Quantum Computing

Summary
Fabian Damken
July 17, 2022

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Contents

List of Figures

List of Tables

List of Algorithms

1 Preliminaries

In this chapter we discuss the groundwork for the upcoming topics. Along with these subjects, basic knowledge from linear algebra is required.

1.1 Complex Numbers

One of the underlying principles of quantum mechanics (QM) and therefore quantum computing (QC), too, are complex numbers. This section summarizes some results for them *very briefly*.

Let $z = a + ib \in \mathbb{C}$ be a complex number with the real and imaginary components $\mathrm{Re}(z) = a, \mathrm{Im}(z) = b \in \mathbb{R}$. Its magnitude is

$$|z| := \sqrt{a^2 + b^2} = \sqrt{zz^*}$$

with the *complex conjugate* $z^* = a - ib$. The complex conjugate is distributive over addition and multiplication[1], i.e., $(z_1 + z_2)^* = z_1^* + z_2^*$ and $(z_1 z_2)^* = z_1^* z_2^*$ holds for two complex numbers $z_1, z_2 \in \mathbb{C}$. Any complex number can also be written in polar form $z = re^{i\varphi}$ with magnitude

$$|z| = \sqrt{zz^*} = \sqrt{re^{i\varphi}re^{-i\varphi}} = \sqrt{r^2 e^{i\varphi - i\varphi}} = \sqrt{r^2} = |r|.$$

Definition 1 (n-th Root of Unity)**.** We call the special complex number $\omega_n = e^{2\pi i/n}$ the *n-th root of unity.*

Theorem 1 (Power Sum of n-th Roots of Unity)**.** *Let ω_n be the n-th root of unity with $n > 1$. Then $\sum_{k=0}^{n-1} \omega_n^k = 0$.*

Proof.

$$\sum_{k=0}^{n-1} \omega_n^k = \frac{1 - \omega_n^n}{1 - \omega_n} = \frac{1 - e^{2i\pi}}{1 - \omega_n} = \frac{1 - 1}{1 - \omega_n} = \frac{0}{1 - \omega_n} = 0$$

□

1.2 Continued Fraction Expansion

Let $x \in (0, 1)$ be a real number[2]. Then we can express this number as its *continued fraction expansion (CFE)*

$$x = \frac{1}{a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \cdots}}}$$

where $a_0, a_1, \cdots \in \mathbb{N}^+$. The CFE of x is finite iff x is rational. The sums

$$\frac{1}{a_0} \qquad \frac{1}{a_0 + \frac{1}{a_1}} \qquad \frac{1}{a_0 + \frac{1}{a_1 + \frac{1}{a_2}}} \qquad \cdots$$

[1] For other useful properties, see `https://en.wikipedia.org/wiki/Complex_conjugate#Properties`.

[2] Note that the restriction on the interval $(0, 1)$ is purely for convenience as we only have x's between zero and one down the line. It is also possible to extend continued fraction expansions to $\mathbb{R}$.

are called *partial sums*. For calculating $a_0, a_1, \ldots$, let

$$x_0 := \cfrac{1}{a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cdots}}} \qquad x_1 := \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cdots}} \qquad x_2 := \cfrac{1}{a_2 + \cdots} \qquad \cdots$$

then the coefficients are $a_i = [1/x_i]$, where the brackets indicate the integral part, i.e., the part in front of the decimal. If for any j, $x_j = 0$, the CFE terminates and the number is exactly represented.

Example 1. *Let* $x = 11\,490/2^{14} \approx 0.701294$. *Then the CFE is calculated as follows:*

i	x_i	$1/x_i$	a_i
0	0.701 294	1.425 94	1
1	0.425 94	2.347 77	2
2	0.347 77	2.875 44	2
3	0.875 44	1.142 28	1
4	0.142 28	7.028 30	7
5	0.028 30	35.3333	35
6	0.333 33	3	3
7	0		

The final CFE is therefore

$$x = \cfrac{1}{1 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{1 + \cfrac{1}{7 + \cfrac{1}{35 + \frac{1}{3}}}}}}}$$

with the coefficients $(a_0, a_1, a_2, a_3, a_4, a_5, a_6) = (1, 2, 2, 1, 7, 35, 3)$.

2 Postulates of Quantum Mechanics

In this chapter we discuss the postulates of QM and some important protocols and results in QC such as the *no-cloning theorem.* This theorem states that it is impossible to copy a quantum state!

2.1 States

In classical computing, a bit is either 0 or 1. A quantum bit, a *qubit,* however, is more general and has the basis states $|0\rangle$ and $|1\rangle$. The states are formed by basis vectors $|0\rangle = (1,0)^\dagger$ and $|1\rangle = (0,1)^\dagger$. More generally, an arbitrary quantum state $|\psi\rangle$ can be a combination of the basis states, $|\psi\rangle = c_0 |0\rangle + c_1 |1\rangle$, a *superposition* (with complex coefficients $c_0, c_1 \in \mathbb{C}$). However, the state has to be normalized, i.e., $|\langle\psi|\psi\rangle|^2 = 1$. The left part of this inner product is called a *bra* vector representing the conjugate transpose of the right side, the *ket* vector.

The following postulate digests this idea more formally.

Postulate 1 (Quantum State)**.** *Any closed physical system can be associated with a Hilbert space $\mathcal{H}$. The state of the system is completely described by a state vector $|\psi\rangle = \sum_{i=0}^{d-1} c_i |i\rangle$ with $\sum_{i=0}^{d-1} |c_i|^2 = 1$ where $\{|i\rangle\}_{i=0}^{d-1}$ forms a basis of $\mathcal{H}^d$.*

Remark 1. *The basis is not confined to the* computational basis $\{|0\rangle, |1\rangle\}$*, although this basis is often used. It may be any other orthonormal basis of $\mathcal{H}$, see* **??**. *For basis of Hilbert spaces with $d > 2$, see section 2.4.*

Instead of writing out the complex coefficients c_0 and c_1, we can also parameterize an arbitrary superposition with angles $\gamma, \varphi, \theta \in \mathbb{R}$:

$$|\psi\rangle = c_0 |0\rangle + c_1 |1\rangle = e^{i\gamma}\left(\cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle\right).$$

However, as we will see in section 2.3, a global phase such as $e^{i\gamma}$ vanishes in all important calculations as $e^{i\gamma}e^{-i\gamma} = 1$. Hence, we can also parameterize any state with just two angles $\varphi \in (0, 2\pi]$ and $\theta \in (0, \pi]$:

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle.$$

Checking that this state is actually normalized is straightforward:

$$\begin{aligned}
\langle\psi|\psi\rangle &= \left(\cos\frac{\theta}{2}\langle 0| + e^{-i\varphi}\sin\frac{\theta}{2}\langle 1|\right)\left(\cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle\right) \\
&= \cos\frac{\theta}{2}\langle 0| + e^{-i\varphi}\sin\frac{\theta}{2}\langle 1|\Big)\left(\cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle\right) \\
&= \cos^2\frac{\theta}{2}\underbrace{\langle 0|0\rangle}_{=1} + e^{i\varphi}\cos\frac{\theta}{2}\sin\frac{\theta}{2}\underbrace{\langle 0|1\rangle}_{=0} + e^{-i\varphi}\cos\frac{\theta}{2}\sin\frac{\theta}{2}\underbrace{\langle 1|0\rangle}_{=0} + e^{i\varphi}e^{-i\varphi}\sin\frac{\theta}{2}\sin\frac{\theta}{2}\underbrace{\langle 1|1\rangle}_{=1} \\
&= \cos^2\frac{\theta}{2} + \sin^2\frac{\theta}{2} = 1.
\end{aligned}$$

Note that in this case $\langle\psi|\psi\rangle = 1$, so $|\langle\psi|\psi\rangle|^2 = 1$ holds, too, and we can drop the absolute-square. In most of the following discussions where we need explicit parametrization, we confine ourselves to real coefficients, i.e., $\varphi = 0$. This simplifies the discussion as now there is only one parameter θ.

2.2 Evolution

The evolution of quantum states, i.e., how they pass between states, is described by linear transformations U, also called *gates.* These gates transform a quantum state $|\psi\rangle$ into another quantum state $|\psi'\rangle$. In quantum circuits, we denote an application of U_1 and then U_2 to a state $|\psi\rangle$, i.e., $U_2U_1|\psi\rangle$, as:

$$|\psi\rangle \;-\boxed{U_1}-\boxed{U_2}-\; U_2U_1|\psi\rangle$$

Postulate 2 (State Evolution)**.** *The evolution* $|\psi(t_0)\rangle \xrightarrow{U} |\psi(t)\rangle$ *of a closed physical system is described by a unitary transformation* $UU^\dagger = \mathbb{1}$.

Theorem 2 (Unitarity of Quantum Gates)**.** *A linear quantum gate* U *is unitary, i.e.,* $UU^\dagger = \mathbb{1}$.

