_2 again, this time controlled by j_3 . The resulting superposition is now $\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0 \cdot j_2 j_3} |1\rangle)$, which is ψ_2 . (We are at the second slice in the figure).
3. On the bottom wire, we can just do a Hadamard-gate to bring $|j_3\rangle$ into the superposition $\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0 \cdot j_3} |1\rangle)$. We then have ψ_1 on the bottom wire. (We are at the third slice in the figure).
4. When applying this circuit, we get the state $\psi_3 \otimes \psi_2 \otimes \psi_1$ as a result. This very close to our desired state $\psi_1 \otimes \psi_2 \otimes \psi_3$, just the order of the wires is flipped. To solve this, we apply a SWAP onto all wires, which flips the order of the wires and delivers the correct output for DFT_{2^3} .

The more general approach to construct the DFT_N as a quantum circuit with n qubits ($N = 2^n$) works as follows:

1. Initialize wires with input $|j\rangle$, so that $|j_1\rangle$ is on the top wire and $|j_n\rangle$ is on the bottom wire. Note, that this is not part of the circuit yet.
2. Start with the top wire. For each wire j_i do the following:
 1. Apply a Hadamard-gate on the wire j_i .
 2. For each wire j_k below the current wire j_i (with $i < k \leq n$), add a R_{k-i+1} -gate controlled by j_k . Start with $k = i + 1$ (if $i < n$, else stop).
3. Perform a SWAP to flip all the wires. This means, that the first wire is swapped with the last wire, the second wire is swapped with the second to last wire and so on.

Note: If the the output of the DFT circuit is measured right after applying it (as in Shor's algorithm) or if the rest of the algorithm allows for it, it is more efficient to perform the SWAP classically, since this is considered to be the cheaper operation.

The more general layout of the quantum circuit for the DFT_N with the SWAP is shown in this figure.

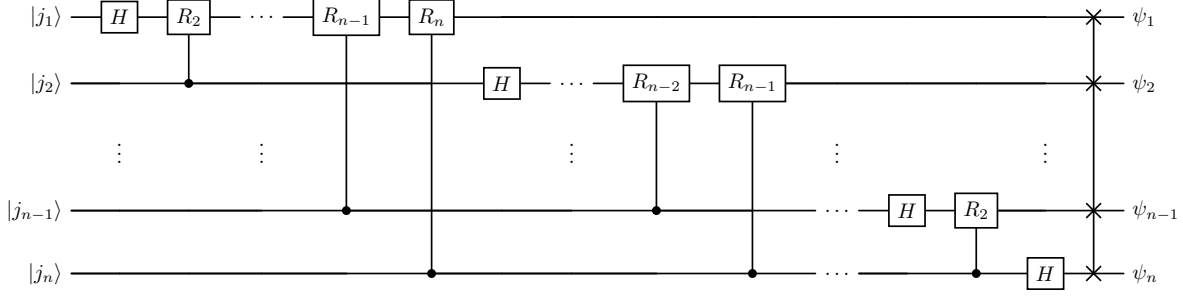


Figure 10.3: The DTF for n qubits

11 Grover's algorithm

Another well known quantum algorithm is Grover's algorithm for searching. It was developed by Lov Grover in 1996.

Grover's algorithm takes a function $f : \{0,1\}^n \rightarrow \{0,1\}$, where exactly one x_0 exists, such that $f(x_0) = 1$. The goal is to find x_0 .

There are a number of interesting problems, which can be reduced to this general definition. One of these problems is the breaking of a (symmetric) encryption. The function f would take a key as an input and output a 1, if the decryption is successful. Otherwise it will output a 0.

Classically, finding this x_0 takes approximately 2^n steps (when simply bruteforcing the function). Using Grover's algorithm, we can reduce this runtime to approximately $\sqrt{2^n}$ steps. As an example, a 128-bit encryption would only take about 2^{64} steps to break it, instead of about 2^{128} steps for the classical bruteforce.

11.1 Preparations

To construct Grover's algorithm, we first need to introduce two new gates V_f and FLIP_* .

11.1.1 Constructing the oracle V_f

In the previous algorithms, we have learned that we can implement a function f as a unitary U_f with $U_f |x, y\rangle = |x, y \oplus f(x)\rangle$. We construct a different unitary called V_f from this, which has the following behavior:

$$V_f |x\rangle = \begin{cases} -|x\rangle & \text{if } f(x) = 1 \\ |x\rangle & \text{else} \end{cases}$$

We can construct V_f from U_f using the following circuit:

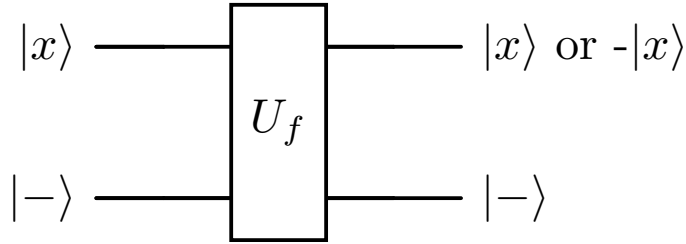


Figure 11.1: The circuit for V_f

The bottom wire can be discarded, since it always contains a $|-\rangle$ and thus is not entangled with the upper wire.

