Exercise solutions for
Quantum Computation and Quantum Information
by Michael A. Nielsen & Isaac L. Chuang

D. G. O.

Preface

The solutions are original, except where there is a citation. The quantum circuits were made using Quantikz[4]. The solutions here might be incorrect, use it at your own risk. If you find any error, please contact me.

1 Introduction and overview

1.4 Quantum algorithms

1.4.4 The Deutsch – Jozsa algorithm

Exercise 1.1 Let n be the number of bits, m the number of samples Alice asks from Bob. Let p be the probability, that Bob will create a constant function. After m samples, Alice will determine, that the function is constant with probability q, that the function is constant, and she will determine, that the function is balanced with probability $(1-q)$ if all reply bits are the same. If she sees the reply bits are not the same, she will determine that the function is balanced.

In this case, the probability, that Alice will get the correct answer is

$$1-\epsilon = pP_{\mathrm{c}} + (1-p)P_{\mathrm{b}} \ , \tag{1}$$

where P_{c} is the probability, that Alice gets the right answer if the function is constant, and P_{b} is the probability, that Alice gets the right answer if the function is balanced.

If the function is constant, Alice will always gets the same bits, and so she will say the function is constant with q probability: $P_{\mathrm{c}} = q$.

If the function is balanced, the probability that the reply bits will be all zeroes or all ones for m randomly choosen function arguments from any random balanced function is

$$r = \frac{2\binom{n-m}{n/2}}{\binom{n}{n/2}} \ , \tag{2}$$

because there $\binom{n-m}{n/2}$ possible functions where all the replies are zero, and the same number of functions where all replies are one from all the possible $\binom{n}{n/2}$ balanced functions.

If all the reply bits are the same for a balanced functions, Alice determines, that it is a balanced function with a probability $(1-q)$, and this case happens with a probability $(1-p)r$. If the reply bits are not the same, she will respond correctly with a probability 1, and this case happens with probability $(1-p)(1-r)$. Which means, that

$$(1-p)P_{\mathrm{b}} = (1-p)(r(1-q)+(1-r)) = (1-p)(1-rq) \ . \tag{3}$$

All in all

$$1-\epsilon = pq + (1-p)(1-rq) = q(p-r+rp) + (1-p) \ . \tag{4}$$

Assuming $(p-r+rp)$ is positive, the best strategy for Alice is to choose $q=1$, which means

$$1-\epsilon = 1 - 2\frac{\binom{n-m}{n/2}}{\binom{n}{n/2}}(1-p) \ . \tag{5}$$

For $m=2$

$$1-\epsilon = 1-2\frac{(n-2)!(n/2)!}{(n/2-2)!n!}(1-p) = 1-2\frac{(n/2-1)\ n/2}{(n-1)n}(1-p) = 1-\frac{n/2-1}{n-1}(1-p) \ . \tag{6}$$

1.6 Quantum Information

1.6.1 Quantum information theory: example problems

Exercise 1.2 The cloning can be achieved by first identifying the state, and applying a measurement on any other quantum state. In case we want to clone the state $|\psi\rangle$, we do a measurement on that other quantum state with a measuring device which has one eigenvalue which is ψ. If the measurement result is ψ, then we achieved the goal. If it is not, then we apply an other kind of measurement (using the other coordinate frame), which has one eigenvalue which is φ. Then we apply on this state the first measurement, and see if it is ψ now. If not, we repeat the process until we get ψ. Similarly if we want to clone the state $|\varphi\rangle$.

For identifying the state, first, we clone the state in question. Then, we do measurement on that state using the measuring device which has one eigenvalue which is ψ. If we do not get ψ, we know that the original state is $|\varphi\rangle$. If we get ψ, we cannot be certain. We clone the original state again, and do a measurement using the other measuring device which has one eigenvalue which is φ. If it is not φ, then the original state was ψ. We repeat the process of cloning and using measuring devices until we get ψ with the first device, and φ with the second device. This method works because $|\psi\rangle$ has components other than $|\varphi\rangle$ if we write it out in the orthogonal system which $|\varphi\rangle$ is part of, and vice-versa.

2 Introduction to quantum mechanics

2.1 Linear algebra

2.1.1 Basis and linear independence

Exercise 2.1 $1 \cdot (1,-1) + 1 \cdot (1,2) + (-1) \cdot (2,1) = (0,0) = 0.$

2.1.2 Linear operators and matrices

Exercise 2.2 Let's say, that

$$A|n\rangle = \sum_m A_{n,m}|m\rangle. \tag{7}$$

For the original basis states $A_{0,0} = 0$, $A_{0,1} = 1$, $A_{1,0} = 1$, $A_{1,1} = 0$,

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{8}$$

If the basis states are $|a\rangle = 2^{-1/2}(|0\rangle + |1\rangle)$, $|b\rangle = 2^{-1/2}(|0\rangle - |1\rangle)$, then

$$A|a\rangle = A\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) = \frac{|1\rangle + |0\rangle}{\sqrt{2}} = |a\rangle, \tag{9}$$

$$A|b\rangle = A\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = \frac{|1\rangle - |0\rangle}{\sqrt{2}} = -|b\rangle. \tag{10}$$

This means, that in this basis $A_{a,a} = 1$, $A_{a,b} = 0$, $A_{b,a} = 0$, $A_{b,b} = -1$,

$$A = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{11}$$

Exercise 2.3 In this case we want to prove, that

$$(AB)_{i,k} = \sum_j A_{i,j} B_{j,k}. \tag{12}$$

We know, that $(AB)|v_i\rangle = \sum_k (AB)_{i,k}|x_k\rangle$. We also know, that

$$\begin{aligned}(AB)|v_i\rangle = B(A|v_i\rangle) = B\left(\sum_j A_{i,j}|w_j\rangle\right) = \sum_j A_{i,j}(B|w_j\rangle) = \\ = \sum_j A_{i,j}\left(\sum_k B_{j,k}|x_k\rangle\right) = \sum_k \underbrace{\left(\sum_j A_{i,j}B_{j,k}\right)}_{(AB)_{i,k}} |x_k\rangle.\end{aligned} \tag{13}$$

From here we can easily recognize what is $(AB)_{i,k}$.

Exercise 2.4 We know, that for for any state $I|v_i\rangle = |v_i\rangle$. This means, that all the coefficients $I_{i,j}$ for $I|v_i\rangle = \sum_j I_{i,j}|v_j\rangle$ are 0, except for $I_{i,i}$, when it is 1.

2.1.4 Inner products

Exercise 2.5 (1) We have to prove the linearity in the second argument,

$$\begin{aligned}\left(|v\rangle, \sum_i \lambda_i |w_i\rangle\right) = \sum_j v_j^* \left(\sum_i \lambda_i |w_i\rangle\right)_j = \\ = \sum_j v_j^* \sum_i \lambda_i (w_i)_j = \sum_i \lambda_i \sum_j v_j^* (w_i)_j = \end{aligned} \tag{14}$$

$$= \sum_i \lambda_i (|v\rangle, |w_i\rangle). \tag{15}$$

(2) We have to prove, that $(|v\rangle, |w\rangle = (|w\rangle, |v\rangle)^*$.

$$(|v\rangle, |w\rangle) = \sum_i v_i^* w_= \sum_i (w_i^* v_i)^* i = \left(\sum_i w_i^* v_i\right)^* = (|w\rangle, |v\rangle)^*. \tag{16}$$

(3) We have to prove, that $(|v\rangle, |v\rangle) > 0$, if $|v\rangle \neq 0$.

$$(|v\rangle, |v\rangle) = \sum_i v_i^* v_i = \sum_i |v_i|^2, \tag{17}$$

and since all $|v_i|^2$ are nonnegative, and in this case there is at least one component which is nonzero, this sum will be greater than zero.
If $|v\rangle = 0$, then all components of it are zero, and summing $0 \cdot 0$ will be 0, in this case $(|v\rangle, |v\rangle) = 0$.

Exercise 2.6 To do that, I'm going to use previous exercise's results.

$$\left(\sum_i \lambda_i |w_i\rangle, |v\rangle\right) = \left(|v\rangle, \sum_i \lambda_i |w_i\rangle\right)^* = \tag{18}$$

$$= \left(\sum_i \lambda_i (|v\rangle, |w_i\rangle)\right)^* = \sum_i \lambda_i^* (|v\rangle, |w_i\rangle)^* = \sum_i \lambda_i^* (|w_i\rangle, |v\rangle). \tag{19}$$

Exercise 2.7 $(1,1)\cdot(1,-1) = 1\cdot 1 + 1\cdot(-1) = 0$. The normalized forms are $2^{-1/2}(1,1)$ and $2^{-1/2}(1,-1)$.

Exercise 2.8 The book fails to mention, but there is an other condition for this process to work: the basis vectors has to be linearly independent.

First, let's prove, that if the vectors $|w_1\rangle, \ldots, |w_d\rangle$ are linearly independent, then any subset of these are linearly independent. If the original vectors are linearly independent, there are no linear combination of these that will produce 0. This is true even if any subset of the coefficients are 0, which is exactly the same as the linear combination of the remaining subset of vectors. Because we can freely choose which coefficients are 0, we can see, that any subset of linear combination of the vectors will be nonzero.

It follows, that (2.17) in the book will never produce zero in the numerator and denumerator, because the numerator is

$$|w_{k+1}\rangle - \sum_{i=1}^{k} \langle v_i | w_{k+1}\rangle |v_i\rangle, \tag{20}$$

and $|v_i\rangle$ is the linear combination of $|w_1\rangle, \ldots, |w_k\rangle$, which means, that this whole expression is the linear combination of $|w_1\rangle, \ldots, |w_{k+1}\rangle$, which can never produce zero because of linear independency.

Now we can prove, the orthonormality. Any $|v_i\rangle$ is obviously normalized. Any two $|v_{k+1}\rangle$, $|v_j\rangle$ are orthogonal, because let's suppose, that $k+1 > j$, then – without writing out the denominators –,

$$\langle v_j | v_{k+1}\rangle \propto \langle v_j | w_{k+1}\rangle - \sum_{i=1}^{k} \langle v_i | w_{k+1}\rangle \langle v_j | v_i\rangle. \tag{21}$$

Let's use induction proof, and assume that orthonormality holds for the vectors $|v_i\rangle$ $i < k+1$. Then, this previous equation becomes

$$\langle v_j | v_{k+1}\rangle \propto \langle v_j | w_{k+1}\rangle - \sum_{i=1}^{k} \langle v_i | w_{k+1}\rangle \delta_{j,i} = \langle v_j | w_{k+1}\rangle - \langle v_j | w_{k+1}\rangle = 0. \tag{22}$$

Exercise 2.9 The Pauli operators expressed in this form are

$$\begin{gathered}\langle 0|\sigma_0|0\rangle = 1, \langle 0|\sigma_0|1\rangle = 0, \langle 1|\sigma_0|0\rangle = 0, \langle 1|\sigma_0|1\rangle = 1,\\ \langle 0|\sigma_1|0\rangle = 0, \langle 0|\sigma_1|1\rangle = 1, \langle 1|\sigma_1|0\rangle = 1, \langle 1|\sigma_1|1\rangle = 0,\\ \langle 0|\sigma_2|0\rangle = 0, \langle 0|\sigma_2|1\rangle = -i, \langle 1|\sigma_2|0\rangle = i, \langle 1|\sigma_2|1\rangle = 0,\\ \langle 0|\sigma_3|0\rangle = 1, \langle 0|\sigma_3|1\rangle = 0, \langle 1|\sigma_3|0\rangle = 0, \langle 0|\sigma_3|1\rangle = -1.\end{gathered}$$

Exercise 2.10 To get the matrix representation, we have to find $(|v_j\rangle\langle v_k|)_{n,m}$, which is

$$(|v_j\rangle\langle v_k|)_{n,m} = \langle v_n|(|v_j\rangle\langle v_k|)|v_m\rangle = \langle v_n|v_j\rangle\langle v_k|v_m\rangle = \delta_{n,j}\ \delta_{k,m}. \tag{23}$$

2.1.5 Eigenvectors and eigenvalues

Exercise 2.11 For σ_1, the solutions are

$$\begin{gathered}|\sigma_1 - \lambda I| = \begin{vmatrix}-\lambda & +1\\ +1 & -\lambda\end{vmatrix} = \lambda^2 - 1 \stackrel{!}{=} 0 \Rightarrow \lambda_+ = +1, \lambda_- = -1,\\ \begin{bmatrix}-1 & +1\\ +1 & -1\end{bmatrix}|x_+\rangle \stackrel{!}{=} 0 \Rightarrow |x_+\rangle = \begin{bmatrix}+1\\ -1\end{bmatrix},\\ \begin{bmatrix}+1 & +1\\ +1 & +1\end{bmatrix}|x_-\rangle \stackrel{!}{=} 0 \Rightarrow |x_-\rangle = \begin{bmatrix}+1\\ +1\end{bmatrix}.\end{gathered} \tag{24}$$

For σ_2, the solutions are

$$\begin{gathered}|\sigma_2 - \lambda I| = \begin{vmatrix}-\lambda & -i\\ +i & -\lambda\end{vmatrix} = \lambda^2 - 1 \stackrel{!}{=} 0 \Rightarrow \lambda_+ = +1, \lambda_- = -1,\\ \begin{bmatrix}-1 & -i\\ +i & -1\end{bmatrix}|x_+\rangle \stackrel{!}{=} 0 \Rightarrow |x_+\rangle = \begin{bmatrix}+1\\ +i\end{bmatrix},\\ \begin{bmatrix}+1 & -i\\ +i & +1\end{bmatrix}|x_-\rangle \stackrel{!}{=} 0 \Rightarrow |x_-\rangle = \begin{bmatrix}+1\\ -i\end{bmatrix}.\end{gathered} \tag{25}$$

For σ_3, the solutions are

$$\begin{gathered}|\sigma_3 - \lambda I| = \begin{vmatrix}1-\lambda & 0\\ 0 & -1-\lambda\end{vmatrix} = \lambda^2 - 1 \stackrel{!}{=} 0 \Rightarrow \lambda_+ = +1, \lambda_- = -1,\\ \begin{bmatrix}0 & 0\\ 0 & -2\end{bmatrix}|x_+\rangle \stackrel{!}{=} 0 \Rightarrow |x_+\rangle = \begin{bmatrix}+1\\ 0\end{bmatrix},\\ \begin{bmatrix}2 & 0\\ 0 & 0\end{bmatrix}|x_-\rangle \stackrel{!}{=} 0 \Rightarrow |x_-\rangle = \begin{bmatrix}0\\ +1\end{bmatrix}.\end{gathered} \tag{26}$$

Because each matrix has the same eigenvalues, the diagonalization for all of them are $|x_+\rangle\langle x_+| - |x_-\rangle\langle x_-|$, which is just $\mathrm{diag}(1, -1)$ if the basis states are the eigenvectors.