Proof. □

2.2.1 Gates

In this section we collect the most important single-qubit gates. They are summarized in Table 2.1 and their semantics are given in the caption.

Theorem 3 (Decomposition of Two-By-Two Unitary Matrices)**.** *Every unitary matrix* $U \in \mathbb{C}^{2\times 2}$ *can be decomposed into three rotations as* $U = e^{i\alpha}R_z(\beta)R_y(\gamma)R_z(\delta)$.

From this theorem, one might think that it is necessary to implement every rotation in the lab for a universal quantum computer. Fortunately, this is not the case! As we will see in chapter 4, only three gates are necessary to implement arbitrary rotations.

2.3 Measurement

We will now discuss the last postulate of QM which is concerned with *measurements.* The central result is that measuring a quantum system is inherently *probabilistic,* i.e., the outcome of a measurement is not deterministic and truly random. For any quantum state $|\psi\rangle$, the probability of measuring an outcome v_i is given by the absolute-square of the inner product between the “measurement state” $|v_i\rangle$ and the state $|\psi\rangle$:

$$P(i) = |\langle v_i|\psi\rangle|^2.$$

The value of this inner product (without the absolute-square) is called the *probability amplitude* and can be negative or even complex. Immediately after a measurement, the state $|\psi\rangle$ collapses into a post-measurement state $|\psi'\rangle$. This post-measurement state is

$$|\psi'\rangle = \frac{M_i|\psi\rangle}{N_i}$$

where $M_i = |v_i\rangle\langle v_i|$ and $N_i = \sqrt{P(i)}$ are the *measurement operator* and *normalization constant,* respectively. These results are digested in the following postulate.

Postulate 3 (Quantum Measurement)**.** *Quantum measurements are described by a collection of measurement operators* $\{M_i\}$ *where* i *indicated the outcome of the experiment. Let* $|\psi\rangle$ *be the state before the measurement, then the state immediately after the measurement is* $|\psi'\rangle = M_i|\psi\rangle/N_i$ *where* $N = \sqrt{P(i)}$ *is for normalization.*

U	$U\ket{0}$	$U\ket{1}$	$U\ket{+}$	$U\ket{-}$
$X = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} = HZH$	$\ket{1}$	$\ket{0}$	$\ket{+}$	$-\ket{-}$
$Y = \begin{bmatrix}0 & -i\\ i & 0\end{bmatrix} \equiv XZ$	$i\ket{1}$	$-i\ket{0}$	$-i\ket{-}$	$i\ket{+}$
$Z = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$	$\ket{0}$	$-\ket{1}$	$\ket{-}$	$\ket{+}$
$H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$	$\ket{+}$	$\ket{-}$	$\ket{0}$	$\ket{1}$
$R_y(\gamma) = \begin{bmatrix}c_\gamma & -s_\gamma\\ s_\gamma & c_\gamma\end{bmatrix}$	$c_\gamma\ket{0} + s_\gamma\ket{1}$	$-s_\gamma\ket{0} + c_\gamma\ket{1}$	$c_\gamma\ket{+} - s_\gamma\ket{-}$	$s_\gamma\ket{+} + c_\gamma\ket{-}$
$R_z(\beta) = \begin{bmatrix}e^{i\beta/2} & 0\\ 0 & e^{-i\beta/2}\end{bmatrix}$	$e^{i\beta/2}\ket{0}$	$e^{-i\beta/2}\ket{1}$	$e^{i\beta/2}\ket{0} + e^{-i\beta/2}\ket{1}$	$e^{i\beta/2}\ket{0} - e^{-i\beta/2}\ket{1}$

Table 2.1: Common qubit gates and their effect on the computational and Hadamard basis. For brevity, let $c_\gamma := \cos(\gamma/2)$ and $s_\gamma := \sin(\gamma/2)$. The gates have the following effects in the computational basis: X implements a logical not, Y combines a phase-flip and logical not, Z implements a phase-flip, H creates an equal superposition, $R_y(\gamma)$ rotates around an arbitrary angle γ, and $R_z(\beta)$ adds a phase. In Hadamard basis, the gates have the following effects: X implement a phase-flip, Y combined a phase-flip and logical not, Z implements a logical not, H creates an equal superposition, $R_y(\gamma)$ rotates around an arbitrary angle γ, and $R_z(\beta)$ adds a phase.

Theorem 4 (Measurement of Pure Quantum States)**.** *For* pure *states $\ket{\psi}$, the post-measurement state $\ket{\psi'}$ after a measurement of $\ket{v_i}$ is $\ket{\psi'} = \ket{v_i}$.*

Proof.

$$\frac{M_i\ket{\psi}}{N_i} = \frac{\ket{v_i}\bra{v_i}\ket{\psi}}{\sqrt{P(i)}} = \frac{\ket{v_i}\bra{v_i}\ket{\psi}}{\sqrt{|\langle v_i|\psi\rangle|^2}} = \frac{\ket{v_i}\langle v_i|\psi\rangle}{|\langle v_i|\psi\rangle|} = \ket{v_i}\frac{\langle v_i|\psi\rangle}{|\langle v_i|\psi\rangle|} \equiv \ket{v_i}$$

□

2.4 Composite Systems and Tensor Products

As in classical computing where we are concerned with more than one bit, QC also works with more than one qubit. The formalism for this are *tensor products* $\mathcal{H}^2 \otimes \mathcal{H}^2$ between the Hilbert spaces of the individual qubits. Its basis vectors are also constructed using tensor products:

$$\ket{0}\otimes\ket{0} = \begin{bmatrix}1\\0\\0\\0\end{bmatrix} \qquad \ket{0}\otimes\ket{1} = \begin{bmatrix}0\\1\\0\\0\end{bmatrix} \qquad \ket{1}\otimes\ket{0} = \begin{bmatrix}0\\0\\1\\0\end{bmatrix} \qquad \ket{1}\otimes\ket{1} = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

For two single-qubit operators $A = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix}$ and $B = \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix}$, the tensor product is carried out as

$$A \otimes B = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \otimes \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} a_{00} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} & a_{01} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} \\ a_{10} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} & a_{11} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a_{00}b_{00} & a_{00}b_{01} & a_{01}b_{00} & a_{01}b_{01} \\ a_{00}b_{10} & a_{00}b_{11} & a_{01}b_{10} & a_{01}b_{11} \\ a_{10}b_{00} & a_{10}b_{01} & a_{11}b_{00} & a_{11}b_{01} \\ a_{10}b_{10} & a_{10}b_{11} & a_{11}b_{10} & a_{11}b_{11} \end{bmatrix}$$

with some abuse of notation. This definition has the effect of applying unitary A to the first and unitary B to the second qubit in a tensor-multiplied Hilbert space, i.e.,

$$(A \otimes B)(|\psi\rangle_1 \otimes |\psi\rangle_2) = (A\,|\psi\rangle_1) \otimes (B\,|\psi\rangle_2)$$

For brevity, we often write product state as $|0\rangle \otimes |1\rangle \doteq |0\rangle\,|1\rangle \doteq |01\rangle$ and the application of product operators as $(A \otimes B)(|0\rangle \otimes |1\rangle) \doteq A \otimes B\,|0\rangle \otimes |1\rangle = A_1 B_2\,|01\rangle$. As long as it is clear which unitary is applied to which qubit, a variety of notations may be used. For brevity, we also often write $|\psi\rangle^{\otimes N} \doteq \underbrace{|\psi\rangle \otimes \cdots \otimes |\psi\rangle}_{N \text{ times}}$ and the same for gates.

2.4.1 Entanglement

A *composite* or *product* state is a state $|\psi_{12}\rangle$ that can be written as the product of two individual states $|\psi_1\rangle = \alpha_1\,|0\rangle + \beta_1\,|1\rangle$ and $|\psi_2\rangle = \alpha_2\,|0\rangle + \beta_2\,|1\rangle$:

$$|\psi_{12}\rangle = |\psi_1\rangle \otimes |\psi_2\rangle = (\alpha_1\,|0\rangle + \beta_1\,|1\rangle) \otimes (\alpha_2\,|0\rangle + \beta_2\,|1\rangle) = \alpha_1\alpha_2\,|00\rangle + \alpha_1\beta_2\,|01\rangle + \beta_1\alpha_2\,|10\rangle + \beta_1\beta_2\,|11\rangle$$

However, there are states that cannot be written like this!

Definition 2 (Entangled State). A quantum state $|\psi_{12}\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$ is called *entangled* if there are no states $|\psi_1\rangle \in \mathcal{H}_1$ and $|\psi_2\rangle \in \mathcal{H}_2$ such that $|\psi_{12}\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$.