11.1.2 Constructing FLIP_*

As a second ingredient for Grover's algorithm, we need the unitary FLIP_* . To realize this, we first need FLIP_0 , which is defined by the unitary

$$\text{FLIP}_0 |x\rangle = \begin{cases} |0\rangle & \text{if } x = 0 \\ -|x\rangle & \text{else.} \end{cases}$$

FLIP_0 is implemented by the following quantum circuit:

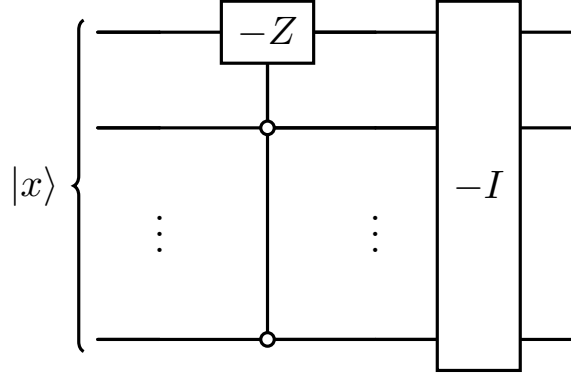


Figure 11.2: The circuit for $FLIP_0$

Z is the Pauli matrix from definition Definition 7.2. The empty circles indicates a negative control wire. So Z is only applied if the other wires are $|0\rangle$.

Now we can define the unitary called $FLIP_*$. This unitary does nothing, if it is applied on the uniform superposition $|*\rangle$. For any other quantum state $|\psi\rangle$ orthogonal to $|*\rangle$ it maps to $-|\psi\rangle$. The uniform superposition $|*\rangle$ simply denotes the superposition over all classical possibilities $|*\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$. So $FLIP_*$ is described by:

$$FLIP_* |\psi\rangle = \begin{cases} |*\rangle & \text{if } |\psi\rangle = |*\rangle \\ -|\psi\rangle & \text{if } |\psi\rangle \perp |*\rangle \text{ (orthogonal)}. \end{cases}$$

We can construct this $FLIP_*$ by the following quantum circuit:

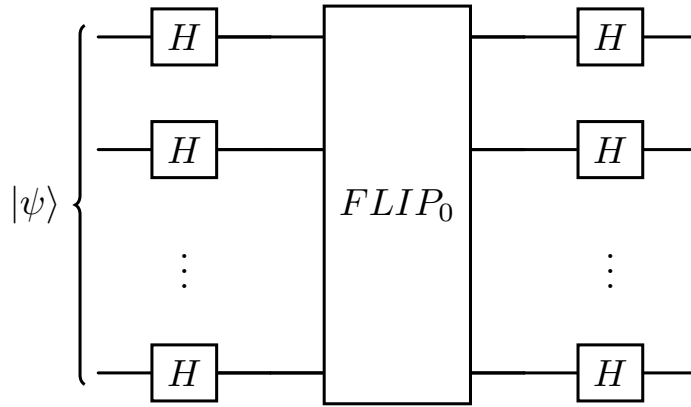


Figure 11.3: The circuit for $FLIP_*$

11.2 The algorithm for searching

The actual algorithm takes a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ and outputs an x_0 with $f(x_0) = 1$. For simplicity, we assume that there is only one x_0 for which $f(x_0) = 1$ holds and for each other $x \neq x_0$ it holds that $f(x) = 0$.

With the two new unitaries V_f and FLIP_* defined, we can construct the circuit for Grover's algorithm, which is shown in the following figure:

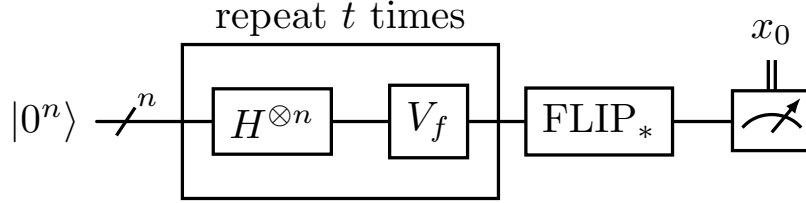


Figure 11.4: The quantum circuit for Grover's algorithm

The algorithm works as follows:

1. We start with a $|0\rangle$ entry on every qubit.
2. We bring the system into the superposition over all entries by applying $H^{\otimes n}$. The quantum state is then $\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$ which we also call $|*\rangle$.
3. We apply the unitary V_f .
4. We apply the unitary FLIP_* .
5. We repeat steps 3 and 4 t times.
6. We do a measurement.