Exercise 2.12 The matrix has only one eigenvalue, which is 1,

$$|\sigma_3 - \lambda I| = \begin{vmatrix} 1-\lambda & 0 \\ 1 & 1-\lambda \end{vmatrix} = (\lambda - 1)^2 \stackrel{!}{=} 0 \Rightarrow \lambda = 1. \tag{27}$$

Let's say, that the operator had the original matrix form in the coordinate system where the basis vectors are $|a\rangle$ and $|b\rangle$. In this case

$$0 = A_{a,b} \stackrel{?}{=} \sum_i \lambda_i \langle a|i\rangle\langle i|b\rangle, \tag{28}$$

and since $\lambda = 1$, we can write, that $\lambda^* = \lambda$, and so

$$\begin{aligned} 1 = A_{b,a} \stackrel{?}{=} \sum_i \lambda_i \langle b|i\rangle\langle i|a\rangle = \left(\sum_i \lambda_i^* \langle b|i\rangle^* \langle i|a\rangle^*\right)^* = \\ = \left(\sum_i \lambda_i \langle a|i\rangle\langle i|b\rangle\right)^* \stackrel{?}{=} A_{b,a}^* = 0 \end{aligned} \tag{29}$$

$$\tag{30}$$

This last equation clearly cannot be true, because it would say $1 = 0$. The solution is that A cannot be written in the diagonalized form.

2.1.6 Adjoints and Hermitian operators

Exercise 2.13 To show what is the operator $(|w\rangle\langle v|)^\dagger$, we have to show how does this operator act on any vector. It is enough to see what this operator looks like sandwiched between the orthonormal basis vectors.

$$\begin{aligned} \langle i|(|w\rangle\langle v|)^\dagger|j\rangle = \big(|i\rangle, (|w\rangle\langle v|)^\dagger|j\rangle\big) = \big((|w\rangle\langle v|)^\dagger|j\rangle, |i\rangle\big)^* = \\ = (|j\rangle, (|w\rangle\langle v|)|i\rangle)^* = ((|w\rangle\langle v|)|i\rangle, |j\rangle)^{**} = (|w\rangle\langle v|i\rangle, |j\rangle) = \\ = \langle v|i\rangle^* (|w\rangle, |j\rangle) = \langle i|v\rangle\langle w|j\rangle = \langle i|(|v\rangle\langle w|)|j\rangle. \end{aligned} \tag{31}$$

Exercise 2.14 Similarly as before, it is enough to see how does the operator look like sandwiched between the orthonormal basis vectors.

$$\begin{aligned} \langle i| \left(\sum_k a_k A_k\right)^\dagger |j\rangle = \left(|i\rangle, \left(\sum_k a_k A_k\right)^\dagger |j\rangle\right) = \\ = \left(\left(\sum_k a_k A_k\right)^\dagger |j\rangle, |i\rangle\right)^* = \left(|j\rangle, \left(\sum_k a_k A_k\right)|i\rangle\right)^* = \\ = \sum_k a_k^* (|j\rangle, A_k|i\rangle)^* = \sum_k a_k^* (A_k|i\rangle, |j\rangle) = \\ = \sum_k a_k^* \left(|i\rangle, A_k^\dagger|j\rangle\right) = \left(|i\rangle, \left(\sum_k a_k^* A_k^\dagger\right)|j\rangle\right) = \langle i| \left(\sum_k a_k^* A_k^\dagger\right)|j\rangle. \end{aligned} \tag{32}$$

Exercise 2.15 Again, it is enough to see how does the operator look like sandwiched between the orthonormal basis vectors.

$$\begin{aligned}\langle i|(A^\dagger)^\dagger|j\rangle = (|i\rangle, (A^\dagger)^\dagger|j\rangle) = ((A^\dagger)^\dagger|j\rangle, |i\rangle)^* = (|j\rangle, A^\dagger|i\rangle)^* = \\ = (A^\dagger|i\rangle, |j\rangle)^{**} = (|i\rangle, A|j\rangle) = \langle i|A|j\rangle.\end{aligned} \tag{33}$$

Exercise 2.16 Because the vectors $|i\rangle$ are orthonormal,

$$P^2 = \left(\sum_{i=1}^{k} |i\rangle\langle i|\right)\left(\sum_{j=1}^{k} |j\rangle\langle j|\right) = \sum_{i=1}^{k}\sum_{j=1}^{k} |i\rangle \underbrace{\langle i|j\rangle}_{\delta i,j}\langle j| = \sum_{i=1}^{k} |i\rangle\langle i| = P. \tag{34}$$

Exercise 2.17 A normal matrix can be decomposed to linear combination of orthonormal vectors, and because of the properties of Hermitian conjugate,

$$\begin{aligned}A &= \sum_i \lambda_i |i\rangle\langle i|, \\ A^\dagger &= \sum_i \lambda_i^* |i\rangle\langle i|.\end{aligned} \tag{35}$$

If the operator is self adjoint, because of the orthonormality of the vectors,

$$\lambda_j = \langle j|Aj\rangle = \langle j|A^\dagger|j\rangle = \lambda_j^*, \tag{36}$$

which means, that the imaginary component of the eigenvalues must be zero, they are real.

If the eigenvalues are real, then we can clearly see, that the diagonalization formula for A and $A^\dagger$ are the same, so $A = A^\dagger$.

Exercise 2.18 If U is unitary, then it normal, which means, that it can be diagonalized. Let $|i\rangle$ be the orthonormal basis.

$$\begin{aligned}1 = \langle k|k\rangle = \langle k|I|k\rangle = \langle k|U^\dagger U|k\rangle = \\ = \langle k|\sum_{i,j} \lambda_i^*\lambda_j |i\rangle \underbrace{\langle i|j\rangle}_{\delta_{i,j}}\langle j|k\rangle = \sum_i |\lambda_i|^2 \underbrace{\langle k|i\rangle}_{\delta k,i}\underbrace{\langle i|k\rangle}_{\delta i,k} = |\lambda_k|^2.\end{aligned} \tag{37}$$

The absolute value of any eigenvalue is 1, which means that it can be written in the form $\lambda_k = e^{i\theta}$.

Exercise 2.19 We can verify that with direct matrix computation. To get the Hermitian conjugate of a matrix the elements have to be mirrored to the main

diagonal, and then complex conjugated.

$$\sigma_1^\dagger = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^\dagger = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \sigma_1,$$
$$\sigma_1^\dagger \sigma_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I,$$
$$\sigma_2^\dagger = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}^\dagger = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = \sigma_2,$$
$$\sigma_2^\dagger \sigma_2 = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}^2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I,$$
$$\sigma_3^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \sigma_3,$$
$$\sigma_3^\dagger \sigma_3 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}^2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I. \tag{38}$$

Exercise 2.20 Let's say, that U is the operator which changes between the orthonormal basis states $|v_i\rangle$ and $|w_i\rangle$: $|w_i\rangle = U|v_i\rangle$. U is unitary, because it preserves the inner products. Which means that it is normal, so we can write down, that

$$|w_j\rangle = U|v_j\rangle = \sum_i \lambda_i |v_i\rangle\langle v_i|v_j\rangle = \lambda_i |v_i\rangle, \tag{39}$$

so it follows, that $U = \sum_i |w_i\rangle\langle v_i|$. After constructing this operator, we can write,

$$A''_{i,j} = \langle w_i|A|w_j\rangle = \langle v_i|(U^\dagger A\ U)|v_j\rangle. \tag{40}$$

This means, that calculating the matrix $(U^\dagger A\ U)'_{i,j}$ which in the $|v_i\rangle$ basis is the matrix $A''_{i,j}$.

Exercise 2.21 After seeing, that $QMP = 0$, it is easy to see, that $PMQ = 0$, because $PMQ = (PMQ)^{\dagger\dagger} = (QM^\dagger P)^\dagger = (QMP)^\dagger = 0^\dagger = 0$. Proving, that QMQ is normal is much easier, since $(QMQ)(QMQ)^\dagger = (QMQ)(QM^\dagger Q) = (QM^\dagger Q)(QMQ)$.

Exercise 2.22 For a Hermitian operator A it is true, that $A^\dagger = A$. Then

$$0 = \langle Aw|v\rangle - \langle Aw|v\rangle = \langle w|A^\dagger|v\rangle - \langle A^\dagger w|v\rangle =$$
$$= \langle w|A|v\rangle - \langle A^\dagger w|v\rangle = \lambda_v\langle w|v\rangle - \lambda_w\langle w|v\rangle = (\lambda_v - \lambda_w)\langle w|v\rangle. \tag{41}$$

Because $(\lambda_v - \lambda_w) \neq 0$, $\langle w|v\rangle$ must be 0.

Exercise 2.23 For a projector operator P it's true, that $P^2 = P$. Let's normalize the eigenvector to 1. Then

$$\lambda = \langle v|\lambda|v\rangle = \langle v|P|v\rangle = \langle v|P^2|v\rangle = \langle v|\lambda^2|v\rangle = \lambda^2. \tag{42}$$

$\lambda = \lambda^2$, and there are only two numbers which satisfy this: 0, 1.

Exercise 2.24 [2] Any operator can be written as

$$A = \underbrace{\frac{1}{2}(A + A^\dagger)}_{B} + i\underbrace{\frac{1}{2i}(A - A^\dagger)}_{C}. \tag{43}$$

From this, it is obious, that $B^\dagger = B$ and $C^\dagger = C$, these are Hermitian operators. Let's write out B and C in their diagonalized forms.

$$\begin{aligned} B &= \sum_i \delta_i |w_i\rangle\langle w_i|, \\ C &= \sum_i \lambda_i |v_i\rangle\langle v_i|, \end{aligned} \tag{44}$$

and insert these into the equation $\langle v|A|v\rangle \geq 0$ with $|v\rangle$ being one of the eigenvectors of C.

$$\begin{aligned} 0 \leq \langle v_k|A|v_k\rangle &= \sum_i \delta_i |\langle v_k|w_i\rangle|^2 + i\sum_i \lambda_i |\langle v_k|v_i\rangle|^2 = \\ &= \sum_i \delta_i |\langle v_k|w_i\rangle|^2 + i\lambda_k. \end{aligned} \tag{45}$$

Because all δ_i, $|\langle v_k|w_i\rangle|^2$ and λ_k are nonnegative, and this whole expression should be real, $\lambda_k = 0$ for all k. This means, that $C = 0$. $A = B$, and B is Hermitian, therefore A is also Hermitian.

Exercise 2.25 For any A and $|v\rangle$

$$\langle v|A^\dagger A|v\rangle = \langle Av|Av\rangle, \tag{46}$$

and the inner product for the same vector is nonnegative because of the definition of the inner product space.

2.1.7 Tensor products

Exercise 2.26

$$\begin{aligned} |\psi\rangle^{\otimes 2} =& \frac{1}{2}(|0\rangle|0\rangle + |0\rangle|1\rangle + |1\rangle|0\rangle + |1\rangle|1\rangle) = \frac{1}{2}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}, \\ |\psi\rangle^{\otimes 3} =& \frac{1}{2}(|0\rangle|0\rangle|0\rangle + |0\rangle|0\rangle|1\rangle + |0\rangle|1\rangle|0\rangle + |0\rangle|1\rangle|1\rangle + \\ &+ |1\rangle|0\rangle|0\rangle + |1\rangle|0\rangle|1\rangle + |1\rangle|1\rangle|0\rangle + |1\rangle|1\rangle|1\rangle) = \frac{1}{2}\begin{bmatrix}1\\1\\1\\1\\1\\1\\1\\1\end{bmatrix}. \end{aligned} \tag{47}$$

Exercise 2.27

$$X \otimes Z = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix},$$

$$I \otimes X = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$X \otimes I = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{48}$$

From the last two equations, it is clear that tensor product is not commutative.

Exercise 2.28 Complex conjugation on an operator only makes sense in a specific matrix representation. In this representation each element is going to be in the form $A_{i,j}B_{k,l}$. The complex conjugate of this is $(A_{i,j}B_{k,l})^* = A^*_{i,j}B^*_{k,l}$, and this is exactly the matrix for the operator $A^* \otimes B^*$.

Similarly, the transpose only makes sense for a representation. It is true, that for the matrix $M = A \otimes B$, $(M^{\mathrm{T}})_{i,j} = M_{j,i}$. The i,jth index of the matrix equals to $A_{k,l}B_{m,n}$. k and m only depends on i. m and n only depends on j. This means, that for the transpose of M we have to get the transpose of A and B and get their tensor product.

Since Hermitian conjugate equals to transforming the matrix with complex conjugation and then getting it's transpose, and these operations are both distributable to the tensor product, the Hermitian conjugate is also distributable. This applies to any matrix representation, so this must be true to the operator too.

Exercise 2.29 Let's act the operator and it's Hermitian conjugate on any tensor product vector.

$$\begin{aligned}(U \otimes V)^\dagger (U \otimes V)|u\rangle \otimes |v\rangle = (U^\dagger \otimes V^\dagger)(U|u\rangle) \otimes (V|v\rangle) = \\ (U^\dagger U|u\rangle) \otimes (V^\dagger V|v\rangle) = |u\rangle \otimes |v\rangle.\end{aligned} \tag{49}$$

Because this acts like the identity for any tensor product vector, this acts like the identity for any of the linear combination of the tensor product vectors, which is any vector in the tensor product space, so the tensor product of two unitary operators is unitary.

Exercise 2.30 Using the previous results for $M = A \otimes B$, where A and B are Hermitian

$$M^\dagger = (A \otimes B)^\dagger = A^\dagger \otimes B^\dagger = A \otimes B = M. \tag{50}$$

Exercise 2.31 A and B are positive operators, so for any $|a\rangle \otimes |b\rangle$

$$(|a\rangle \otimes |b\rangle, (A \otimes B)|a\rangle \otimes |b\rangle) = (|a\rangle \otimes |b\rangle, |Aa\rangle \otimes |Bb\rangle) = \\ = \langle a|A|a\rangle\langle b|B|b\rangle. \tag{51}$$

Both parts of this expression are nonnegative, so this whole expression is nonnegative. Any tensor product vector is the linear combination of these kinds of tensor product vectors, and because the inner product is distributive and linear, $\langle v|A \otimes B|v\rangle$ is nonnegative for any vector, so this operator is positive.

Exercise 2.32 P and Q are projectors. Then

$$(P \otimes Q)(P \otimes Q)|p\rangle \otimes |q\rangle = (P \otimes Q)|Pp\rangle \otimes |Qq\rangle = \\ = |P^2p\rangle \otimes |Q^2q\rangle = |Pp\rangle \otimes |Qq\rangle = (P \otimes Q)|p\rangle \otimes |q\rangle. \tag{52}$$

This is true for any $|p\rangle \otimes |q\rangle$, and any vector in the tensor product space is the linear combination of these, and using the distributability and linearity properties of $(P \otimes Q)$ we get, that this is true for any vector in the tensor product space.