Theorem 5 (Simple Entangled States). *All states* $|\psi_\theta\rangle = \cos\frac{\theta}{2}\,|00\rangle + \sin\frac{\theta}{2}\,|11\rangle$, $\theta \in (0, \pi/2]$ *are entangled.*

Proof. Let $|\psi_1\rangle := \alpha_1\,|0\rangle + \beta_1\,|1\rangle$ and $|\psi_2\rangle := \alpha_2\,|0\rangle + \beta_2\,|1\rangle$ with coefficients $\alpha_1, \alpha_2, \beta_1, \beta_2 \in \mathbb{C}$. Assume that $|\psi_\theta\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$. Hence,

$$|\psi_\theta\rangle = \alpha_1\alpha_2\,|00\rangle + \alpha_1\beta_2\,|01\rangle + \beta_1\alpha_2\,|10\rangle + \beta_1\beta_2\,|11\rangle \overset{!}{=} \cos\frac{\theta}{2}\,|00\rangle + \sin\frac{\theta}{2}\,|11\rangle$$

By comparing coefficients, all of the following must hold: $\alpha_1\alpha_2 =\neq 0$, $\beta_1\beta_2 =\neq 0$, and $\alpha_1\beta_2 = \beta_1\alpha_2 = 0$. From the first two constraints it follows that all coefficients must be non-zero which contradicts the last constraint. Hence, the state is entangled. □

One important special case of this result is the *Bell state* $|\Phi^+\rangle := \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ which we will study further in chapter 7.

Multipartite

So far, we only studied entanglement of two parties $\mathcal{H}_1$ and $\mathcal{H}_2$. However, it is also possible to describe entanglement between three or more parties. For three parties $\mathcal{H}_1$, $\mathcal{H}_2$, and $\mathcal{H}_3$, there can be a variety of different entanglements:

$$|\psi_{123}\rangle = |\psi_{12}\rangle \otimes |\psi_3\rangle \qquad |\psi_{123}\rangle = |\psi_1\rangle \otimes |\psi_{23}\rangle \qquad |\psi_{123}\rangle = |\psi_2\rangle \otimes |\psi_{13}\rangle$$

For more than two parties, a state $|\psi\rangle_{123}$ that cannot be expressed as a product of its components is called *genuine multipartite entangled (GME)*. To check whether some state is GME can be done explicitly analogous to the above proof of two-party entanglement by checking all the above cases along with

$$|\psi 123\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes |\psi_3\rangle .$$

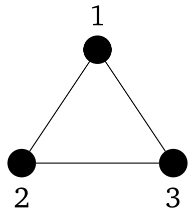
However, for N qubits the (potentially) entangled state has 2^N coefficients! The complexity of this checking is therefore $\mathcal{O}(\text{scary})$. There are, however, less straightforward, but easier-to-check procedures for validating whether a state is GME, but these are out of scope of this course.

Graph States

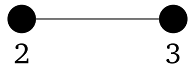
Although general methods for checking GME is out of scope, we will still look at the most famous example: *graph states*. Graph states a multi-qubit states corresponding to the mathematical structure of a graph. Let $G = (V, E)$ be a graph with vertices V and edges E. Then the corresponding multi-qubit state is

$$|G\rangle = \prod_{e \in E} CZ_e \, |+\rangle^{\otimes |V|}$$

where $CZ_e = \text{diag}(1, 1, 1, -1)$ is a controlled-Z-gate (see subsection 2.4.2) acting on the qubits of the edge. These graph states allowed for a new language to reason about quantum states. For instance, when measuring the first qubit of the following graph state in Z-basis,



it just disappears, dropping the connections to the second and third qubit:

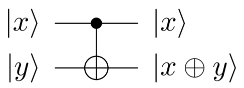


Similar rules exist for other measurements, but these are again out of scope for this course.

2.4.2 Multi-Qubit Gates

So far, we only discussed local gates acting on a single qubit (remember, gates combined with tensor products are applied on each gate individually). While this already allows some calculations, it does not allow interplay of multiple qubits or generation of entangled states which are very important for various protocols (see section 2.5). Hence, we need *multi-qubit gates* U that cannot be written as the product of local gates, i.e., $U \neq U_1 \otimes \cdots \otimes U_N$.

CNOT-Gate The simplest is the CNOT-gate:

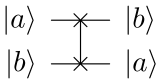


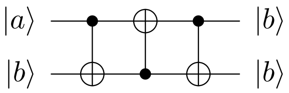
$$CNOT_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Input	Output
$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|01\rangle$
$\|10\rangle$	$\|11\rangle$
$\|11\rangle$	$\|10\rangle$

This gate is a *controlled* gate and applied the X-gate to the second qubit iff the first qubit is 1. The indices $CNOT_{ij}$ indicate that the gate is acting on the j-th qubit (the *target*) and controlled by the i-th qubit. This gate can be extended to more than two qubits (with $n-1$ control qubits and a single target). For $n = 3$, it is called the *Toffoli gate* which can be used to represent classical logical operations like logical not, and, or, and not-and.

SWAP-Gate Another important two-qubit gate is the SWAP-gate





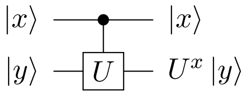
which simply switches the state of two qubits. The circuit on the right is the implementation of the SWAP-gate. Showing their equivalence is straightforward:

$$|00\rangle \xrightarrow{CNOT_{12}} |00\rangle \xrightarrow{CNOT_{21}} |00\rangle \xrightarrow{CNOT_{12}} |00\rangle \qquad |01\rangle \xrightarrow{CNOT_{12}} |01\rangle \xrightarrow{CNOT_{21}} |11\rangle \xrightarrow{CNOT_{12}} |10\rangle$$

$$|10\rangle \xrightarrow{CNOT_{12}} |11\rangle \xrightarrow{CNOT_{21}} |01\rangle \xrightarrow{CNOT_{12}} |01\rangle \qquad |11\rangle \xrightarrow{CNOT_{12}} |10\rangle \xrightarrow{CNOT_{21}} |10\rangle \xrightarrow{CNOT_{12}} |11\rangle$$

As unitary transformations are linear, we almost always only have to show the equivalence for the basis states as every state can be expressed as a superposition of them. This simplifies a lot of derivations! As the above circuit implements swapping for the basis states, it is a valid implementation of the SWAP-gate.

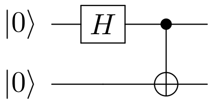
Controlled-U-Gate Note that any gate U can be used in a controlled fashion:



$$CU_{12} = \begin{bmatrix} \mathbb{1} & 0 \\ 0 & U \end{bmatrix}$$

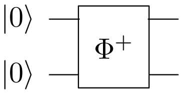
To implement this gate in practice, it can be decomposed into CNOT-gates and single-qubit gates (see **??**).

Preparing the Bell State Equipped with these tools, we can prepare the Bell state $|\Phi^+\rangle$ with the following circuit:



$$|00\rangle \xrightarrow{H_1} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\,|0\rangle \xrightarrow{CNOT_{12}} \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \left|\Phi^+\right\rangle$$

For brevity, we will write



from now on whenever a Bell state is prepared between two qubits. Also, we will leave out the explicit derivation of the Bell state will from derivations.

2.5 Protocols

In this section we discuss some essential protocols in QC and the no-cloning theorem. These protocols are not complete algorithms (which are discussed in chapter 5), but illustrate essential ideas supporting some of the algorithms.

2.5.1 No-Cloning

While the no-cloning theorem is not really a protocol, it is an extremely important result for QC and thus also covered here.

Theorem 6 (No-Cloning)**.** *Let $|\psi\rangle$ be some state. Then there exists no U such that $U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle$. That is, no circuit exists that copies an arbitrary quantum state.*

Proof. Assume that U is a cloning circuit and let $|\psi\rangle$ and $|\phi\rangle$ be arbitrary states. Then we can compute

$$(\langle\phi|\langle\phi|)(|\psi\rangle|\psi\rangle) = \langle\phi|\psi\rangle\langle\psi|\phi\rangle = (\langle\phi|\psi\rangle)^2.$$

However, we can also express the composite states as $|\phi\rangle|\phi\rangle = U|\phi\rangle|0\rangle$ and $|\psi\rangle|\psi\rangle = U|\psi\rangle|0\rangle$ using the definition of the cloning circuit U. Hence,

$$\langle 0|\langle\phi|\underbrace{U^\dagger U}_{=\mathbb{1}}|\psi\rangle|0\rangle = \langle 0|\langle\phi|\psi\rangle|0\rangle = \langle 0|0\rangle\langle\phi|\psi\rangle = \langle\phi|\psi\rangle\,.$$

Therefore, $(\langle\phi|\psi\rangle)^2 = \langle\phi|\psi\rangle$ holds. The states $|\phi\rangle$ and $|\psi\rangle$ are therefore orthogonal, $\langle\phi|\psi\rangle = 0$, or equal, $\langle\phi|\psi\rangle = 1$. This corresponds to classical data (either 0 or 1) and no arbitrary quantum states. Hence, there exists no such U. □

This theorem is a fundamental result of QC and hinders some algorithms down the line. But it is not new! In fact, the no-cloning theorem is *equivalent* to Heisenberg's uncertainty principles stating that for any quantum system there exist two properties which cannot both be measured with certainty. Proofing this equivalence would go as follows (proofing both directions using contraposition):

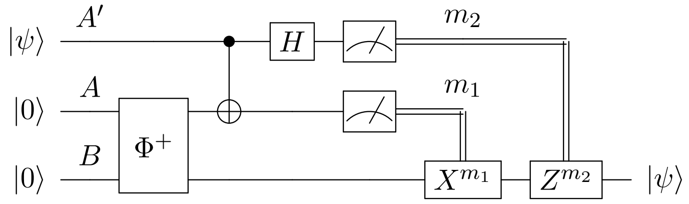
- From no-cloning to Heisenberg: if Heisenberg's uncertainty principle would be false, we could measure everything with certainty and thus prepare a second state simply by transferring the measured data, violating the no-cloning theorem.