The measurement in step 6 will then give us x_0 with high probability.

11.2.1 Understanding the algorithm for searching

When looking at the quantum circuit, it is not completely intuitive why the algorithm gives the correct result. We therefore now look into what is happening in each step.

The desired quantum state after the algorithm finishes is $|x_0\rangle$. At the beginning of the algorithm, we bring the system into the uniform superposition $|*\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$. We know that $|x_0\rangle$ is part of this superposition, therefore we can rewrite $|*\rangle$ as follows

$$|*\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle = \frac{1}{\sqrt{2^n}} \underbrace{|x_0\rangle}_{\text{good}} + \underbrace{\sqrt{\frac{2^n - 1}{2^n}} \sum_{x \neq x_0} \frac{1}{\sqrt{2^n - 1}} |x\rangle}_{\text{bad}}$$

So the current state can be seen as a superposition of a “good” state *good* and a “bad” state *bad*.

The geometric interpretation of this superposition can be drawn as follows:

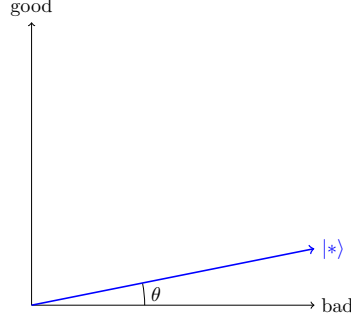


Figure 11.5: Geometric interpretation of $|*\rangle$

The angle θ denotes, how “good” the resulting outcome will be. If $\theta = 0$, the state is completely bad, if $\theta = \frac{\pi}{2}$, the state is completely good.

We can calculate $\cos \theta = |\langle * | bad \rangle| = \frac{\sqrt{2^n - 1}}{\sqrt{2^n}}$. From this we can derive that the angle θ is $\cos^{-1} \sqrt{\frac{2^n - 1}{2^n}}$ at the beginning, which is approximately $\sqrt{\frac{1}{2^n}}$.

We now apply V_f on this quantum state. This will negate the amplitude of our desired $|x_0\rangle$ and not change the amplitude of the rest of the state.

$$V_f |*\rangle = -\frac{1}{\sqrt{2^n}} \underbrace{|x_0\rangle}_{\text{good}} + \sqrt{\frac{2^n - 1}{2^n}} \underbrace{\sum_{x \neq x_0} |x\rangle}_{\text{bad}}$$

This looks like this in the geometric interpretation:

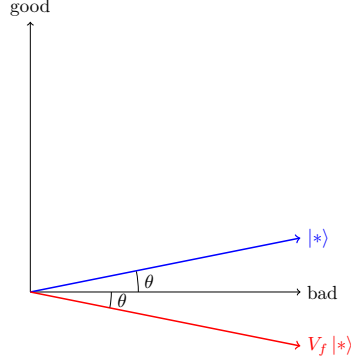


Figure 11.6: Geometric interpretation after V_f

We can see that by applying V_f , we mirror the vector across the *bad* axis.

After V_f , we apply the FLIP_* operation on the quantum state. Since FLIP_* does nothing on the $|*\rangle$ entries and negates the amplitude of any vector orthogonal to it, FLIP_* mirrors the vector across $|*\rangle$. This can be seen in the following figure:

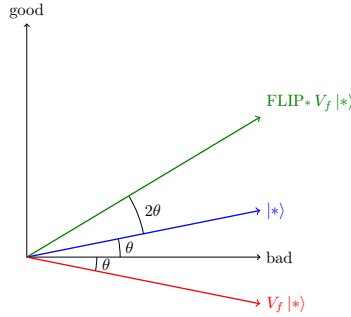


Figure 11.7: Geometric interpretation after FLIP_*

All in all, we have seen that by applying V_f and FLIP_* , we can increase the angle of the quantum state in relation to the “good” and “bad” states by 2θ . Therefore two reflection give rotation. By repeating this step often enough, we can get the amplitude of $|x_0\rangle$ close to 1.

To be more precise: Since we know θ and we know that we will increase the *good*-ness of our quantum state by 2θ each time, we can calculate that only t iterations are necessary with

$$t \approx \frac{\frac{\pi/2}{\theta} - 1}{2} \approx \frac{\pi}{4\theta} \approx \frac{\pi}{4} \cdot \sqrt{2^n}.$$

Grover’s algorithm therefore takes $O(\sqrt{2^n})$ steps, where an evaluation of the circuit counts as one step.