Exercise 2.33 Let's use induction. For $n = 1$ H can be rewritten as

$$H^{\otimes 1} = H = \frac{1}{\sqrt{2}}(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| + |1\rangle\langle 1|) = \\ = \frac{1}{\sqrt{2^1}}((-1)^{0\cdot 0}|0\rangle\langle 0| + (-1)^{0\cdot 1}|0\rangle\langle 1| + (-1)^{1\cdot 0}|1\rangle\langle 0| + (-1)^{1\cdot 1}|1\rangle\langle 1|) = \\ = \frac{1}{\sqrt{2^1}}\sum_{x,y}(-1)^{xy}|x\rangle\langle y|. \tag{53}$$

Let's assume, that $H^{\otimes n}$ can be written in that form. Tensor multiply this with H.

$$H^{\otimes(n+1)} = H^{\otimes n} \otimes H = \\ = \frac{1}{\sqrt{2^n}}\sum_{x^n,y^n}(-1)^{x^n y^n}|x^n\rangle\langle y^n| \otimes \frac{1}{\sqrt{2}}\sum_{x_{n+1},y_{n+1}}(-1)^{x_{n+1}y_{n+1}}|x_{n+1}\rangle\langle y_{n+1}| = \\ = \frac{1}{\sqrt{2^{n+1}}}\sum_{x_1,y_1}\dots\sum_{x_n,y_n}(-1)^{x_1y_1+\dots+x_ny_n}|x_1\rangle\langle y_1| \otimes \dots \otimes |x_n\rangle\langle y_n| \otimes \\ \otimes \sum_{x_{n+1},y_{n+1}}(-1)^{x_{n+1}y_{n+1}}|x_{n+1}\rangle\langle y_{n+1}| = \\ = \frac{1}{\sqrt{2^{n+1}}}\sum_{x^{n+1},y^{n+1}}(-1)^{x^{n+1}y^{n+1}}|x^{n+1}\rangle\langle y^{n+1}|. \tag{54}$$

Because $\langle 0|H|0\rangle = \langle 0|H|1\rangle = \langle 1|H|0\rangle = 1/\sqrt{2}$ and $\langle 1|H|1\rangle = -1/\sqrt{2}$,

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix}. \tag{55}$$

Then $H^{\otimes 2}$ can be written in the matrix form using the rule written in (2.50) in the book

$$H^{\otimes 2} = \frac{1}{2}\begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}. \tag{56}$$

2.1.8 Operator functions

Exercise 2.34 The matrix has eigenvalues and normalized eigenvectors

$$\begin{aligned} |v_1\rangle &= \frac{1}{\sqrt{2}}\begin{bmatrix} +1 \\ +1 \end{bmatrix}, \lambda_1 = 7, \\ |v_2\rangle &= \frac{1}{\sqrt{2}}\begin{bmatrix} +1 \\ -1 \end{bmatrix}, \lambda_2 = 1. \end{aligned} \tag{57}$$

Because the matrix is normal, we can diagonalize it.

$$\frac{\lambda_1}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \end{bmatrix} + \frac{\lambda_2}{2}\begin{bmatrix} 1 \\ -1 \end{bmatrix}\begin{bmatrix} 1 & -1 \end{bmatrix} = \frac{\lambda_1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \frac{\lambda_2}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \tag{58}$$

Then, using the definition of the operator functions

$$\begin{aligned} \sqrt{\begin{bmatrix} 4 & 3 \\ 3 & 4 \end{bmatrix}} &= \frac{\sqrt{7}}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \frac{\sqrt{1}}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} \sqrt{7}+1 & \sqrt{7}-1 \\ \sqrt{7}-1 & \sqrt{7}+1 \end{bmatrix}, \\ \log\left(\begin{bmatrix} 4 & 3 \\ 3 & 4 \end{bmatrix}\right) &= \frac{\log(7)}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \underbrace{\frac{\log(1)}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}}_{=0} = \frac{\log(7)}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}. \end{aligned} \tag{59}$$

Exercise 2.35 Pauli matrices are Hermitian, so their linear combination, which is $i\theta\sum_i v_i\sigma_i$ is also Hermitian, so this expression is a normal operator, it has a diagonalization. To diagonalize, let's first find out it's eigenvalues and eigenvectors of $\sum_i v_i\sigma_i$.

$$\begin{aligned} \sum_i v_i\sigma_i &= \begin{bmatrix} v_3 & v_2 - iv_1 \\ v_2 + iv_1 & -v_3 \end{bmatrix}, \\ 0 \overset{!}{=} \begin{vmatrix} v_3 - \lambda & v_2 - iv_1 \\ v_2 + iv_1 & -v_3 - \lambda \end{vmatrix} &= \lambda^2 \underbrace{-v_1^2 - v_2^2 - v_3^2}_{=-1} \;\Rightarrow\; \lambda_{+1} = +1, \lambda_{-1} = -1. \end{aligned} \tag{60}$$

To get the eigenvectors we have to solve the linear equations.

$$\begin{aligned} \begin{bmatrix} v_3 - 1 & v_2 - iv_1 \\ v_2 + iv_1 & -v_3 - 1 \end{bmatrix}|+1\rangle \overset{!}{=} 0 \;\Rightarrow\; |+1\rangle &= n_+\begin{bmatrix} v_1 - iv_2 \\ 1 - v_3 \end{bmatrix}, \\ \begin{bmatrix} v_3 + 1 & v_2 - iv_1 \\ v_2 + iv_1 & -v_3 + 1 \end{bmatrix}|-1\rangle \overset{!}{=} 0 \;\Rightarrow\; |-1\rangle &= n_-\begin{bmatrix} v_1 - iv_2 \\ 1 + v_3 \end{bmatrix}. \end{aligned} \tag{61}$$

The normalization condition for the eigenvectors is that $\langle +1|+1\rangle = \langle -1|-1\rangle = 1$, therefore

$$n_+ = \frac{1}{\sqrt{2(1-v_3)}},\ |+1\rangle = \frac{1}{\sqrt{2(1-v_3)}}\begin{bmatrix} v_1 - iv_2 \\ 1 - v_3 \end{bmatrix},$$
$$n_- = \frac{1}{\sqrt{2(1+v_3)}},\ |-1\rangle = \frac{1}{\sqrt{2(1+v_3)}}\begin{bmatrix} iv_2 - v_1 \\ 1 + v_3 \end{bmatrix}. \tag{62}$$

If for an operator A it is true, that $A|v\rangle = \lambda|v\rangle$, then it true, that $(\alpha A)|v\rangle = (\alpha\lambda)|v\rangle$. This means, that for $i\theta\sum_i v_i\sigma_i$ the eigenvalues will be multiplied by $i\theta$, and the eigenvectors will stay the same compared to the results we just calculated. Let's find out the outer products of the eigenvectors.

$$|+1\rangle\langle +1| = \frac{1}{2(1-v_3)}\begin{bmatrix} v_1 - iv_2 \\ 1 - v_3 \end{bmatrix}\begin{bmatrix} v_1 + iv_2 & 1 - v_3 \end{bmatrix} =$$
$$= \frac{1}{2(1-v_3)}\begin{bmatrix} (v_1 - iv_2)(v_1 + iv_2) & (v_1 - iv_2)(1 - v_3) \\ (1 - v_3)(v_1 + iv_2) & (1 - v_3)(1 - v_3) \end{bmatrix} =$$
$$= \frac{1}{2}\begin{bmatrix} \frac{v_1^2+v_2^2}{1-v_3} & v_1 - iv_2 \\ v_1 + iv_2 & 1 - v_3 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 + v_3 & v_1 - iv_2 \\ v_1 + iv_2 & 1 - v_3 \end{bmatrix} =$$
$$= \frac{1}{2}\left(I + \sum_i v_i\sigma_i\right). \tag{63}$$

(Where I used the fact, that $1/(1-v_3) = (1+v_3)/((1+v_3)(1-v_3)) = (1+v_3)/(1-v_3^2) = (1+v_3)/(v_1^2+v_2^2)$.) Similarly for the other eigenvector.

$$|-1\rangle\langle -1| = \frac{1}{2(1+v_3)}\begin{bmatrix} iv_2 - v_1 \\ 1 + v_3 \end{bmatrix}\begin{bmatrix} iv_2 + v_1 & 1 + v_3 \end{bmatrix} =$$
$$= \frac{1}{2}\left(I - \sum_i v_i\sigma_i\right). \tag{64}$$

To get $\exp(i\theta\sum_i v_i\sigma_i)$ we use these diagonalizations.

$$\exp(i\theta\sum_i v_i\sigma_i) = \exp(i\theta)|+1\rangle\langle +1| + \exp(-i\theta)|-1\rangle\langle -1| =$$
$$= \frac{1}{2}(\cos(\theta) + i\sin(\theta))\left(I + \sum_i v_i\sigma_i\right) + \frac{1}{2}(\cos(\theta) - i\sin(\theta))\left(I - \sum_i v_i\sigma_i\right) =$$
$$= \cos(\theta)I + i\sin(\theta)\sum_i v_i\sigma_i. \tag{65}$$

Exercise 2.36 σ_1 and σ_2 diagonal elements are zero, therefore $\mathrm{tr}(\sigma_1) = 0$ and $\mathrm{tr}(\sigma_2) = 0$. The diagonal elements of σ_3 are +1 and -1, therefore $\mathrm{tr}(\sigma_3) = 0$.

Exercise 2.37 $\mathrm{tr}(AB) = \sum_i (AB)_{i,i} = \sum_i\sum_j A_{i,j}B_{j,i} = \sum_j\sum_i B_{j,i}A_{i,j} = \sum_j (BA)_{j,j} = \mathrm{tr}(BA)$.

Exercise 2.38 $\mathrm{tr}(A+B) = \sum_i (A+B)_{i,i} = \sum_i \langle i|(A+B)|i\rangle = \sum_i \langle i|(A|i\rangle + B|i\rangle) = \sum_i \langle i|A|i\rangle + \langle i|B|i\rangle = \sum_i A_{i,i} + \sum_i B_{i,i} = \mathrm{tr}(A) + \mathrm{tr}(B)$. $\mathrm{tr}(zA) = \sum_i (zA)_{i,i} = \sum_i zA_{i,i} = z\sum_i A_{i,i} = z\mathrm{tr}(A)$.

Exercise 2.39

(i) The inner product should be linear in the first argument.

$$\begin{aligned}(A, B+C) = \mathrm{tr}(A^\dagger(B+C)) = \mathrm{tr}(A^\dagger B + A^\dagger C) = \\ = \mathrm{tr}(A^\dagger B) + \mathrm{tr}(A^\dagger C) = (A,B) + (A,C), \\ (A, zB) = \mathrm{tr}(A^\dagger(zB)) = \mathrm{tr}(z(A^\dagger B)) = z\mathrm{tr}(A^\dagger B) = z(A,B). \end{aligned} \tag{66}$$

The inner product should have conjugate symmetry.

$$\begin{aligned}(A,B) = \mathrm{tr}(A^\dagger B) = \sum_i (A^\dagger B)_{i,i} = \sum_i \sum_j (A^\dagger)_{i,j} B_{j,i} = \\ = \sum_i \sum_j A^*_{j,i} B_{j,i} = \sum_i \sum_j (B^*_{j,i} A_{j,i})^* = \left(\sum_i \sum_j B^\dagger_{i,j} A_{j,i}\right)^* = \\ = \left(\sum_i (B^\dagger A)_{i,i}\right)^* = (\mathrm{tr}(B,A))^* = (B,A)^*. \end{aligned} \tag{67}$$

The last property inner product should have is that $(A,A) \geq 0$ for all A, and 0 only if $A = 0$. We already know, that $A^\dagger A$ is positive, and it is a Hermitian operator. This means, that there is an orthonormal basis where it only has diagonal elements, which are all positive. This means, that $(A,A) = \mathrm{tr}(A^\dagger A) \geq 0$. If $A = 0$, then $(A,A) = 0$. Let's prove the implication for the other way. Let's again look at the operator $A^\dagger A$ in the orthonormal basis where it only has diagonal positive elements. Then $0(A,A) = \mathrm{tr}(A^\dagger A)$ can only happen if the diagonal elements are zero, and because all other elements of the matrix are 0, then $A = 0$.

(ii) Any vector can be written up as the linear combination of the orthonormal basis vectors. Also, and operator is fully described by how it acts on the vectors.

$$\begin{aligned}|w\rangle = A|v\rangle, \\ \langle i|w\rangle = \sum_j \underbrace{\langle i|A|j\rangle}_{A_{i,j}} \langle j|v\rangle. \end{aligned} \tag{68}$$

This means, that if we know the numbers $A_{i,j}$ then we fully know how the operator acts on any vector. There are d^2 of these numbers, and knowing these numbers the operator can be written in the form

$$A = \sum_i \sum_j A_{i,j} |i\rangle\langle j|, \tag{69}$$

so that this linear combination of these operators will act just like the original A. There are d^2 of these.

(iii) Let's define the following d^2 operators.

$$E_{i,j} = \delta_{i,j}|i\rangle\langle j| - \frac{i}{\sqrt{2}}(1-\delta_{i,j})(|i\rangle\langle j| - |j\rangle\langle i|). \tag{70}$$

Let's calculate it's Hermitian conjugate

$$E_{i,j}^\dagger = \delta_{i,j}|j\rangle\langle i| + \frac{i}{\sqrt{2}}(1-\delta_{i,j})(|j\rangle\langle i| - |i\rangle\langle j|). \tag{71}$$

This is the same operator, which means that it is Hermitian. Also, these are orthonormal, because

$$\begin{aligned}\mathrm{tr}(E_{i,j}^\dagger E_{k,l}) = \Big(\delta_{j,i}|j\rangle\langle i| + i(1-\delta_{j,i})(|j\rangle\langle i| - |i\rangle\langle j|)/\sqrt{2}\Big)\cdot \\ \cdot\Big(\delta_{k,l}|k\rangle\langle l| - i(1-\delta_{k,l})(|k\rangle\langle l| - |l\rangle\langle k|)/\sqrt{2}\Big).\end{aligned} \tag{72}$$

To simplify this, we have to use, that $\mathrm{tr}(|i\rangle\langle j|) = \delta_{i,j}$, and $\langle i|j\rangle = \delta_{i,j}$. Also, it is true, that terms like

$$(1-\delta_{i,j})\delta_{k,l}\delta_{i,k}\delta_{j,l} \tag{73}$$

will be 0, because the last three δ tells, that those are nonzero only when $i=k$, $k=l$, $l=j$, which means, that $i=j$, but if that's true, the first $(1-\delta_{i,j})$ will be zero. This term is always 0. All in all, we get, that

$$\begin{aligned}\mathrm{tr}(E_{i,j}^\dagger E_{k,l}) = \\ = \delta_{i,j}\delta_{k,l}\delta_{i,k}\delta_{j,l} + (1-\delta_{i,j})(1-\delta_{k,l})(\delta_{i,k}\delta_{j,l} - \delta_{i,l}\delta_{j,k}).\end{aligned} \tag{74}$$

There are 2×2 possibilities: $i \stackrel{?}{=} j$ and $k \stackrel{?}{=} l$. If they are both true, the second term is zero, and this can only be 1, if $i=k$ and $j=l$. If only one of these is true, then the first and second term is zero, but this also means, that $i \neq k$ or $j \neq l$. If none of these is true, the first term is zero, and the last term in the last parenthesis is necessarily zero, and the first term in that parenthesis only 1, if $i=k$ and $j=l$. All in all this means, that

$$(E_{i,j}, E_{k,l}) = \delta i,k\delta_{j,l}, \tag{75}$$

which is the condition for the orthonormality.