- From Heisenberg to no-cloning: if the no-cloning theorem would be false, we could copy an arbitrary quantum state an arbitrary number of times and thus measure the state with arbitrary precision, violating Heisenberg's uncertainty principle.

2.5.2 Quantum Teleportation

With *quantum teleportation,* it is possible to teleport an arbitrary quantum state from one position to another (e.g., from Alice's to Bob's lab) using entanglement. Both parties (Alice and Bob) previously shared a Bell state $|\Phi^+\rangle$ and now Alice wants to transmit her state $|\psi\rangle$ over to Bob, but they cannot meet and have no secure communication channel. However, Alice can publicly announce two classical bits of information that Bob will read.

Consider the following circuit:



(2.1)

Note how the state $|\psi\rangle$ is teleported from qubit A to qubit B. Also note that the state is not cloned as Alice's measurement destroys her copy. To see that the above circuit actually copies the state, we can simply calculate what it does to the circuit. Let $|\psi\rangle = c_0 |0\rangle + c_1 |1\rangle$ be the qubit to be copied. Right before the measurements, the system has the following state:

$$\begin{aligned}
& (c_0 |0\rangle + c_1 |1\rangle)_{A'} |00\rangle_{AB} \\
\xrightarrow{\Phi^+_{AB}} \quad & \frac{1}{\sqrt{2}} (c_0 |0\rangle + c_1 |1\rangle)_{A'} (|00\rangle + |11\rangle)_{AB} \\
\xrightarrow{CNOT_{A'A}} \quad & \frac{1}{\sqrt{2}} \big(c_0 |0\rangle_{A'} (|00\rangle + |11\rangle)_{AB} + c_1 |1\rangle_{A'} (|10\rangle + |01\rangle)_{AB}\big) \\
\xrightarrow{H_{A'}} \quad & \frac{1}{\sqrt{2}} \big(c_0 |+\rangle_{A'} (|00\rangle + |11\rangle)_{AB} + c_1 |-\rangle_{A'} (|10\rangle + |01\rangle)_{AB}\big) \\
= \quad & \frac{1}{2} \big(c_0 (|0\rangle + |1\rangle)_{A'} (|00\rangle + |11\rangle)_{AB} + c_1 (|0\rangle - |1\rangle)_{A'} (|10\rangle + |01\rangle)_{AB}\big) \\
= \quad & \frac{1}{2} \big(|00\rangle_{A'A} (c_0 |0\rangle + c_1 |1\rangle)_B + |01\rangle_{A'A} (c_0 |1\rangle + c_1 |0\rangle)_B \\
& \qquad + |10\rangle_{A'A} (c_0 |0\rangle - c_1 |1\rangle) + |11\rangle_{A'A} (c_0 |1\rangle - c_1 |0\rangle)_B\big)
\end{aligned}$$

When now measuring the first two qubits, the following outcomes and post-measurement states are present, and the corresponding corrections have to be applied to recover $|\psi\rangle$:

m_1	m_2	$\lvert\psi'\rangle_B$	Correction
0	0	$c_0 \lvert0\rangle + c_1 \lvert1\rangle$	$\mathbb{1}$
0	1	$c_0 \lvert1\rangle + c_1 \lvert0\rangle$	X
1	0	$c_0 \lvert0\rangle - c_1 \lvert1\rangle$	Z
1	1	$c_0 \lvert1\rangle - c_1 \lvert0\rangle$	ZX

With $U^1 = U$ and $U^0 = \mathbb{1}$, the corrections can be summarized into $Z^{m_2} X^{m_1}$ which are the last two gates of circuit (2.1).

We therefore teleported a qubit from A to B! Note that this does *not* allow transmission of information faster-than-light as the two classical bits have to be transmitted. Without them, the qubit is worthless as Bob cannot interpret it correctly[1]. It also does not violate the no-cloning theorem as Alice's copy is destroyed during the measurement.

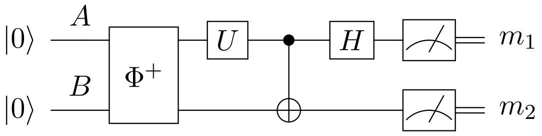
Concatenated Teleportation

2.5.3 Dense-Coding

We are now concerned with the "opposite" problem of qubit teleportation: instead of teleporting a qubit's state, we physically transport it to another location but encode two classical bits of information in it. That

[1] One might argue that Bob might get lucky and read out the correct information. But this kind of "faster-than-light transportation" is also possible classically: you can just guess what the information is—but does not actually transmit information!

is, we transmit two bits of classical information by only transmitting a single qubit. Consider the following circuit:



After creating the Bell state, Alice applies a unitary $U \in \{\mathbb{1}, X, Z, ZX\}$ and subsequently transmits the qubit to Bob. He, now in possession of both qubits A and B, now applies the rest of the gates to read out what unitary Alice applied. The measurement results are as follows:

U	m_1	m_2
$\mathbb{1}$	0	0
X	0	1
Z	1	0
ZX	1	1

Validating this is analogous to the teleportation and left as an exercise to the reader.

Again, this protocol does not allow faster-than-light communication as the qubit has to be physically transmitted. A combination with the teleportation protocol is possible, but this in turns requires the classical transmission of two bits, so still no faster-than-light transmission is possible.

2.6 Why these postulates?

One might ask *why* the postulates are as is (e.g., Why probabilities in the first place? Why amplitudes and not real positive numbers? Why the Euclidean norm and not an arbitrary p-norm? Why linearity?). The hard way to understand this is:

1. learn classical physics
2. learn why classical physics is not sufficient
3. learn quantum physics
4. maybe hear about why amplitudes and not probabilities

However, this course is not the place to squeeze in at least one year worth of lectures just to understand the postulates. Instead, we will take a more pragmatic approach, starting from why we use the Euclidean norm.

Why the Euclidean Norm? Consider $\boldsymbol{v} = (v_1, v_2, \ldots, v_N)$ describing the probabilities of an event with N possible outcomes. We impose a condition $\|\boldsymbol{v}|_p = 1$ to ensure normalization. The most natural choice would be $p = 1$, i.e., requiring that the sum of the magnitudes is unity. However, remember that we want to apply transformations $\boldsymbol{A}$ to the vector and still keep the normalization condition: $\|\boldsymbol{v}\|_p = \|\boldsymbol{A}\boldsymbol{v}\|_p = 1$. For any p, this condition only allows permutations $v_i \mapsto v_j$ and sign flips $v_i \mapsto -v_i$. None of these are capable of encoding everything interesting! However, for $p \in \{1, 2\}$, these matrices can encode more things. For $p = 1$, stochastic matrices are allows and for $p = 2$, we can use unitary matrices! For higher p, no interesting behavior can be encoded. A very practical argument why we use $p = 2$ is therefore that otherwise QM would be very boring.

Why Complex Numbers? Again, we can bring up a very practical argument: only complex number are algebraically closed. Consider, for instance, a unitary gate U. Applying this gate takes t time. If we want to apply it for only $t/2$ time, we need to take its square-root $U = VV = V^2$. With being closed under this operation, it might be that there is no such gate! But as we are able to apply it for only $t/2$ time, there must be some form of square-root-U in the universe. Hence, we have to use complex numbers. Take, for instance, the gate $U = Z = \mathrm{diag}(1, -1) = \big(\mathrm{diag}(1, i)\big)^2$.

Why Linearity? We always have the assumption that gates progress our state linearly. If it would not, i.e., if it would progress nonlinearly, we could solve NP-complete problems! But this is unrealistic, so we confine ourselves to linear evolution…

What is Quantum Mechanics About? Quantum mechanics is not about matter, energy, waves, nor particles. Instead, its solely about information, probabilities, and observables and how these relate to each other! Whenever seeing two linear operators in QM, the sole answer to whether they commute conveys large amounts of information.

3 Computational Complexity

In this chapter we cover the basic ideas of complexity theory. As the core motivation behind QC is to speed up certain tasks, we first have to lay the ground for discussing what “speed up” actually means. In computer science, the *complex* of an algorithm describes the resources required to run it. This resource is often *space* or *time*. That is, how much memory or time it takes to run a specific algorithm. To assign a complexity to a problem instead of a specific algorithm, we assign say that the problem has the complexity of the best algorithm solving it.

The most common complexity classes are depicted in Figure 3.1. These are:

- P: problems that are solvable in polynomial time (graph connectivity, testing if a number is prime, matchmaking, sorting, linear search, . . .)
- Bounded-Error Quantum Polynomial (BQP): problems solvable on a quantum computer with bounded error probability (e.g., $P(\text{error}) \leq 2/3$) (factoring, discrete logarithm, . . . ?)
- NP: problems believed to not be solvable in polynomial time (graph isomorphism, . . .)
- NP-complete: hard problems that can be reduced on each other and for which the solution can be checked in polynomial time (box packing, map coloring, traveling salesman, $n \times n$ Sudoku, . . .)
- PSPACE: problems which need polynomial amount of memory ($n \times n$ chess, $n \times n$ Go, . . .)

These complexity classes are defined such that P $\subseteq$ BQP $\subseteq$ NP $\subseteq$ PSPACE. A big open problem of computer science is whether P $\neq$ NP, i.e., whether we can solve all problems “fast.” A similar question comes up for QC: is P $\neq$ BQP, i.e., are there problems that can actually be solved faster on a quantum computer?

To assess the complexity of a quantum algorithm, we count the gates required to implement the circuit. We will see in the next chapter (chapter 4) how this scales with the problem size.

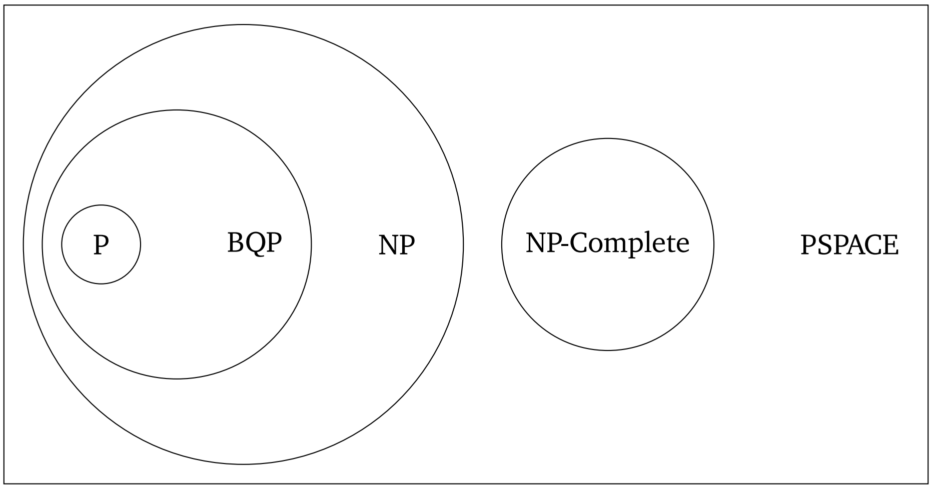
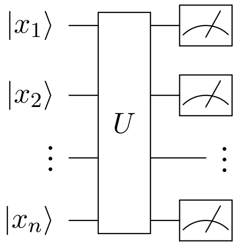


Figure 3.1: The Computational Complexity Zoo

4 Universal Computation

In general, a quantum circuit is just a unitary U and measurements,



with an initialization $|x_1x_2\ldots x_n\rangle$. However, this circuit has to be constructed somehow from gates we have available in the lab. this raises the natural question of what gates we have to implement to build every unitary—and whether there are actually a set of gates fulfilling this.

Definition 3 (Universal Set of Gates)**.** A set of gates $\mathcal{G}$ is called *universal* if any unitary can be approximated with arbitrary accuracy using only gates from this set. With a gate U and its approximation V, let

$$E(U,V) = \max_{|\psi\rangle} \|(U-V)\,|\psi\rangle\|$$

be the *error* between U and V. Note that this error is an upper bound on the probability error $|P_U - P_V| \leq 2E(U,V)$ quantifying the difference in the probability distributions induced by applying U or V to a state.

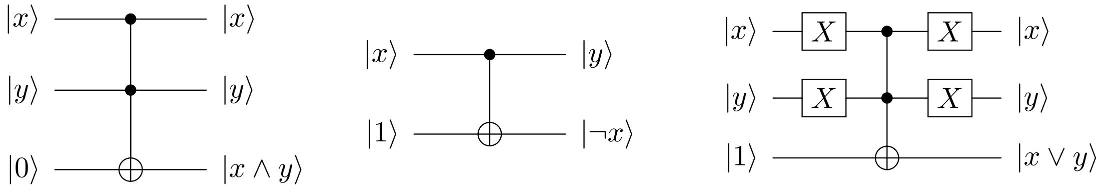
Note that is analogous to classical computing. There we could also decompose every algorithm into a set of universal logical gates. For instance, the sets $\{\,\text{AND},\text{OR},\text{NOT}\,\}$ and $\{\,\text{NAND}\,\}$ are both universal and can represent every possible classical circuit. We formulate this into a theorem:

Theorem 7 (Classical Set of Universal Gates)**.** *The sets $\{\,AND,OR,NOT\,\}$ and $\{\,NAND\,\}$ and universal for all classical logic gates.*

Proof. By induction. □

Theorem 8 (Embedding of Classical Circuits)**.** *Every classical circuit can be embedded in a quantum circuit performing the equivalent operation, but reversibly.*

Proof. To proof this, we use that $\{\,\text{AND},\text{NOT},\text{OR}\,\}$ is universal for all classical gates (Theorem 7). We therefore only have to show that these gates can be resembled using quantum circuits. For this, we use the X-, CNOT-, and Toffoli-gate:



Showing the equivalence is trivial. Note that in the logical or, the Toffoli-gate functions as a not-and due to the target qubit being set to $|1\rangle$. □

4.1 Universal Quantum Gates

In this section we will go over the proof of universality. Some common groups of quantum gates that are discussed are, for instance, the *Pauli group* $\mathcal{P} = \langle X, Z\rangle$ from which all the Pauli-gates can be constructed:

$$\langle X, Z\rangle \longrightarrow \{X^2 = Z^2 = \mathbb{1}, X, XZ = iY, ZX = -iY, Z\} \equiv \{\mathbb{1}, X, Y, Z\}.$$

Another important group is the *Clifford group* $\mathcal{C} = \langle H, S, CNOT\rangle$ with

$$\langle H, S, CNOT\rangle \longrightarrow \{H^2 = \mathbb{1}, S^2 = Z, \dots\}.$$

However, we have the following result:

Theorem 9 (Gottesman-Knill Theorem)**.** *Circuits build using solely gates from the Clifford group can be efficiently simulated on a classical computer.*

Hence, the Clifford group is not enough as we will never see a speedup when just using its gates! However, if we add the T-gate $T = \operatorname{diag}(1, e^{i\pi/4})$, we get a universal set of gates:

Theorem 10 (Universal Set of Quantum Gates)**.** *The following set of quantum gates is universal:* $\langle H, S, CNOT, T\rangle$.

Proof Sketch. The proof of this theorem has three parts:

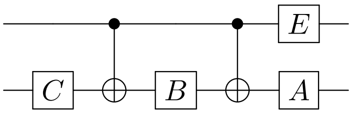
1. every unitary matrix can be decomposed into the product of two-level[1] unitary matrices
2. every two-level unitary matrix can be decomposed into CNOT- and single-qubit gates
3. every single-qubit gate can be approximated with arbitrary accuracy by $\langle H, T\rangle$

which we will cover in greater detail.

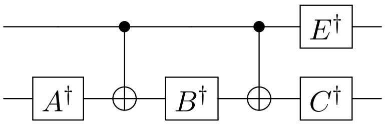
Part 1/3: For an n-qubit gate U, at most $2^{n-1}(2^n - 1) \in \mathcal{O}(4^n)$ two-level unitary matrices are needed.

Part 2/3: Let $\tilde{U} \in \mathbb{C}^{2n+1 \times 2^{n+1}}$ be a two-level unitary matrix acting on $n+1$ qubits. Assume w.l.o.g. that $\tilde{U}$ is a block diagonal matrix $\tilde{U} = \operatorname{diag}(\mathbb{1}, U)$, where $U \in \mathbb{C}^{2\times 2}$ contains the four non-trivial entries of $\tilde{U}$. Hence, $\tilde{U}$ is a the n-controlled version of U, i.e., U is only applied to the $(n+1)$-th qubit iff the first n qubits are 1. To show that this gate can be constructed using just single-qubit gates and CNOT-gates, we first construct a controlled-U-gate, then a controlled-controlled-U-gate, and subsequently expand this to an n-controlled-U-gate.