2.1.9 The commutator and anticommutator

Exercise 2.40

$$\begin{aligned}[X,Y] &= \begin{bmatrix}0&1\\1&0\end{bmatrix}\begin{bmatrix}0&-i\\i&0\end{bmatrix} - \begin{bmatrix}0&-i\\i&0\end{bmatrix}\begin{bmatrix}0&1\\1&0\end{bmatrix} = \begin{bmatrix}i&0\\0&-i\end{bmatrix} - \begin{bmatrix}-i&0\\0&i\end{bmatrix} = 2i\begin{bmatrix}1&0\\0&1\end{bmatrix} = 2iZ,\\
[Y,Z] &= \begin{bmatrix}0&-i\\i&0\end{bmatrix}\begin{bmatrix}1&0\\0&-1\end{bmatrix} - \begin{bmatrix}1&0\\0&-1\end{bmatrix}\begin{bmatrix}0&-i\\i&0\end{bmatrix} = \begin{bmatrix}0&i\\i&0\end{bmatrix} - \begin{bmatrix}0&-i\\-i&0\end{bmatrix} = 2i\begin{bmatrix}0&1\\1&0\end{bmatrix} = 2iX,\\
[Z,X] &= \begin{bmatrix}1&0\\0&-1\end{bmatrix}\begin{bmatrix}0&1\\1&0\end{bmatrix} - \begin{bmatrix}0&1\\1&0\end{bmatrix}\begin{bmatrix}1&0\\0&-1\end{bmatrix} = \begin{bmatrix}0&1\\-1&0\end{bmatrix} - \begin{bmatrix}0&-1\\1&0\end{bmatrix} = 2i\begin{bmatrix}0&-i\\i&0\end{bmatrix} = 2iY.\end{aligned} \tag{76}$$

Exercise 2.41

$$\begin{aligned}
\{X,Y\} &= \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix} + \begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} = \begin{bmatrix}i & 0\\ 0 & -i\end{bmatrix} + \begin{bmatrix}-i & 0\\ 0 & i\end{bmatrix} = \begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix},\\
\{Y,Z\} &= \begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} + \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix} = \begin{bmatrix}0 & i\\ i & 0\end{bmatrix} + \begin{bmatrix}0 & -i\\ -i & 0\end{bmatrix} = \begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix},\\
\{Z,X\} &= \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} + \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} = \begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix} + \begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix} = \begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}.
\end{aligned} \tag{77}$$

$$\begin{aligned}
I^2 &= I,\\
X^2 &= \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},\\
Y^2 &= \begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},\\
Z^2 &= \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}.
\end{aligned} \tag{78}$$

(79)

Exercise 2.42

$$\frac{[A,B]+\{A,B\}}{2} = \frac{AB-BA+AB+BA}{2} = \frac{2AB}{2} = AB \tag{80}$$

Exercise 2.43 From the previous results, that

$$[\sigma_j,\sigma_k] = 2i\sum_{l=1}^{3}\epsilon_{jkl}\sigma_l,\ \{\sigma_j,\sigma_k\} = 2\delta_{j,k}I, \tag{81}$$

it is easy to see, that

$$\sigma_j\sigma_k = \frac{1}{2}\left([\sigma_j,\sigma_k]+\{\sigma_j,\sigma_k\}\right) = i\sum_{l=1}^{3}\epsilon_{jkl}\sigma_l + \delta_{j,k}I. \tag{82}$$

Exercise 2.44 $AB = ([A,B]+A,B)/2$, and both the commutator and anticommutator are 0, which means, that $AB = 0$. We multiply this with A^{-1} from the left to get $B = 0$.

Exercise 2.45 $[A,B]^\dagger = (AB-BA)^\dagger = (AB)^\dagger-(BA)^\dagger = B^\dagger A^\dagger - A^\dagger B^\dagger = [B^\dagger,A^\dagger]$.

Exercise 2.46 $[A,B] = AB-BA = -(BA-AB) = -[B,A]$.

Exercise 2.47 $(i[A,B])^\dagger = -i[A,B]^\dagger = -i[B^\dagger,A^\dagger] = -i[B,A] = i[A,B]$.

2.1.10 The polar and singular value decompositions

Exercise 2.48 $P = I \cdot P$, $U = U \cdot I$. For the H Hermitian matrix $J = \sqrt{H^\dagger H} = \sqrt{H^2} = |H|$. Let's write this out in a diagonalized form.

$$J = |H| = \sum_i |\lambda_i||i\rangle\langle i|. \tag{83}$$

To get the original matrix H we need the U unitary matrix to be of the form

$$U = \sum_j \text{sign}(\lambda_j)|j\rangle\langle j|, \tag{84}$$

because this will ensure, that $UJ = H$. We can clearly see, that this U is unatary.

Exercise 2.49 For a normal matrix

$$\begin{aligned} A &= \sum_i \lambda_i |i\rangle\langle i|, \\ A^\dagger &= \sum_i \lambda_i^* |i\rangle\langle i|, \end{aligned} \tag{85}$$

which means, that

$$J = \sqrt{A^\dagger A} = \sum_i |\lambda_i||i\rangle\langle i|. \tag{86}$$

Only the absolute value of the complex $\lambda_i = \exp(i\varphi_i)|\lambda_i|$ remains in J. To get the correct phases, and get A, the unitary U must be

$$U = \sum_j \exp(i\varphi_j)|j\rangle\langle j|. \tag{87}$$

Exercise 2.50 In order to do it, first, we have to calculate $A^\dagger A$, and calculate it's eigenvalues and eigenvectors, and calculate the diagonalization of the operator. Then, from here, we can use operator function to calculate $J = \sqrt{A^\dagger A}$. To calculate S we have to calculate J^{-1} using operator function again, and use $S = AJ^{-1}$.

$$\begin{aligned} A &= \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \\ A^\dagger &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \\ J^2 = A^\dagger A &= \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}. \end{aligned} \tag{88}$$

Let's find eigenvalues and eigenvectors of J^2.

$$0 = \begin{vmatrix} 2-\lambda & 1 \\ 1 & 1-\lambda \end{vmatrix} = \lambda^2 - 3\lambda + 1,$$

$$\lambda_\pm = \frac{3}{2} \pm \frac{\sqrt{5}}{2}.$$

$$\begin{bmatrix} 2-\lambda_+ & 1 \\ 1 & 1-\lambda_+ \end{bmatrix} |+\rangle = 0.$$

$$|+\rangle = n_+ \begin{bmatrix} 1 \\ \frac{\sqrt{5}-1}{2} \end{bmatrix}.$$

$$\begin{bmatrix} 2-\lambda_- & 1 \\ 1 & 1-\lambda_- \end{bmatrix} |-\rangle = 0.$$

$$|-\rangle = n_- \begin{bmatrix} 1 \\ \frac{-\sqrt{5}-1}{2} \end{bmatrix}. \tag{89}$$

$|+\rangle$ and $|-\rangle$ should be normalized to 1, which means, that

$$|+\rangle = \frac{1}{\sqrt{\frac{5-\sqrt{5}}{2}}} \begin{bmatrix} 1 \\ \frac{\sqrt{5}-1}{2} \end{bmatrix}, \lambda_+ = \frac{3+\sqrt{5}}{2},$$

$$|-\rangle = \frac{1}{\sqrt{\frac{5+\sqrt{5}}{2}}} \begin{bmatrix} 1 \\ \frac{-\sqrt{5}-1}{2} \end{bmatrix}, \lambda_- = \frac{3-\sqrt{5}}{2}. \tag{90}$$

Then, we can calculate the outer products.

$$|+\rangle\langle+| = \frac{1}{\frac{5-\sqrt{5}}{2}} \begin{bmatrix} 1 & \frac{\sqrt{5}-1}{2} \\ \frac{\sqrt{5}-1}{2} & \frac{3-\sqrt{5}}{2} \end{bmatrix},$$

$$|-\rangle\langle-| = \frac{1}{\frac{5+\sqrt{5}}{2}} \begin{bmatrix} 1 & \frac{-\sqrt{5}-1}{2} \\ \frac{-\sqrt{5}-1}{2} & \frac{3+\sqrt{5}}{2} \end{bmatrix}. \tag{91}$$

From this, J can be calculated.

$$J = \sqrt{J^2} = \sqrt{\lambda_+}|+\rangle\langle+| + \sqrt{\lambda_-}|-\rangle\langle-| =$$

$$= \frac{\sqrt{\frac{3+\sqrt{5}}{2}}}{\frac{5-\sqrt{5}}{2}} \begin{bmatrix} 1 & \frac{\sqrt{5}-1}{2} \\ \frac{\sqrt{5}-1}{2} & \frac{3-\sqrt{5}}{2} \end{bmatrix} + \frac{\sqrt{\frac{3-\sqrt{5}}{2}}}{\frac{5+\sqrt{5}}{2}} \begin{bmatrix} 1 & \frac{-\sqrt{5}-1}{2} \\ \frac{-\sqrt{5}-1}{2} & \frac{3+\sqrt{5}}{2} \end{bmatrix} = \frac{1}{\sqrt{5}} \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}. \tag{92}$$

To calculate S, we first calculate J^{-1}.

$$J^{-1} = \sqrt{J^2}^{-1} = \frac{1}{\sqrt{\lambda_+}}|+\rangle\langle+| + \frac{1}{\sqrt{\lambda_-}}|-\rangle\langle-| =$$

$$= \frac{1}{\sqrt{\frac{3+\sqrt{5}}{2}\left(\frac{5-\sqrt{5}}{2}\right)}} \begin{bmatrix} 1 & \frac{\sqrt{5}-1}{2} \\ \frac{\sqrt{5}-1}{2} & \frac{3-\sqrt{5}}{2} \end{bmatrix} + \frac{1}{\sqrt{\frac{3-\sqrt{5}}{2}\left(\frac{5+\sqrt{5}}{2}\right)}} \begin{bmatrix} 1 & \frac{-\sqrt{5}-1}{2} \\ \frac{-\sqrt{5}-1}{2} & \frac{3+\sqrt{5}}{2} \end{bmatrix} = \frac{1}{\sqrt{5}} \begin{bmatrix} 2 & -1 \\ -1 & 3 \end{bmatrix}. \tag{93}$$

And finally, S is

$$S = AJ^{-1} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \frac{1}{\sqrt{5}} \begin{bmatrix} 2 & -1 \\ -1 & 3 \end{bmatrix} = \frac{1}{\sqrt{5}} \begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix}. \tag{94}$$

2.2 The postulates of quantum mechanics

2.2.2 Evolution

Exercise 2.51

$$H^\dagger H = \frac{1}{2}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} = I. \tag{95}$$

Exercise 2.52 $H^\dagger = H$, and we already saw, that $H^\dagger H = I$, then $H^2 = I$.

Exercise 2.53 The eigenvalues are calculated with the characteristic equation.

$$0 \overset{!}{=} \begin{vmatrix}1/\sqrt{2}-\lambda & 1/\sqrt{2}\\ 1/\sqrt{2} & -1/\sqrt{2}-\lambda\end{vmatrix} = \lambda^2 - 1 \;\Rightarrow\; \lambda_+ = +1, \lambda_- = -1. \tag{96}$$

The eigenvectors are calculated with linear equations.

$$\begin{aligned} 0 &\overset{!}{=} \begin{bmatrix}1/\sqrt{2}-1 & 1/\sqrt{2}\\ 1/\sqrt{2} & -1/\sqrt{2}-1\end{bmatrix}|+\rangle \;\Rightarrow\; |+\rangle = n_+\begin{bmatrix}1/\sqrt{2}\\ 1/\sqrt{2}-1\end{bmatrix},\\ 0 &\overset{!}{=} \begin{bmatrix}1/\sqrt{2}+1 & 1/\sqrt{2}\\ 1/\sqrt{2} & -1/\sqrt{2}+1\end{bmatrix}|-\rangle \;\Rightarrow\; |-\rangle = n_-\begin{bmatrix}1/\sqrt{2}\\ 1/\sqrt{2}+1\end{bmatrix}. \end{aligned} \tag{97}$$

After normalization the normalized eigenvectors are

$$|+\rangle = \frac{1}{\sqrt{2-\sqrt{2}}}\begin{bmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}-1\end{bmatrix}, \tag{98}$$

$$|-\rangle = \frac{1}{\sqrt{2+\sqrt{2}}}\begin{bmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}+1\end{bmatrix}. \tag{99}$$

Exercise 2.54 If $[A, B] = 0$, and they are Hermitian, then there exists an orthonormal basis, and both A and B are diagonal with respect to this same basis.

$$\exp(A)\exp(B) = \left(\sum_i \exp(A_i)|i\rangle\langle i|\right)\left(\sum_j \exp(B_j)|j\rangle\langle j|\right) =$$

$$= \sum_i\sum_j \exp(A_i+B_j)|i\rangle \underbrace{\langle i|j\rangle}_{\delta i,j}\langle j| = \sum_i \exp(A_i+B_i)|i\rangle\langle i| = \exp(A+B), \tag{100}$$

because $A + B$ has the same eigenvectors with $A_i + B_i$ eigenvalues, so the diagonalization of them can be used to construct $\exp(A + B)$ easily.