Let $U =: e^{i\alpha} AXBXC$ be a decomposition of U such that $ABC = \mathbb{1}$ (note that this is always possible due to Theorem 3). The following circuit implements a controlled-U-gate:



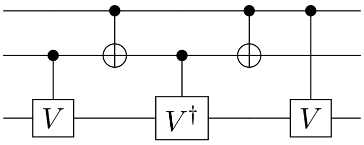
with $E := \operatorname{diag}(1, e^{i\alpha})$. For $|0x\rangle$, both E and the CNOT-gates have no effect and therefore due to $ABC = \mathbb{1}$, the state is left as is. For $|1x\rangle$, both E and the CNOT-gates are applied and therefore the unitary $CXBXA$ acts on the second qubit together with the global phase $e^{i\alpha}$, which is equivalent to applying U. Similarly,



[1] A two-level unitary matrix is a matrix that only acts non-trivially on at most two vector components.

implements the controlled-$U^\dagger$-gate necessary for the next step (the proof is analogous).

The controlled-controlled-U-gate is realized by



(4.1)

with V being the half-U-gate, i.e., $V^2 = U$. For $|00x\rangle$, no gate is ever applied. For $|01x\rangle$, V and $V^\dagger$ are applied, canceling each other out. For $|10x\rangle$, $V^\dagger$ and V are applied, canceling each other out. For $|11x\rangle$, however, both V but not $V^\dagger$ are applied (due to the CNOT-gates canceling the activation on the second qubit), resulting in $VV = V^2 = U$ being applied to the this qubit. Hence, this circuit realizes the controlled-controlled-U-gate. By some clever arrangement, this circuit needs eight single-qubit gates and six CNOT-gates.

To build the n-controlled-U-gate, we add the new control bits to the front and expand the control lines of (4.1) to these qubits, using an $(n-1)$-controlled-gate. Finally, we end up using $\mathcal{O}(n^2)$ CNOT- and single-qubit gates.

Part 3/3: By the *Solovay-Kitaev theorem,* approximating a circuit with m CNOT- and single-qubit up to an accuracy ϵ requires $\mathcal{O}\big(m \log^2(m/\epsilon)\big)$ gates from $\langle H, T\rangle$.

This concludes the proof sketch and we end up with

$$\mathcal{O}\left(4^n 2^n \log^2\left(\frac{4^n 2^n}{\epsilon}\right)\right)$$

gates to approximate an arbitrary n-qubit quantum circuit. □

From this discussion and the final gate count that scales exponentially with the number of qubit, it does not appear clear why anyone should think that BQP $\neq$ P. Even classical circuits need an exponential amount of time on a quantum computer! This is the reason why algorithms that are efficient on a quantum computer are rare and require a large amount of creativity. The next chapter covers nearly all quantum algorithms that we know so far, which only reinforces the argument how much creativity is necessary to invent new ones.

5 Algorithms

5.1 Quantum Parallelism

5.1.1 Interference

5.1.2 The Query Unitary

5.1.3 Deutsch's Approach

5.2 Deutsch-Josza Algorithm

5.3 Bernstein-Vazirani Algorithm

5.4 Simon's Algorithm

5.4.1 Problem

5.4.2 Classical Approach

5.4.3 Quantum Approach

Circuit

Post-Processing

Remarks

5.5 Quantum Fourier Transform

5.5.1 Binary Fraction Expansion

5.5.2 Quantum Circuit

5.5.3 Remarks

5.6 Shor's Algorithm

5.6.1 Period Finding

Using Quantum Fourier Transform

Post-Processing

Maximizing the $P(y)$

Recovering the Period

Remarks

5.6.2 From Period Finding to Factoring

Remarks

5.6.3 Summary

5.7 Grover's Algorithm

5.7.1 Classical Approach

5.7.2 Quantum Approach

Circuit

Illustration

Algebraic Proof

Multiple Solutions

Remarks

6 Quantum Error Correction

So far, we assumed that applying gates works perfectly, i.e., that the state transform exactly as our mathematical model predicts. However, in the real world, (quantum) computers are noisy and by no means perfect. In classical computing, the errors that might occur when sending a message (e.g., through the internet or through time by saving it to disk) are confined to bit-flips. In order to make a computer resilient towards bit-flips, the most simplistic approach is to just copy the data and define *logical bits*

$$0 \mapsto 000 =: 0_L \qquad\qquad 1 \mapsto 111 =: 1_L.$$

This is called *encoding* and the protocol is called the *code* (in this case, *repetition* code). Decoding the message works by a majority vote, i.e.:

$$\begin{array}{cccc} 000 \mapsto 0 & 001 \mapsto 0 & 010 \mapsto 0 & 011 \mapsto 1 \\ 100 \mapsto 0 & 101 \mapsto 1 & 110 \mapsto 1 & 111 \mapsto 1 \end{array}$$

This means that a decoding error happens only if more than one bit is flipped. In other words: the encoding is resilient w.r.t. to one bit-flip. Assume that a bit-flip happens with probability p (see Figure 6.1a). Then the error probability is

$$p_e^{(1)} := P(\text{two or more flipped}) = 3P(\text{exactly two flipped}) + P(\text{all flipped}) = 3p^2(1-p) + p^3 < 3p^2,$$

where $3p^2$ is an upper bound. If we re-apply the encoding (and therefore encode a logical bit into nine physical bits), the new error probability is $p_e^{(2)} = 3(3p^2)^2$. In general, after k applications of the encoding, the error probability is proportional to $(3p)^{2^k}$. Hence, $(3p)^{2^k} \to_{k\to\infty} 0$ for $p < 1/3$ and we can achieve an arbitrarily good encoding for finite k.

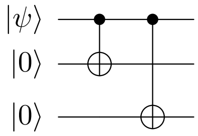
But can we directly transfer this principle to quantum error correction (QEC)? No, because:

- We cannot copy data due to the no-cloning theorem—but would it even help as measurements destroy everything anyway?
- We do not have discrete error but a continuum—do we need infinite precision to locate the error?
- Measuring destroys the message—how to do decode?

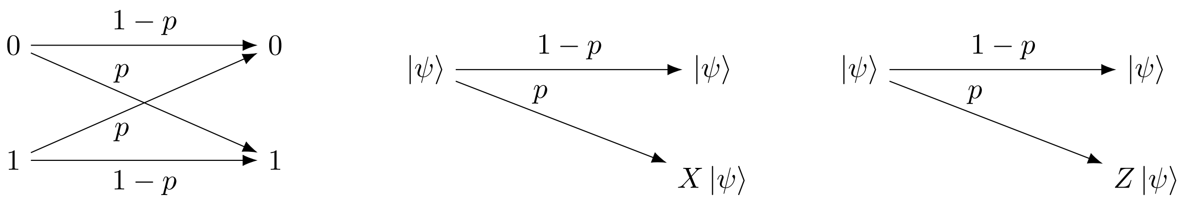
Fortunately though, all of these problems can be circumvented and QEC is possible. We will now cover two kinds of errors: bit-flips and phase-flips and will subsequently see that correcting these suffices to correct all possible errors, effectively reducing the continuum of errors to a finite set.

6.1 Tackling Bit-Flips

We start by fixing bit-flips. To encode the message, we copy the coefficients of an arbitrary state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ to two encoding qubits:



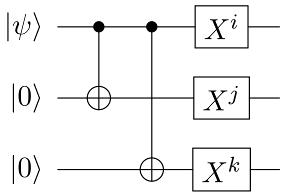
$$= \alpha|000\rangle + \beta|000\rangle =: \alpha|0\rangle_L + \beta|1\rangle_L$$



(a) Classical Bit-Flip (b) Quantum Phase-Flip (c) Quantum Bit-Flip

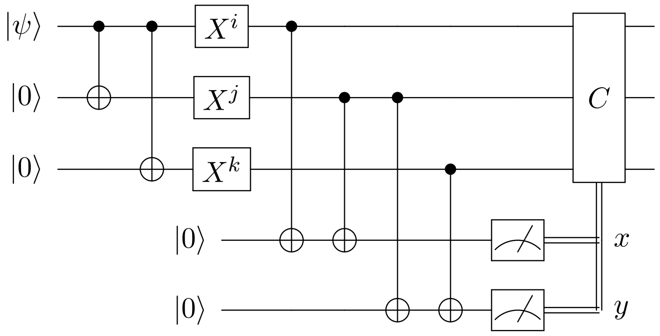
Figure 6.1: State diagram for classical/quantum bit/phase-flip channels while sending a message. A bit/phase-flip happens with probability p and the message is left untouched with probability $1-p$.

Note that we did not actually copy the state, but only the *coefficients* which does not violate the no-cloning theorem! We now assume that an error only happens on exactly one qubit (because otherwise we are not able to recover anyway). We encode this as follows:



i	j	k	State
0	0	0	$\alpha\,\|000\rangle + \beta\,\|111\rangle$
1	0	0	$\alpha\,\|100\rangle + \beta\,\|011\rangle$
0	1	0	$\alpha\,\|010\rangle + \beta\,\|101\rangle$
0	0	1	$\alpha\,\|001\rangle + \beta\,\|110\rangle$

Fortunately, all possible result states are orthogonal! Hence, we can uniquely identify my measuring even though they are all in superposition. However, if we would simply measure, we would destroy the state. We therefore copy the state again into two *helper qubits* which we can then measure:



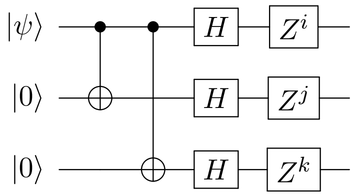
x	y	Correction C
0	0	$\mathbb{1}$
0	1	X_3
1	0	X_1
1	1	X_2

The interpretation of this is as follows: for $x = 1$, we conclude that the first and second qubit have different values. Similarly, $y = 1$ encodes that the second and third qubit have different values. If they have the same value, the CNOT-gates would either be not applied or cancel each other out. Note again that the sole reason why this works is that the state are orthogonal! From this discussion, we can deduce the above corrections.

This approach is also called a *parity check* of Z_1Z_2 and Z_2Z_3, i.e., a parity check in Z-basis.

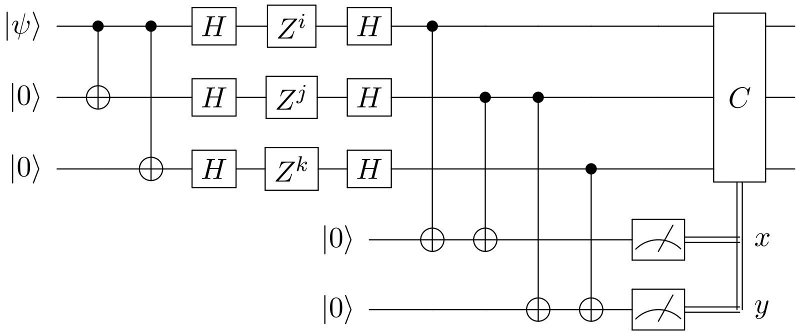
6.2 Tackling Phase-Flips

Correcting phase-flips (see Figure 6.1c) is analogous to correcting bit-flips, but in Hadamard basis. The encoding circuit including the error model along with the resulting states therefore is:



i	j	k	State
0	0	0	$\alpha\,\lvert +++\rangle + \beta\,\lvert ---\rangle$
1	0	0	$\alpha\,\lvert -++\rangle + \beta\,\lvert +--\rangle$
0	1	0	$\alpha\,\lvert +-+\rangle + \beta\,\lvert -+-\rangle$
0	0	1	$\alpha\,\lvert ++-\rangle + \beta\,\lvert --+\rangle$

By applying Hadamard again in the decoding circuit, the identity $HZH = X$ directly tells us that we need to apply the same corrections as for bit-flips:



x	y	Correction C
0	0	$\mathbb{1}$
0	1	X_3
1	0	X_1
1	1	X_2

This approach is a parity check of X_1X_2 and X_2X_3, i.e., a parity check in X-basis.

6.3 Shor's Code

The idea of *Shor's code* is to concatenate the circuits discussed before, using the bit-flip correction three times for each physical qubit of the phase-flip parity check. The whole circuit is shown in Figure 6.2. One can easily verify that Shor's code used the following mapping for logical qubits:

$$\begin{aligned} |0\rangle &\mapsto \frac{|000\rangle + |111\rangle}{\sqrt{2}} \otimes \frac{|000\rangle + |111\rangle}{\sqrt{2}} \otimes \frac{|000\rangle + |111\rangle}{\sqrt{2}} =: |0\rangle_L \\ |1\rangle &\mapsto \frac{|000\rangle - |111\rangle}{\sqrt{2}} \otimes \frac{|000\rangle - |111\rangle}{\sqrt{2}} \otimes \frac{|000\rangle - |111\rangle}{\sqrt{2}} =: |1\rangle_L \end{aligned} \tag{6.1}$$

An important feature of Shor's code is that it is not only capable of correcting X or Z errors, but also XZ errors, i.e., simultaneous bit- and phase-flips. With this feature, it is possible to correct any possible error (see subsection 6.3.1). However, Shor's code is not optimal as it uses so many qubits. It is also possible to use only seven or even five qubits, see section 6.4.

6.3.1 Universal Error Correction

As already discussed is the Shor code capable of correcting XZ-type errors acting on a single qubit. From this we can conclude that the Shor code is in fact capable of correcting arbitrary errors, even non-unitary ones, as long as they act on a single qubit and are valid quantum operations.

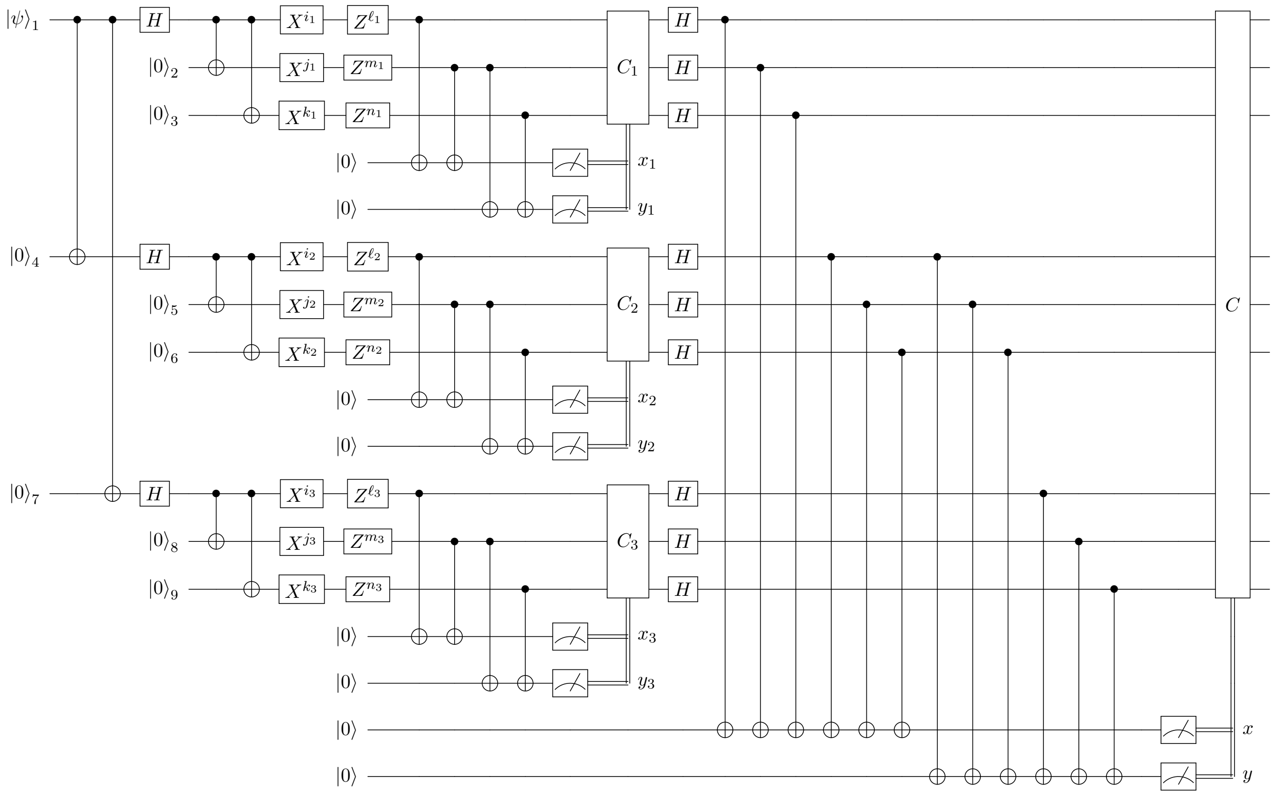


Figure 6.2: Quantum circuit for encoding in Shor's code (the left part until the error gates X and Z) and for decoding (everything to the right of the error gates). The matrices C_1, C_2, C_3, and C denote the corresponding correction matrices for the bit flip ("inner" code) and the phase flip ("outer" code). The "error indices" are mutually exclusive per index or all zero, i.e., $(i_a, j_a, k_a), (\ell_b, m_b, n_b) \in \big\{(0,0,0), (1,0,0), (0,1,0), (0,0,1)\big\}$ for $a, b \in \{1,2,3\}$.