Exercise 2.55 Let's say, that the eigenvectors of H are $|E\rangle$ with E eigenvalue, and that $\alpha = -(t_2 - t_1)/\hbar$ and $U = U(t_1, t_2)$. Then

$$U^\dagger U = \exp(i\alpha H)^\dagger \exp(i\alpha H) =$$
$$= \left(\sum_E \exp(i\alpha E)|E\rangle\langle E|\right)^\dagger \left(\sum_j \exp(i\alpha E)|E\rangle\langle E|\right) =$$
$$= \left(\sum_E \exp(-i\alpha E)|E\rangle\langle E|\right) \left(\sum_j \exp(i\alpha E)|E\rangle\langle E|\right) =$$
$$= \sum_E \underbrace{\exp(i\alpha E)\exp(-i\alpha E)}_{=1} |E\rangle\langle E| = I. \tag{101}$$

Exercise 2.56 Because a unitary operator has unit absolute value eigenvalues, and it has a diagonalization, we can write it out using real φ_i values as

$$U = \sum_i \exp(i\varphi_i)|i\rangle\langle i|. \tag{102}$$

These eigenvalues are an orthonormal set, which means, that

$$K^\dagger = (-i\log(U))^\dagger = \left(\sum_i (-i\log(\exp(i\varphi_i)))|i\rangle\langle i|\right)^\dagger =$$
$$= \sum_i (+i\log(\exp(-i\varphi_i)))|i\rangle\langle i| = \sum_i (-i\log(\exp(i\varphi_i)))|i\rangle\langle i| = -i\log(U). \tag{103}$$

2.2.3 Quantum measurement

Exercise 2.57 After the first measurement, the state will be

$$\frac{L_l}{\sqrt{\langle\psi|L_l^\dagger L_l|\psi\rangle}}|\psi\rangle, \tag{104}$$

with probability $p_L(l) = \langle\psi|L_l^\dagger L_l|\psi\rangle$. After that, we measure again with $\{M_m\}$ operator set to get

$$\frac{M_m}{\sqrt{\frac{\langle\psi|L_l^\dagger M_m^\dagger M_m L_l|\psi\rangle}{\langle\psi|L_l^\dagger L_l|\psi\rangle}}} \frac{L_l}{\sqrt{\langle\psi|L_l^\dagger L_l|\psi\rangle}}|\psi\rangle = \frac{M_m L_l}{\sqrt{\langle\psi|L_l^\dagger M_m^\dagger M_m L_l|\psi\rangle}}|\psi\rangle, \tag{105}$$

with probability that the measurement is $\{m, l\}$ from the original $|\psi\rangle$

$$p_{M,L}(\{m,l\}) = \frac{\langle\psi|L_l^\dagger M_m^\dagger M_m L_l|\psi\rangle}{\langle\psi|L_l^\dagger L_l|\psi\rangle} p_L(l) = \langle\psi|L_l^\dagger M_m^\dagger M_m L_l|\psi\rangle. \tag{106}$$

We can see, that this final state and probability is exactly the same if we used the measurement operator set $\{M_m L_l\}$.

2.2.5 Projective measurements

Exercise 2.58 The average is $\langle M\rangle = \langle\psi|M|\psi\rangle = \langle\psi|m|\psi\rangle = m\langle\psi|\psi\rangle = m$, and the standard deviation is $\langle M^2\rangle - \langle M\rangle^2 = \langle\psi|MM|\psi\rangle - m^2 = m^2 - m^2 = 0$.

Exercise 2.59 Becasue the X operator is the "bitflip" operator, and $|0\rangle = (1,0)$, we can calculate the average easily $\langle X\rangle = \langle 0|X|0\rangle = \langle 0|1\rangle = 0$ and the standard deviation $\langle X^2\rangle - \langle X\rangle^2 = \langle 0|XX|0\rangle - 0 = \langle 0|X|1\rangle = \langle 0|0\rangle = 1$.

Exercise 2.60 See **Exercise 2.35**, where it was already shown.

Exercise 2.61 From the previous exercise we saw, that the $+1$ eigenvalue corresponds to the projector operator $(I + \sum_i v_i\sigma_i)/2$. This means, that the probability that we measure $+1$ for the state $|0\rangle$ is

$$\begin{aligned}\langle 0|\frac{1}{2}(I + v_xX + v_yY + v_zZ)|0\rangle = \\ = \frac{1}{2}(\langle 0|I|0\rangle + v_x\langle 0|X|0\rangle + v_y\langle 0|Y|0\rangle + v_z\langle 0|Z|0\rangle) = \frac{1}{2}(1+v_z).\end{aligned} \tag{107}$$

After the measurement the state will be

$$\begin{aligned}\frac{\frac{1}{2}(I + v_xX + v_yY + v_zZ)|0\rangle}{\sqrt{(1+v_z)/2}} = \frac{(|0\rangle + v_x|1\rangle + v_yi|1\rangle + v_z|0\rangle)}{\sqrt{2(1+v_z)}} = \\ = \frac{(1+v_z)|0\rangle + (v_x+iv_y)|1\rangle}{\sqrt{2(1+v_z)}} = \frac{1}{\sqrt{2(1+v_z)}}\begin{bmatrix}1+v_z\\ v_x+iv_y\end{bmatrix}.\end{aligned} \tag{108}$$

It is the same as the eigenvector corresponding to the $+1$ eigenvalue of the measurement operator, and between them there is a unit phase multiplier $(v_1 - iv_2)/\sqrt{(1-v_3)(1+v_3)} = (v_1 - iv_2)/\sqrt{v_1^2+v_2^2}$.

2.2.6 POVM measurements

Exercise 2.62 We have to prove, that if $E_m = M_m$, then M_m is a projector. By definition, we know, that $E_m = M_m^\dagger M_m$. From this, we can see, that $(M_m)^\dagger = E_m^\dagger = (M_m^\dagger M_m)^\dagger = M_m^\dagger M_m = E_m = M_m$. Now, we can calculate whether M_m is a projector. $M_m^2 = M_mM_m = M_m^\dagger M_m = E_m = M_m$.

Exercise 2.63 Let's not write out the m index for the operators. Because of the polar decomposition theorem, there exist operators V and J such that $M = VJ$, and V is unitary and J is positive. This means, that $J^\dagger = J$, and

$$E = M^\dagger M = (VJ)^\dagger(VJ) = J^\dagger V^\dagger VJ = J^2. \tag{109}$$

This also means, that J^2 and E has the same exact diagonalization, $\lambda_i^{J^2} = \lambda_i^E$. Moreover, the diagonalization can be made in the same basis for J and J^2, and $(\lambda_i^J)^2 = \lambda_i^{J^2}$. Because J is positive, the only solution to this is that $\lambda_i^J = \sqrt{\lambda_i^{J^2}}$.

From this, we can conclude, that J and $\sqrt{E}$ has the same exact diagonalization, because $\lambda_i^J = \sqrt{\lambda_i^{J^2}} = \sqrt{\lambda_i^E}$, which means, that $\sqrt{E} = J$, and if we chose $U = V$ we get, that

$$M = U\sqrt{E}. \tag{110}$$

Exercise 2.64 Let's construct $|\phi_i\rangle$ vectors ($i = 1 \dots m$) where $|\phi_i\rangle$ is perpendicular to all $|\psi_j\rangle$ vectors, where $j \neq i$, and $|\phi_i\rangle$'s component which is parallel to $|\psi_i\rangle$ is one. That is $\langle\phi_i|\psi_i\rangle = \delta_{ij}$. This can be done for all i by first creating an orthonormal basis $|j\rangle$ from the $|\psi_j\rangle$ vectors, and then constructing the matrix $A_{kl} = \langle\psi_k|l\rangle$. We introduce two vectors: $(\delta_i)_k = \delta_{ik}$ and $(\phi_i)_l = \langle l|\phi_i\rangle$. To find $|\phi_i\rangle$, the following linear equation has to be solved.

$$A\phi_i = \delta_i. \tag{111}$$

We can now introduce the POVM operators. $E_i = \alpha|\phi_i\rangle\langle\phi_i|$ when $i = 1 \dots m$, and $E_m = I - \sum_{i=1}^m E_i$. The E_i operators are obviously positive, if α is positive. E_m operator is positive if α is sufficiently small.

2.2.7 Phase

Exercise 2.65 The following states also form an orthonormal basis.

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \tag{112}$$

$$|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{113}$$

It is impossible to get these states from each other by relative phase shift, because they are orthogonal to each other.

2.2.8 Composite systems

Exercise 2.66

$$\begin{aligned}
&\left(\frac{\langle 00| + \langle 11|}{\sqrt{2}}\right) X_1 Z_2 \left(\frac{|00\rangle + |11\rangle}{\sqrt{2}}\right) = \\
&= \frac{1}{2}(\langle 00| + \langle 11|) \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}_1 \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}_2 (|00\rangle + |11\rangle) = \\
&= \frac{1}{2}(\langle 00| + \langle 11|) \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}_1 (|00\rangle - |11\rangle) = \frac{1}{2}(\langle 00| + \langle 11|)(|10\rangle - |01\rangle) = \\
&= \frac{1}{2}(\langle 00|10\rangle + \langle 11|10\rangle - \langle 00|01\rangle - \langle 11|01\rangle) = 0
\end{aligned} \tag{114}$$

Exercise 2.67 The condition for U' to be unitary is $(U')^\dagger U' = I$. This is achieved with the following condition.

$$\delta_{kl} = \sum_i (U')^\dagger_{ki} U_{il} = \sum_i (U')^*_{ik} U'_{il}. \tag{115}$$

If the k and l index represents the W subspace, then this is automatically true, since for these $U'_{ik} = U_{ik}$, $U'_{il} = U_{il}$, and U already satisfies this.

If n is the dimension of V and m is the dimension of W have to find $n - m$ orthonormal, n dimensional vectors v_k, which are all orthogonal to the vectors w_k, whose components are $(w_k)_i = U_{ik}$. Note, that there are m w_k vectors. Finding these $n - m$ v_k orthonormal vectors is always possible, because the vectors are n dimensional, and in this case we can find n orthonormal vectors, out of which m already exist as represented by w_k. All in all then

$$(U')_{il} = (v_l)_i \tag{116}$$

solves the equation.

Exercise 2.68 Let's try to write down the state $|a\rangle \otimes |b\rangle$. Any single qubit is written in terms of a linear combination of the $|0\rangle$ and $|1\rangle$ state.

$$\begin{aligned} |a\rangle &= \alpha_0|0\rangle + \alpha_1|1\rangle, \\ |b\rangle &= \beta_0|0\rangle + \beta_1|1\rangle. \end{aligned} \tag{117}$$

From this, we can see from the rules of tensor product, that

$$|a\rangle \otimes |b\rangle = \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle. \tag{118}$$

We also can see that

$$\langle 00|\psi\rangle = \frac{1}{\sqrt{2}},\ \langle 11|\psi\rangle = \frac{1}{\sqrt{2}},\ \langle 01|\psi\rangle = 0. \tag{119}$$

From the third equation we can conclude, that $\alpha_0\beta_1 = 0$. Also, from the first equation we can see, that $\alpha_0\beta_0 \neq 0$, which means, that β_1 must be 0. But if $\beta_1 = 0$, then $\alpha_0\beta_1 = 0$, which is supposed to be $1/\sqrt{2}$ from the second equation. This is impossible.

2.3 Application: superdense coding

Exercise 2.69 The states $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$ are orthonormal. It follows, that these states are unit length, because all of them are

$$\left(\frac{\langle a| \pm \langle b|}{\sqrt{2}}\right)\left(\frac{|a\rangle \pm |b\rangle}{\sqrt{2}}\right) = \frac{\langle a|a\rangle + \langle b|b\rangle}{2} = \frac{2}{2} = 1. \tag{120}$$

Notice, that the no state from the first two states have any single basis vector in common with the last two states, which means, that they are perpendicular. We only need to check the first two states and the last two states whether they are perpendicular. They can be written in the form

$$\left(\frac{\langle a| + \langle b|}{\sqrt{2}}\right)\left(\frac{|a\rangle - |b\rangle}{\sqrt{2}}\right) = \frac{\langle a|a\rangle - \langle b|b\rangle}{2} = \frac{1-1}{2} = 0. \tag{121}$$

Exercise 2.70 Note, that for the single qubit components $\langle a|b\rangle = \delta_{ab}$. For the inner products only the terms which have the same qubit as second component will be nonzero because the second operator in $E \otimes I$ is the identity.
For the first state, the result is

$$\left(\frac{\langle 00| + \langle 11|}{\sqrt{2}}\right)(E \otimes I)\left(\frac{|00\rangle + |11\rangle}{\sqrt{2}}\right) = \frac{1}{2}(\langle 0|E|0\rangle\langle 0|0\rangle + \langle 1|E|1\rangle\langle 1|1\rangle) = \\ = \frac{\langle 0|E|0\rangle + \langle 1|E|1\rangle}{2}. \tag{122}$$

For the second state the result is

$$\left(\frac{\langle 00| - \langle 11|}{\sqrt{2}}\right)(E \otimes I)\left(\frac{|00\rangle - |11\rangle}{\sqrt{2}}\right) = \frac{\langle 0|E|0\rangle + \langle 1|E|1\rangle}{2}. \tag{123}$$

For the third state, the result is

$$\left(\frac{\langle 01| + \langle 10|}{\sqrt{2}}\right)(E \otimes I)\left(\frac{|01\rangle + |10\rangle}{\sqrt{2}}\right) = \frac{1}{2}(\langle 0|E|0\rangle\langle 1|1\rangle + \langle 1|E|1\rangle\langle 0|0\rangle) = \\ = \frac{\langle 0|E|0\rangle + \langle 1|E|1\rangle}{2}.$$

For the fourth state, the result is

$$\left(\frac{\langle 01| - \langle 10|}{\sqrt{2}}\right)(E \otimes I)\left(\frac{|01\rangle - |10\rangle}{\sqrt{2}}\right) = \frac{\langle 0|E|0\rangle + \langle 1|E|1\rangle}{2}. \tag{124}$$

Any kind of measurement will result with the same probability as we have just seen for any of the Bell states. This means, that Eve will not be able to know anything about what information is sent, not even probabilistically.

2.4 The density operator

2.4.2 General properties of the density operator

Exercise 2.71

$$\mathrm{tr}(\rho^2) = \sum_i \sum_j p_i p_j \mathrm{tr}(|\psi_i\rangle \underbrace{\langle \psi_i|\psi_j\rangle}_{\delta_{ij}} \langle \psi_j|) = \sum_i p_i^2 \leq \sum_i p_i = 1. \tag{125}$$

The last inequality is because $p_i \leq 1$, and then $p_i^2 \leq p_i$. If ρ is pure, there is only one p_i, which equals 1, so $p_i^2 = p_i$, which means for this case $\mathrm{tr}(\rho^2) = 1$. Conversely, if $\mathrm{tr}(\rho^2) = 1$ can only be true if all of the $p_i = 1$, which can only be for one state, which means ρ is pure.

Exercise 2.72

(1) First of all, let's realize, that a general density operator can be generated by iteratively adding new $|\psi'\rangle\langle\psi'|$ states to the density operator.

$$\rho' = (1 - p')\rho + p'|\psi'\rangle\langle\psi'|. \tag{126}$$

The general mixed state qubit can also be iteratively generated using this formula, and in that case, every $|\psi'\rangle$ are single qubit states. Let's prove, that if it is true, that

$$\rho = \frac{I + \sum_i r_i \sigma_i}{2},\ r^2 = \sum_i r_i^2 \leq 1, \tag{127}$$

then it is also true, that there are r_i' real numbers such that

$$\rho' = (1-p')\rho + p'|\psi'\rangle\langle\psi'| = \frac{I + \sum_i r_i' \sigma_i}{2},\ r'^2 = \sum_i r_i'^2 \leq 1, \tag{128}$$

where $|\psi'\rangle$ is a general single qubit state, and $0 \leq p' \leq 1$. The general $|\psi'\rangle$ is

$$|\psi'\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle, \tag{129}$$

which means, that

$$|\psi'\rangle\langle\psi'| =$$
$$= \cos^2\left(\frac{\theta}{2}\right)|0\rangle\langle 0| + \sin^2\left(\frac{\theta}{2}\right)|1\rangle\langle 1| + \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right)(e^{i\varphi}|1\rangle\langle 0| + e^{-i\varphi}|0\rangle\langle 1|). \tag{130}$$

It can also be seen, by writing out the Pauli matrices, that

$$\frac{I + \sum_i v_i \sigma_i}{2} =$$
$$= \frac{1}{2}((1+v_3)|0\rangle\langle 0| + (1-v_3)|1\rangle\langle 1| + (r_1 + ir_2)|1\rangle\langle 0| + (r_1 - ir_2)|0\rangle\langle 1|). \tag{131}$$

From these last two equation we can see, that we choose v_i to be

$$\begin{aligned} v_1 &= 2\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\varphi = \sin\theta\cos\varphi, \\ v_2 &= 2\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\varphi = \sin\theta\sin\varphi, \\ v_3 &= 2\cos^2\left(\frac{\theta}{2}\right) - 1 = \cos\theta, \end{aligned} \tag{132}$$

then it is true, that $v^2 = 1$ and

$$|\psi'\rangle\langle\psi'| = \frac{I + \sum_i v_i \sigma_i}{2} \tag{133}$$

Then, we already proved half of the statement in question, which is that if we choose $r_i' = (1-p')r_i + p'v_i$, then

$$\rho' = \frac{I + \sum_i r_i' \sigma_i}{2}. \tag{134}$$

Next, we have to prove, that $r'^2 \leq 1$.

$$r'^2 = \sum_i ((1-p')r_i + p'v_i)^2 =$$
$$(1-p')^2 \underbrace{r^2}_{\leq 1} + p'^2 \underbrace{v^2}_{=1} + 2(1-p')p' \sum_i r_i v_i. \tag{135}$$

Because of the Schwartz inequality, it is true, that

$$\sum_i r_i v_i \leq \underbrace{\sqrt{\sum_i r_i^2}}_{\leq 1} \underbrace{\sqrt{\sum_i v_i^2}}_{=1} \leq 1. \tag{136}$$

All in all

$$r'^2 \leq (1-p')^2 + p'^2 + 2(1-p')p' = 1. \tag{137}$$

And with this, all statements in question have been proven.

(2) If $\rho = I/2$, we can clearly see, that $r_i = 0$ for all i.

(3) First, let's prove, that $\left(\sum_i r_i \sigma_i\right)^2 = r^2$.

$$\left(\sum_i r_i \sigma_i\right)^2 = \sum_{ij} r_i r_j \sigma_i \sigma_j = \sum_{ij} r_i r_j \left(\delta_{ij} I + i \sum_l \epsilon_{ijl} \sigma_l\right) =$$
$$= \sum_{ij} r_i r_j \delta_{ij} I + i \underbrace{\sum_{ij} r_i r_j \sum_l \epsilon_{ijl} \sigma_l}_{=0} = I \cdot r^2. \tag{138}$$

Next, we just have to calculate the following. Note, that σ matrices are traceless.

$$\mathrm{tr}(\rho^2) = \mathrm{tr}\left(\frac{I^2 + 2\sum_i r_i \sigma_i + I \cdot r^2}{4}\right) = \frac{1+r^2}{2}. \tag{139}$$

From this, we can see, that $\mathrm{tr}(\rho^2) = 1$ (which is equivalent to the statement that ρ is pure) if and only if $r^2 = 1$.

(4) This has been already shown in (1), where it has been shown that there are numbers v_1, v_2, v_3, $v^2 = 1$ that will satisfy this condition.

Exercise 2.73 The ρ is a density operator. Because this is a positive operator, we can write it as

$$\rho = \sum_i \lambda_i |i\rangle\langle i| = \sum_i |\tilde{i}\rangle\langle\tilde{i}| \ , \ \sqrt{\lambda_i}|i\rangle = |\tilde{i}\rangle, \tag{140}$$

$|i\rangle$ are the orthonormal eigenvectors of ρ. The support of ρ is the space spanned by the vectors $|i\rangle$, where $\lambda_i \neq 0$. A general state vector in this support space is therefore

$$|\psi\rangle = \sum_{\substack{j \\ \lambda_j \neq 0}} \psi_j |j\rangle \ , \ \langle\psi|\psi\rangle = 1. \tag{141}$$

We have to show, that there exist a set of $|\psi_i\rangle$ vectors where the number of elements are the same as the rank of ρ, and that $|\psi_i\rangle$ vectors form $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$, and for one of $|\psi_i\rangle$ vectors it is true, that $|\psi_i\rangle = |\psi\rangle$.

The rank of ρ is the linearly independent columns of the matrix form of ρ. ρ can be diagonalized, and in the diagonalized form it has as many linearly independent columns as eigenvectors which have nonzero corresponding eigenvalue.

Let's say, that $|\psi\rangle = |\psi_1\rangle$. Is there a vector u_j such that

$$|\tilde{\psi}\rangle = \sum_{\substack{j \\ \lambda_j \neq 0}} u_j |\tilde{j}\rangle, \tag{142}$$

with $\sum_j |u_j|^2 = 1$, and $|\tilde{\psi}\rangle = p|\psi\rangle$ for some $p \leq 1$? If there is, then we can extend u_j to an u_{ij} ($\lambda_i \neq 0$) unitary operator such that $|\tilde{\psi}_i\rangle = \sum_j u_{ij}|\tilde{j}\rangle$, which means, that $\rho = \sum_i |\tilde{\psi}_i\rangle\langle\tilde{\psi}_i|$ (because of Theorem 2.6). Because there are same number of $|\tilde{\psi}_i\rangle$ vectors as the rank of ρ, the normalization of the $|\tilde{\psi}_i\rangle$ vectors are the minimal ensemble, and it contains our chosen $|\psi\rangle$.

Let's chose

$$u_j = \frac{\psi_j}{\sqrt{n\lambda_j}} \ , \ n = \sum_{\substack{j \\ \lambda_j \neq 0}} \frac{|\psi_j|^2}{\lambda_j}. \tag{143}$$

Then

$$|\tilde{\psi}\rangle = \sum_{\substack{j \\ \lambda_j \neq 0}} \frac{\psi_j}{\sqrt{n\lambda_j}} |\tilde{j}\rangle = \frac{1}{\sqrt{n}} \sum_{\substack{j \\ \lambda_j \neq 0}} \psi_j |j\rangle = \frac{1}{\sqrt{n}} |\psi\rangle,$$

$$\sum_{\substack{j \\ \lambda_j \neq 0}} |u_j|^2 = \frac{1}{n} \sum_{\substack{j \\ \lambda_j \neq 0}} \frac{|\psi_j|^2}{\lambda_j} = 1. \tag{144}$$

We can also see, that $n \geq 1$, because $\lambda_j \leq 1$, so $n = \sum_j |\psi_j|^2/\lambda_j \geq \sum_j |\psi_j|^2 = 1$. We can read off that $p = 1/\sqrt{n}$. Consequently $p \leq 1$.

Let's calculate $\langle\psi_i|\rho^{-1}|\psi_i\rangle$.

$$\langle\psi_i|\rho^{-1}|\psi_i\rangle = \frac{1}{p_i}\langle\tilde{\psi}_i|\rho^{-1}|\tilde{\psi}_i\rangle = \frac{1}{p_i}\sum_{jk} u^*_{ji}\sqrt{\lambda_j}u_{ik}\sqrt{\lambda_k}\langle j|\rho^{-1}|k\rangle =$$

$$= \frac{1}{p_i}\sum_{jk} u^*_{ji}u_{ik}\sqrt{\lambda_j}\sqrt{\lambda_k}\underbrace{\langle j|\lambda_k^{-1}|k\rangle}_{\lambda_j^{-1}\delta_{jk}} = \frac{1}{p_i}\underbrace{\sum_j u^*_{ji}u_{ij}}_{=1} = \frac{1}{p_i}. \tag{145}$$

Consequently, $p_i = 1/\langle\psi_i|\rho^{-1}|\psi_i\rangle$.

2.4.3 The reduced density operator

Exercise 2.74 The whole system is in one state only, so that one state's probability is one.

$$\rho^{AB} = |a \otimes b\rangle\langle a \otimes b|. \tag{146}$$

The reduced density operator for system A is therefore

$$\rho^A = \mathrm{tr}_B(\rho^{AB}) = |a\rangle\langle a|\mathrm{tr}(|b\rangle\langle b|) = |a\rangle\langle a|\underbrace{\langle b|b\rangle}_{=1} = |a\rangle\langle a|. \tag{147}$$

Exercise 2.75

$$\begin{aligned}
2\rho^1_{00} &= \mathrm{tr}_1((|00\rangle + |11\rangle)(\langle 00| + \langle 11|)) = \\
&= \mathrm{tr}_1(|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|) = \\
&= \langle 0|0\rangle|0\rangle\langle 0| + \langle 1|0\rangle\langle 0|1\rangle + \langle 0|1\rangle|1\rangle\langle 0| + \langle 1|1\rangle|1\rangle\langle 1| = \\
&= |0\rangle\langle 0| + |1\rangle\langle 1| = I, \\
2\rho^1_{01} &= \mathrm{tr}_1((|00\rangle - |11\rangle)(\langle 00| - \langle 11|)) = \\
&= \mathrm{tr}_1(|00\rangle\langle 00| - |00\rangle\langle 11| - |11\rangle\langle 00| + |11\rangle\langle 11|) = \\
&= \langle 0|0\rangle|0\rangle\langle 0| - \langle 1|0\rangle\langle 0|1\rangle - \langle 0|1\rangle|1\rangle\langle 0| + \langle 1|1\rangle|1\rangle\langle 1| = \\
&= |0\rangle\langle 0| + |1\rangle\langle 1| = I, \\
2\rho^1_{10} &= \mathrm{tr}_1((|01\rangle + |10\rangle)(\langle 01| + \langle 10|)) = \\
&= \mathrm{tr}_1(|01\rangle\langle 01| + |01\rangle\langle 10| + |10\rangle\langle 01| + |10\rangle\langle 10|) = \\
&= \langle 0|0\rangle|1\rangle\langle 1| + \langle 1|0\rangle\langle 1|0\rangle + \langle 0|1\rangle|0\rangle\langle 1| + \langle 1|1\rangle|0\rangle\langle 0| = \\
&= |1\rangle\langle 1| + |0\rangle\langle 0| = I, \\
2\rho^1_{11} &= \mathrm{tr}_1((|01\rangle - |10\rangle)(\langle 01| - \langle 10|)) = \\
&= \mathrm{tr}_1(|01\rangle\langle 01| - |01\rangle\langle 10| - |10\rangle\langle 01| + |10\rangle\langle 10|) = \\
&= \langle 0|0\rangle|1\rangle\langle 1| - \langle 1|0\rangle\langle 1|0\rangle - \langle 0|1\rangle|0\rangle\langle 1| + \langle 1|1\rangle|0\rangle\langle 0| = \\
&= |1\rangle\langle 1| + |0\rangle\langle 0| = I.
\end{aligned} \tag{148}$$

Moreover, the tr_2 of each of these is the same. It is because for all states it is true, that the switch $1 \leftrightarrow 2$ will only multiply the state with a ± 1, so that $|xy\rangle = (\pm 1)|yx\rangle$, and then $|xy\rangle\langle xy| = |yx\rangle\langle yx|$.

2.5 The Schmidt decomposition and purifications

Exercise 2.76 Let's suppose, that there are m vectors in the orthonormal basis system for system A, and n for system B, and that $m > n$.

$$|\psi\rangle = \underbrace{\sum_{j=1}^{n}\sum_{k=1}^{n} a_{jk}|j\rangle|k\rangle}_{|\psi_n\rangle} + \sum_{j=n+1}^{m}\sum_{k=1}^{n} a_{jk}|j\rangle|k\rangle. \tag{149}$$

Because of theorem 2.7, we can write that

$$|\psi_n\rangle = \sum_{i=1}^{n} \lambda_i |i_A\rangle|i_B\rangle, \tag{150}$$

and because $|i_B\rangle = \sum_k v_{ik}|k\rangle$, where v is unitary

$$|k\rangle = \sum_{i=1}^{n} v^*_{ki}|i_B\rangle. \tag{151}$$

Therefore

$$|\psi\rangle = \sum_{i=1}^{n} \underbrace{\left(\lambda_i |i_A\rangle + \sum_{j=n+1}^{m} \sum_{k=1}^{n} a_{jk} v_{ki}^* |j\rangle \right)}_{|i'_A\rangle} |i_B\rangle, \tag{152}$$

where I introduced $|i'_A\rangle$. Let's make an orthonormal system $|i''_A\rangle$ from $|i'_A\rangle$, so that

$$|i'_A\rangle = \sum_j a'_{ij} |j''_A\rangle. \tag{153}$$

Let's use the Gram–Schmidt procedure. We can make an orthonormal system from the vectors $|i'_A\rangle$ where $\lambda_i \neq 0$ by iteratively "adding" the vectors $|i'_A\rangle$ one after another, since those are linearly independent. With the vectors where $\lambda_i = 0$ we try to first see whether it's a vector which is linearly independent to the rest. If it is, we use again the Gram–Schmidt procedure. If it is not, then we can certainly find a'_{ij} such that the last equation is satisfied, and let's define $|i''_A\rangle = |i'_A\rangle$ which is guaranteed to be linearly independent to the rest.

All in all this means, that there is an orthonormal basis $|j''_A\rangle$ and complex numbers a'_{ij} such that

$$|\psi\rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} a'_{ij} |j''_A\rangle |i_B\rangle, \tag{154}$$

and then we can use theorem 2.7 again, because now a'_{ij} is square.