Theorem 11 (Universal Error Correct)**.** *The Shor code is capable of correcting arbitrary errors.*

Proof. W.l.o.g, assume that the error happens on the first qubit and is described by $E_1 \in \mathbb{C}^{2\times 2}$. We can express this matrix using the three Pauli matrices:

$$E = \alpha_0 \mathbb{1}_1 + \alpha_1 X_1 + \alpha_2 Z_1 + \alpha_3 X_1 Z_1$$

This is easy to verify by calculating the right-hand-side, equating it to E_1, and explicitly solving the resulting linear system of equations:

$$\begin{bmatrix} e_{11} & e_{21} \\ e_{21} & e_{22} \end{bmatrix} = \begin{bmatrix} \alpha_0 + \alpha_2 & \alpha_1 - \alpha_3 \\ \alpha_1 + \alpha_3 & \alpha_0 - \alpha_2 \end{bmatrix} \qquad \implies \qquad \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} e_{11} + e_{22} \\ e_{12} + e_{21} \\ e_{11} - e_{22} \\ e_{21} - e_{12} \end{bmatrix}$$

Let $|\psi\rangle_L = \alpha\,|0\rangle_L + \beta\,|1\rangle_L$ be a logical qubit encoded with Shor's code (see (6.1) for the logical basis states). Then, by linearity, an application of the error operator E_1 yields

$$E_1\,|\psi\rangle_L = \alpha_0\,|\psi\rangle_L + \alpha_1 X_1\,|\psi\rangle_L + \alpha_2 Z_1\,|\psi\rangle_L + \alpha_3 X_1 Z_1\,|\psi\rangle_L\,.$$

By dropping the last six qubits from the basis states (as the error only acts on the first qubit), each individual error has the following effect on $|\psi\rangle_K$:

$$\begin{aligned}
\mathbb{1}_1\,|\psi\rangle_L &= \frac{\alpha}{\sqrt{2}}(|000\rangle + |111\rangle) \otimes |\cdots\rangle + \frac{\beta}{\sqrt{2}}(|000\rangle - |111\rangle) \otimes |\cdots\rangle \\
X_1\,|\psi\rangle_L &= \frac{\alpha}{\sqrt{2}}(|100\rangle + |011\rangle) \otimes |\cdots\rangle + \frac{\beta}{\sqrt{2}}(|100\rangle - |011\rangle) \otimes |\cdots\rangle \\
Z_1\,|\psi\rangle_L &= \frac{\alpha}{\sqrt{2}}(|000\rangle - |111\rangle) \otimes |\cdots\rangle + \frac{\beta}{\sqrt{2}}(|000\rangle + |111\rangle) \otimes |\cdots\rangle \\
X_1 Z_1\,|\psi\rangle_L &= \frac{\alpha}{\sqrt{2}}(|100\rangle - |011\rangle) \otimes |\cdots\rangle + \frac{\beta}{\sqrt{2}}(|100\rangle + |011\rangle) \otimes |\cdots\rangle
\end{aligned}$$

As these states are mutually orthogonal, measuring the parity qubits uniquely identifies one of the four errors and the state collapses onto either one of them. Subsequently, Shor's code can be used to correct the individual errors. Hence, Shor's code can be used to correct arbitrary errors. □

6.4 Other Codes

As already mentioned, the Shor code is not *optimal* in the sense of saving qubits. There exist several smaller ones, for instance the *Steane code*

$$|0\rangle_L = \frac{1}{\sqrt{8}}(|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle)$$

$$|1\rangle_L = \frac{1}{\sqrt{8}}(|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle + |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle)$$

or five-qubit codes. However, the Steane code has some nice properties which we will discuss in the next section.

6.5 Fault-Tolerance and Transversality

In fault-tolerant QC, the idea is to perform all operations in encoded states (i.e., logical qubits), without ever decoding. One code supporting this is the Steane code we saw in the previous section. It allows us to execute certain gates *transversely.* That is, we can design a gate that has the same effect on the individual physical qubits as on the logical qubits. For the Steane code, the basic gates are simply tensor-products of single-qubit gates:

$$X_L = X_1 \otimes X_2 \otimes \cdots \otimes X_7 \qquad Z_L = Z_1 \otimes Z_2 \otimes \cdots \otimes Z_7 \qquad H_L = H_1 \otimes H_2 \otimes \cdots \otimes H_7$$

While they acting on the individual qubits is trivial, one can also show that:

$$\begin{aligned} X_L \left|0\right\rangle_L &= \left|1\right\rangle_L & X_L \left|1\right\rangle_L &= \left|0\right\rangle_L \\ Z_L \left|0\right\rangle_L &= \left|0\right\rangle_L & Z_L \left|1\right\rangle_L &= -\left|1\right\rangle_L \\ H_L \left|0\right\rangle_L &= \frac{1}{\sqrt{2}}(\left|0\right\rangle_L + \left|1\right\rangle_L) = \left|+\right\rangle_L & H_L \left|1\right\rangle_L &= \frac{1}{\sqrt{2}}(\left|0\right\rangle_L - \left|1\right\rangle_K) = \left|-\right\rangle_L \end{aligned}$$

An important property to consider in fault-tolerant QC is how errors propagate. For the X-gate, for instance, an error does not propagate to other physical qubits as the gate is local and can be corrected with the Steane code. Application of a CNOT-gate withing a Steane block, however, may propagate an error onto a second qubit which can not be corrected! Instead, a logical CNOT-gate

$$CNOT_L \left|x\right\rangle_L \left|y\right\rangle_L \mapsto \left|x\right\rangle_L \left|x \oplus y\right\rangle_L = CNOT_{1,8} \otimes CNOT_{2,9} \otimes \cdots \otimes CNOT_{7,14}$$

has to be used that only works across two blocks (see Figure 6.3 for the circuit). The T-gate, however, cannot be implemented transversely. Its fault-tolerant implementation is more complicated and out of scope for this course.

The general scheme of performing fault-tolerant QC is:

1. prepare quantum state in logical qubit
2. apply fault-tolerant gate
3. correct errors
4. go to second step until the algorithm is finished

As we already saw, the second step is the fundamental challenge as it is hard to come up with fault-tolerant gates, which we define as follows:

Definition 4 (Fault-Tolerance)**.** A procedure is called *fault-tolerant* if the failure of one component causes at most one error in each encoded block at the output step.

6.6 Threshold Theorem

So far, we assumed that the QEC itself works perfectly. But what if the error correction itself causes errors again? Like for classical repetition codes, we can simply re-encode the bits, building up a layered coding architecture. Let m be the number of places where an error can occur, then the probability of an error using a single coding layer is upper-bounded by $p_e^{(1)} = (mp)^2$. For k layers, the error probability is $p_e^{(k)} = (mp)^{2^k}$ which, analogous to classical error correction, converges to zero for $k \to \infty$ if $p < 1/m$. Hence, we can achieve any error probability ϵ with a large enough k.

This brings us to the *threshold theorem:*

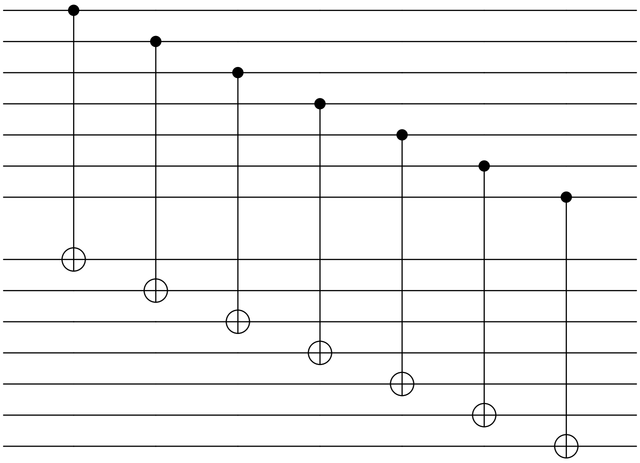


Figure 6.3: Fault-Tolerant CNOT-Gate

Theorem 12 (Threshold Theorem)**.** *A quantum circuit with $p(n)$ gates on n qubits may be simulated with an error probability of at most $\epsilon > 0$ using*[1]

$$\mathcal{O}\left(p(n)\,\log^c \frac{p(n)}{\epsilon}\right)$$

gates (for some constant c when the gates fail with probability of at most p given that $p < p_{\text{th}}$ for some code-dependent constant threshold p_{th}.

For the Steane code, $p_{\text{th}} \approx 10^{-5}$. Hence, to factor numbers with around 2000, we would needs millions of millions of qubits! This is the sole reason why we do not have quantum computing at the moment!

[1] Note that in the lecture, this bound is written as $\mathcal{O}\big(p(n)\,\text{poly}(\log(p(n)/\epsilon))\big)$.

7 Quantum Nonlocality

7.1 Elements of Reality

7.2 CHSH Inequality

7.3 Quantum Violation of the CHSH Inequality

7.4 Tsirelson’s Bound and Quantum Key Distribution

8 Measurement-Based Quantum Computing

8.1 Identity

8.2 Arbitrary Rotations

8.3 CNOT

8.4 Cluster States

8.5 Handling Errors

8.6 Important Gates