Exercise 2.77 Observe the following state, with each component being a 2 dimensional substate.

$$|\psi\rangle = \frac{1}{\sqrt{2}} |000\rangle + \frac{1}{\sqrt{2}} |011\rangle. \tag{155}$$

Let's prove by contradiction, that it is not possible to write this in the form

$$|\psi\rangle = \lambda_0 |0'0'0'\rangle + \lambda_1 |1'1'1'\rangle, \tag{156}$$

where λ_0 and λ_1 are nonnegative numbers, and $|0'\rangle$, $|1'\rangle$ are orthonormal basis for the respective systems. Note, that then $\lambda_0^2 + \lambda_1^2 = 1$. The most general form we can write any of the three subsystem is

$$\begin{aligned} |0'\rangle_i &= e^{i\delta_{0i}} (\cos\varphi_i |0\rangle - e^{i\phi_i} \sin\varphi_i |1\rangle), \\ |1'\rangle_i &= e^{i\delta_{1i}} (\sin\varphi_i |0\rangle + e^{i\phi_i} \cos\varphi_i |1\rangle). \end{aligned} \tag{157}$$

All in all, this means, that in general we can write

$$\begin{gathered} \lambda_0 |0'0'0'\rangle = \\ = \lambda_0 e^{i\delta_0} (\cos\varphi_1 |0\rangle - e^{i\phi_1} \sin\varphi_1 |1\rangle)(\cos\varphi_2 |0\rangle - e^{i\phi_2} \sin\varphi_2 |1\rangle)(\cos\varphi_3 |0\rangle - e^{i\phi_3} \sin\varphi_3 |1\rangle), \\ \lambda_1 |1'1'1'\rangle = \\ = \lambda_1 e^{i\delta_1} (\sin\varphi_1 |0\rangle + e^{i\phi_1} \cos\varphi_1 |1\rangle)(\sin\varphi_2 |0\rangle + e^{i\phi_2} \cos\varphi_2 |1\rangle)(\sin\varphi_3 |0\rangle + e^{i\phi_3} \cos\varphi_3 |1\rangle). \end{gathered} \tag{158}$$

Let's calculate some components which are 0.

$$0 = \langle 001|\psi\rangle = -\lambda_0 e^{i(\delta_0+\phi_3)}\cos\varphi_1\cos\varphi_2\sin\varphi_3 + \lambda_1 e^{i(\delta_1+\phi_3)}\sin\varphi_1\sin\varphi_2\cos\varphi_3, \tag{159}$$

$$0 = \langle 010|\psi\rangle = -\lambda_0 e^{i(\delta_0+\phi_2)}\cos\varphi_1\sin\varphi_2\cos\varphi_3 + \lambda_1 e^{i(\delta_1+\phi_2)}\sin\varphi_1\cos\varphi_2\sin\varphi_3, \tag{160}$$

$$0 = \langle 101|\psi\rangle = \lambda_0 e^{i(\delta_0+\phi_1+\phi_3)}\sin\varphi_1\cos\varphi_2\sin\varphi_3 + \lambda_1 e^{i(\delta_1+\phi_1+\phi_3)}\cos\varphi_1\sin\varphi_2\cos\varphi_3, \tag{161}$$

$$0 = \langle 110|\psi\rangle = \lambda_0 e^{i(\delta_0+\phi_1+\phi_2)}\sin\varphi_1\sin\varphi_2\cos\varphi_3 + \lambda_1 e^{i(\delta_1+\phi_1+\phi_2)}\cos\varphi_1\cos\varphi_2\sin\varphi_3. \tag{162}$$

Let's multiply (159) with $-e^{i(-\delta_0-\phi_3)}\cos\varphi_1$, and (161) with $e^{i(-\delta_0-\phi_1-\phi_3)}\sin\varphi_1$, and add the two equations together.

$$\lambda_0\cos\varphi_2\sin\varphi_3 = 0. \tag{163}$$

Let's multiply (160) with $e^{i(-\delta_1-\phi_2)}\sin\varphi_1$, and (162) with $e^{i(-\delta_1-\phi_1-\phi_2)}\cos\varphi_1$, and add the two equations together.

$$\lambda_1\cos\varphi_2\sin\varphi_3 = 0. \tag{164}$$

These last two equations say, that if $\cos\varphi_2\sin\varphi_3 \neq 0$, then both $\lambda_1 = \lambda_2 = 0$, which is not possible, since $|\psi\rangle \neq 0$. This means, that

$$\cos\varphi_2\sin\varphi_3 = 0. \tag{165}$$

Similarly, by multiplying (159) with $e^{i(-\delta_1-\phi_3)}\sin\varphi_1$ and multiplying (161) with $e^{i(-\delta_1-\phi_1-\phi_3)}\cos\varphi_1$ adding them together yields $\lambda_1\sin\varphi_2\cos\varphi_3 = 0$. Multiplying (160) with $-e^{i(-\delta_0-\phi_2)}\cos\varphi_1$ and multiplying (162) with $e^{i(-\delta_0-\phi_1-\phi_2)}\sin\varphi_1$ and adding them together yields $\lambda_0\sin\varphi_2\cos\varphi_3 = 0$. Also similarly, because both λ_0 and λ_1 cannot be zero simultenously, it is true, that

$$\sin\varphi_2\cos\varphi_3 = 0. \tag{166}$$

From these last two equations we can see, that if $\cos\varphi_2 = 0$, then $\sin\varphi_2 = \pm 1$, which means, that $\cos\varphi_3 = 0$. Similarly, if $\sin\varphi_3 = 0$, then $\sin\varphi_2 = 0$. The two possibilities are that

$$\cos\varphi_2 = 0, \sin\varphi_2 = \pm 1, \cos\varphi_3 = 0, \sin\varphi_3 = \pm 1,$$
$$\text{or}$$
$$\cos\varphi_2 = \pm 1, \sin\varphi_2 = 0, \cos\varphi_3 = \pm 1, \cos\varphi_3 = 0. \tag{167}$$

All in all, the two possible states for $|\psi\rangle$ are

$$|\psi\rangle = \lambda_0 e^{i\delta_0'}(\cos\varphi_1|0\rangle - e^{i\phi_1}\sin\varphi_1|0\rangle)|1\rangle|1\rangle + \lambda_1 e^{i\delta_1'}(\sin\varphi_1|0\rangle + e^{i\phi_1}\cos\varphi_1|0\rangle)|0\rangle|0\rangle,$$
$$\text{or}$$
$$|\psi\rangle = \lambda_0 e^{i\delta_0'}(\cos\varphi_1|0\rangle - e^{i\phi_1}\sin\varphi_1|0\rangle)|0\rangle|0\rangle + \lambda_1 e^{i\delta_1'}(\sin\varphi_1|0\rangle + e^{i\phi_1}\cos\varphi_1|0\rangle)|1\rangle|1\rangle. \tag{168}$$

This can only be $(|000\rangle/\sqrt{2} + |011\rangle/\sqrt{2})$ if $\cos\varphi_1 = 0$ and $\sin\varphi_1 = 0$ simultenously, which is impossible.

Exercise 2.78 If the state is a product state $|\psi\rangle|\phi\rangle$, then we can write $|0_A\rangle = |\psi\rangle$ and $|0_B\rangle = |\phi\rangle$. This Schmidt decomposition only has one nonzero λ_i, which is $\lambda_0 = 1$.

If a general $|\psi\rangle$ state has Schmidt number 1, then by definition there exist $|0_A\rangle$ and $|0_B\rangle$ normal states, such that we can Schmidt decompose this into $|\psi\rangle = |0_A\rangle|0_B\rangle$, which is a product state.

It's been proved in **Exercise 2.74** that if $|\psi\rangle$ is a product state, then it's reduced density operator is also pure.

ρ can be written as

$$\rho = |\psi\rangle\langle\psi|, \tag{169}$$

and because of Schmidt decomposition, $|\psi\rangle$ can be written as

$$|\psi\rangle = \sum_i \lambda_i |i_A i_B\rangle, \tag{170}$$

so that

$$\rho = \sum_{ij} \lambda_i \lambda_j |i_A i_B\rangle\langle j_A j_B|. \tag{171}$$

Calculating it's partial trace we get

$$\rho^A = \mathrm{tr}_B \rho = \sum_{ij} p_i \lambda_i \lambda_j |i_A\rangle\langle j_A| \underbrace{\langle i_B | j_B\rangle}_{\delta_{ij}} = \sum_i \lambda_i^2 |i_A\rangle\langle i_A|. \tag{172}$$

If ρ^A is pure, $|\rho^A\rangle = |a\rangle\langle a|$, and then

$$\langle j_A|a\rangle = \langle j_A|a\rangle\langle a|a\rangle = \langle j_A|\rho^A|a\rangle = \sum_i \lambda_i^2 \underbrace{\langle j_A|i_A\rangle}_{\delta_{ij}}\langle i_A|a\rangle = \langle j_A|a\rangle\lambda_j^2. \tag{173}$$

$\langle j_A|a\rangle \neq 0$, because then ρ^A would be 0. Let's divide by it.

$$1 = \lambda_i^2. \tag{174}$$

We know, that $\sum_i \lambda_i^2 = 1$, consequently, there can be only one λ, and $|\psi\rangle$ is pure.

Exercise 2.79

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}|0\rangle|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|1\rangle,$$

$$\frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2} = \underbrace{\frac{|0\rangle + |1\rangle}{\sqrt{2}}}_{|0'\rangle}\underbrace{\frac{|0\rangle + |1\rangle}{\sqrt{2}}}_{|0'\rangle} = 1\cdot|0'\rangle|0'\rangle + 0\cdot|1'\rangle|1'\rangle.$$

For the last problem, let's observe, that we can also write that expression in an other way.

$$\frac{|0_A\rangle|0_B\rangle + |0_A\rangle|1_B\rangle + |1_A\rangle|0_B\rangle}{\sqrt{3}} = \underbrace{\begin{bmatrix}|1_A\rangle & |0_A\rangle\end{bmatrix}}_{v_A^\dagger}\underbrace{\frac{1}{\sqrt{3}}\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}}_{A}\underbrace{\begin{bmatrix}|0_B\rangle\\ |1_B\rangle\end{bmatrix}}_{v_B}. \tag{175}$$

If we singular decompose A into $A = UDV$, where D is diagonal, and U, V unitary, then we can simply rewrite this into $(U^\dagger v_A)^\dagger \cdot D \cdot (V v_B)$, and this will be a Schmidt decomposed form, since D is diagonal. In **Exercise 2.50** we already polar decomposed the matrix $\sqrt{3}A$. $\sqrt{3}A = SJ$. We also know, that because J is positive, it can be written in the form $J = TDT^\dagger$, where D is diagonal, and T is unitary. Consequently, $U = ST$, $V = T^\dagger$, and the diagonal elements of D are the same as in **Exercise 2.50** divided by $\sqrt{3}$. T is going to be a matrix with elements constructed from the eigenvectors.

$$T = \begin{bmatrix} |+\rangle & |-\rangle \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{\sqrt{5-\sqrt{5}}} & \frac{\sqrt{2}}{\sqrt{5+\sqrt{5}}} \\ \frac{1}{\sqrt{2}}\frac{\sqrt{5}-1}{\sqrt{5-\sqrt{5}}} & -\frac{1}{\sqrt{2}}\frac{\sqrt{5}+1}{\sqrt{5+\sqrt{5}}} \end{bmatrix} = \sqrt{\frac{5+\sqrt{5}}{40}} \begin{bmatrix} 2 & \sqrt{5}-1 \\ \sqrt{5}-1 & -2 \end{bmatrix}. \tag{176}$$

Also, note, that we calculated S to be

$$S = \frac{1}{\sqrt{5}} \begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix}. \tag{177}$$

From this, we can calculate that

$$\begin{aligned} U^\dagger = (ST)^\dagger &= \sqrt{\frac{5+\sqrt{5}}{200}} \begin{bmatrix} 5-\sqrt{5} & 2\sqrt{5} \\ 2\sqrt{5} & \sqrt{5}-5 \end{bmatrix}, \\ V = T^\dagger &= \sqrt{\frac{5+\sqrt{5}}{40}} \begin{bmatrix} 2 & \sqrt{5}-1 \\ \sqrt{5}-1 & -2 \end{bmatrix}. \end{aligned} \tag{178}$$

From this

$$\begin{aligned} \begin{bmatrix} |0'_A\rangle \\ |1'_A\rangle \end{bmatrix} = U^\dagger \begin{bmatrix} |1_A\rangle \\ |0_A\rangle \end{bmatrix} &= \sqrt{\frac{5+\sqrt{5}}{200}} \begin{bmatrix} (5-\sqrt{5})|1_A\rangle + 2\sqrt{5}|0_A\rangle \\ 2\sqrt{5}|1_A\rangle + (\sqrt{5}-5)|0_A\rangle \end{bmatrix}, \\ \begin{bmatrix} |0'_B\rangle \\ |1'_B\rangle \end{bmatrix} = V \begin{bmatrix} |0_B\rangle \\ |1_B\rangle \end{bmatrix} &= \sqrt{\frac{5+\sqrt{5}}{40}} \begin{bmatrix} 2|0_B\rangle + (\sqrt{5}-1)|1_B\rangle \\ (\sqrt{5}-1)|0_B\rangle - 2|1_B\rangle \end{bmatrix}, \\ \lambda_+ = \frac{3+\sqrt{5}}{2\sqrt{3}}, \lambda_- &= \frac{3-\sqrt{5}}{2\sqrt{3}}. \end{aligned} \tag{179}$$

(the eigenvalues are from **Exercise 2.50**, divided by $\sqrt{3}$), and the Schmidt decomposition of the state is

$$\frac{|0_A\rangle|0_B\rangle + |0_A\rangle|1_B\rangle + |1_A\rangle|0_B\rangle}{\sqrt{3}} = \lambda_+|0'_A\rangle|0'_B\rangle + \lambda_-|1'_A\rangle|1'_B\rangle. \tag{180}$$

Exercise 2.80 If $|\psi\rangle$ and $|\varphi\rangle$ have the same Schmidt coefficients, then we can decompose them as

$$\begin{aligned} |\psi\rangle &= \sum_i \lambda_i |\psi_{A,i}\rangle|\psi_{B,i}\rangle, \\ |\varphi\rangle &= \sum_i \lambda_i |\varphi_{A,i}\rangle|\varphi_{B,i}\rangle. \end{aligned} \tag{181}$$

$|\psi_{A,i}\rangle$ and $|\varphi_{A,i}\rangle$ is an orthonormal basis for A, and similarly $|\psi_{B,i}\rangle$ and $|\varphi_{B,i}\rangle$ is an orthonormal basis for B. Let's define U and V as

$$U = \sum_i |\psi_{A,i}\rangle\langle\varphi_{A,i}|,$$
$$V = \sum_i |\psi_{B,i}\rangle\langle\varphi_{B,i}|. \tag{182}$$

It is easy to see, that U and V are unitary, since it's an outer product of an orthonormal basis system. Then, it is true that

$$(U \otimes V)|\varphi\rangle = \sum_i \lambda_i (U|\varphi_{A,i}\rangle)(V|\varphi_{B,i}\rangle) = \sum_i \lambda_i |\psi_{A,i}\rangle|\psi_{B,i}\rangle = |\psi\rangle. \tag{183}$$

Exercise 2.81 The two purifications are

$$|AR_1\rangle = \sum_i \sqrt{p_i}|i^A\rangle|i^{R_1}\rangle,$$
$$|AR_2\rangle = \sum_i \sqrt{p_i}|i^A\rangle|i^{R_2}\rangle. \tag{184}$$

Because $|i^{R_1}\rangle$ and $|i^{R_2}\rangle$ are orthonormal basis systems, the following U_R operator is unitary.

$$U_R = \sum_i |i^{R_1}\rangle\langle i^{R_2}|. \tag{185}$$

Then, it is true, that

$$(I_A \otimes U_R)|AR_2\rangle = \sum_i \sqrt{p_i}(I_A|i^A\rangle)(U_R|i^{R_2}\rangle) = \sum_i \sqrt{p_i}|i^A\rangle|i^{R_1}\rangle = |AR_1\rangle. \tag{186}$$

Exercise 2.82

(1) $|AR\rangle$ is a purification of ρ if $\rho = \mathrm{tr}_R(|AR\rangle\langle AR|)$. $|i\rangle$ is an orthonormal basis for R.

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|, \tag{187}$$
$$|AR\rangle = \sum_i \sqrt{p_i}|\psi_i\rangle|i\rangle. \tag{188}$$

Then, it is true, that

$$\mathrm{tr}_R(|AR\rangle\langle AR|) = \sum_{ij} \sqrt{p_i p_j}\mathrm{tr}_R(|\psi_i\rangle|i\rangle\langle\psi_j|\langle j|) =$$
$$= \sum_{ij} \sqrt{p_i p_j}|\psi_i\rangle\langle\psi_j| \underbrace{\mathrm{tr}_R(|i\rangle\langle j|)}_{\delta_{ij}} = \sum_i p_i |\psi_i\rangle\langle\psi_i| = \rho. \tag{189}$$

(2) Let's define the projective measurement operator corresponding to measuring $|i\rangle$ in system R, and leaving the system A as it is.

$$P_i = I_A \otimes |i\rangle\langle i|. \tag{190}$$

Then, the probability of obtaining this result is $q_i = \langle RA|P_i|RA\rangle$ and the system will be in state $P_i|RA\rangle/\sqrt{q_i}$, which is

$$q_i = \langle RA|P_i|RA\rangle = \sum_{jk} \sqrt{p_j p_k}\langle\psi_j|I_A|\psi_k\rangle \underbrace{\langle j|i\rangle}_{\delta_{ji}} \underbrace{\langle i|k\rangle}_{\delta_{ik}} = p_i\langle\psi_i|\psi_i\rangle = p_i,$$

$$\frac{P_i|RA\rangle}{\sqrt{q_i}} = \frac{1}{\sqrt{p_i}} \sum_j \sqrt{p_j}(I_A|\psi_j\rangle) \underbrace{(|i\rangle\langle i|j\rangle)}_{|i\rangle\delta_{ij}} = |\psi_i\rangle|i\rangle. \tag{191}$$

(3) [2] $|AR\rangle$ is composed of two system, which is Schmidt decomposable.

$$|AR\rangle = \sum_i \lambda_i |\varphi_i^A\rangle|\varphi_i^R\rangle, \tag{192}$$

where $\langle\varphi_i^A|\varphi_j^A\rangle = \delta_{ij}$ and $\langle\varphi_i^R|\varphi_j^R\rangle = \delta_{ij}$. It is also true, that $|AR\rangle$ is a purification of ρ for the system A, which means, that

$$\sum_i p_i|\psi_i\rangle\langle\psi_i| = \rho = \mathrm{tr}_R(|AR\rangle\langle AR|) =$$

$$= \mathrm{tr}_R\left(\sum_{ij} \lambda_i\lambda_j |\varphi_i^A\rangle|\varphi_i^R\rangle\langle\varphi_j^A|\langle\varphi_j^R|\right) = \sum_{ij} \lambda_i\lambda_j |\varphi_i^A\rangle\langle\varphi_j^R| \underbrace{\mathrm{tr}(|\varphi_i^R\rangle\langle\varphi_j^R|)}_{\delta_{ij}} =$$

$$= \sum_i \lambda_i^2 |\varphi_i^A\rangle\langle\varphi_i^A|. \tag{193}$$

This is a density operator, so we can use Theorem 2.6: there is a unitary matrix u_{ij} such that

$$\lambda_i|\varphi_i^A\rangle = \sum_j u_{ij}\sqrt{p_j}|\psi_j\rangle. \tag{194}$$

Let's substitute this into the Schmidt decomposition of $|AR\rangle$.

$$|AR\rangle = \sum_i \sum_j u_{ij}\sqrt{p_j}|\psi_j\rangle|\varphi_i^R\rangle =$$

$$= \sum_j \sqrt{p_j}|\psi_j\rangle \underbrace{\sum_i u_{ij}|\varphi_i^R\rangle}_{:=|j^R\rangle} = \sum_i \sqrt{p_i}|\psi_i\rangle|i^R\rangle, \tag{195}$$

where I introduced the states $|i^R\rangle$. Let's prove, that this is an orthonormal basis.

$$\langle i^R|j^R\rangle = \sum_{kl} (u^*)_{ik}u_{jl} \underbrace{\langle\varphi_k^R|\varphi_l^R\rangle}_{\delta_{kl}} = \sum_k u_{jk}(u^\dagger)_{ki} = \delta_{ij}. \tag{196}$$

$|i^R\rangle$ is an orthonormal system on R, and $|AR\rangle = \sum_i \sqrt{p_i}|\psi_i\rangle|i^R\rangle$ is just the usual purification for the system A, and we can use the results in (2) to prove, that the system A will be in state $|\psi_i\rangle$ with probability p_i.

3 Introduction to computer science

3.1 Models for computation

3.1.1 Turing machines

Exercise 3.1 A Turing machine's initial and final state are restricted. We have to find a function which has input and output domain's which are not possible for a Turing machine. The whole program can only change finite number of tape squares, because the machine runs a finite number of steps. That means, that for example the machine will not generate the digits of π in binary format on the tape if the tape initially had only finite number of squares which are nonblank. Nature – as far as we know now – deals with continuous real numbers, and has processes which are analog, not digital. So a general analog machine is not possible on a Turing machine, but might be possible using other physical machines.

Exercise 3.2 The program can be given a unique number in the following way. Each state q can be given a number, and each x symbol can also be given a number. Each program line $\langle q, x, q', x', s\rangle$ can be coded using numbers, using a formula

$$l = p_1^q\ p_2^x\ p_3^{q'}\ p_4^{x'}\ p_5^s. \tag{197}$$

Then, the whole program can be coded into one number using the program lines coded (using the previous formula).

$$p = p_1^{l_1}\ p_2^{l_2}\ \ \ldots\ p_n^{l_n}. \tag{198}$$

The machine's initial tape can also be given a number. If the symbols on the tape initially are $s_1, s_2, \ldots s_n$, then the tape's state's unique number is

$$t = p_1^{s_1}\ p_2^{s_2}\ \ \ldots\ p_n^{s_n}. \tag{199}$$

All in all, the whole machine can have the unique number

$$p_1^p\ p_2^t. \tag{200}$$

Exercise 3.3 Let's use a single tape machine. The algorithm will first start out as

$$\triangleright\texttt{1010001011101bbb}\ldots$$

with the symbol marking the left side of the tape, the number that has to be reversed, and the `b` blank symbols all the way through. Next, the program will put an `X` at the end of the number

$$\triangleright\texttt{1010001011101Xbb}\ldots$$

marking the end of the number. Then, the program will copy each bit one by one in reversed order after the `X`, and leaving `b` at the original place where the bit was already copied.

$$\triangleright\texttt{1010001011101Xbb}\ldots$$
$$\triangleright\texttt{101000101110bX1b}\ldots$$
$$\triangleright\texttt{10100010111bbX10}\ldots$$
$$\ldots$$

If there are no bits left to reverse copy, the machine will copy back the reversed bits to the beginning, before the `X`. After it's done, the machine will override the `X` with the blank symbol, and halts.

Let's construct this program. First, the program will put an `X` at the end. Since the tape head starts from the left side of the tape, we have to only find the first blank symbol, and put an `X` there.

$$\langle \text{start}, \triangleright, \text{search end to put X}, \triangleright, +1\rangle,$$
$$\langle \text{search end to put X}, \texttt{0}, \text{search end to put X}, \texttt{0}, +1\rangle$$
$$\langle \text{search end to put X}, \texttt{1}, \text{search end to put X}, \texttt{1}, +1\rangle$$
$$\langle \text{search end to put X}, \texttt{b}, \text{find to reverse copy}, \texttt{X}, -1\rangle$$

Next, we find the next symbol to reverse copy. If found, we head over after the `X`, and copy it. We also have to distinguish which symbol we copy, and store it in the state whether it's a `0` or a `1` we wish to reverse copy. And move that bit until `X`.

$$\langle \text{find to reverse copy}, \texttt{b}, \text{find to reverse copy}, \texttt{X}, -1\rangle$$
$$\langle \text{find to reverse copy}, \texttt{0}, \text{move reverse copy before X } 0, \texttt{b}, +1\rangle$$
$$\langle \text{find to reverse copy}, \texttt{1}, \text{move reverse copy before X } 1, \texttt{b}, +1\rangle$$
$$\langle \text{move reverse copy before X } 0, \texttt{b}, \text{move reverse copy before X } 0, \texttt{b}, +1\rangle$$
$$\langle \text{move reverse copy before X } 0, \texttt{X}, \text{move reverse copy after X } 0, \texttt{X}, +1\rangle$$
$$\langle \text{move reverse copy before X } 1, \texttt{b}, \text{move reverse copy before X } 1, \texttt{b}, +1\rangle$$
$$\langle \text{move reverse copy before X } 1, \texttt{X}, \text{move reverse copy after X } 1, \texttt{X}, +1\rangle$$

After moving the bit until `X`, the bit will be moved further, and stored at the end.

$$\langle \text{move reverse copy after X } 0, \texttt{0}, \text{move reverse copy after X } 0, \texttt{0}, +1\rangle$$
$$\langle \text{move reverse copy after X } 0, \texttt{1}, \text{move reverse copy after X } 0, \texttt{1}, +1\rangle$$
$$\langle \text{move reverse copy after X } 0, \texttt{b}, \text{move to reverse copy after X}, \texttt{0}, -1\rangle$$
$$\langle \text{move reverse copy after X } 1, \texttt{0}, \text{move reverse copy after X } 1, \texttt{0}, +1\rangle$$
$$\langle \text{move reverse copy after X } 1, \texttt{1}, \text{move reverse copy after X } 1, \texttt{1}, +1\rangle$$
$$\langle \text{move reverse copy after X } 1, \texttt{b}, \text{move to reverse copy after X}, \texttt{1}, -1\rangle$$

After the copying happened, let's move the head before `X`, and then we can start the process all over again.

$$\langle \text{move to reverse copy after X}, \texttt{0}, \text{move to reverse copy after X}, \texttt{0}, -1\rangle$$
$$\langle \text{move to reverse copy after X}, \texttt{1}, \text{move to reverse copy after X}, \texttt{1}, -1\rangle$$
$$\langle \text{move to reverse copy after X}, \texttt{X}, \text{find to reverse copy}, \texttt{X}, -1\rangle$$

We have to make sure, that the process stops after every bit is reverse copied. If it did, then we can start the normal copying process, which is to move to the first bit after `X`, copy that to the last nonblank place to the left.

$$\langle \text{find to reverse copy}, \triangleright, \text{find bit to copy back}, \triangleright, +1\rangle$$
$$\langle \text{find bit to copy back}, \texttt{b}, \text{find bit to copy back}, \texttt{b}, +1\rangle$$
$$\langle \text{find bit to copy back}, \texttt{X}, \text{find bit to copy back}, \texttt{X}, +1\rangle$$

When we find the bit to copy back, store it in the state, and find the leftmost 0 or 1, because we have to copy right next to it.

$$\langle \text{find to reverse copy}, \texttt{0}, \text{move bit to copy back } 0, \texttt{b}, -1\rangle$$
$$\langle \text{move bit to copy back } 0, \texttt{b}, \text{move bit to copy back } 0, \texttt{b}, -1\rangle$$
$$\langle \text{move bit to copy back } 0, \texttt{X}, \text{move bit to copy back } 0, \texttt{X}, -1\rangle$$
$$\langle \text{move bit to copy back } 0, \texttt{0}, \text{copy back } 0, \texttt{0}, +1\rangle$$
$$\langle \text{move bit to copy back } 1, \texttt{0}, \text{copy back } 0, \texttt{1}, +1\rangle$$
$$\langle \text{find to reverse copy}, \texttt{1}, \text{move bit to copy back } 1, \texttt{b}, -1\rangle$$
$$\langle \text{move bit to copy back } 1, \texttt{b}, \text{move bit to copy back } 1, \texttt{b}, -1\rangle$$
$$\langle \text{move bit to copy back } 1, \texttt{X}, \text{move bit to copy back } 1, \texttt{X}, -1\rangle$$
$$\langle \text{move bit to copy back } 1, \texttt{0}, \text{copy back } 1, \texttt{0}, +1\rangle$$
$$\langle \text{move bit to copy back } 1, \texttt{1}, \text{copy back } 1, \texttt{1}, +1\rangle$$

We can now copy back the bit, and then start the search for the new bit to copy, but before that, we have to find out whether there are any more bits to copy back. If we have just filled the space between the start symbol and the `X`, there are no more bits, we can override `X` to `b`. If we didn't, there are more, we start again the copy back procedure.

$$\langle \text{copy back } 0, \texttt{b}, \text{are there more bits to copy back?}, \texttt{0}, +1\rangle$$
$$\langle \text{copy back } 1, \texttt{b}, \text{are there more bits to copy back?}, \texttt{1}, +1\rangle$$
$$\langle \text{are there more bits to copy back?}, \texttt{b}, \text{find bit to copy back}, \texttt{b}, +1\rangle$$
$$\langle \text{are there more bits to copy back?}, \texttt{X}, \text{halt}, \texttt{b}, 0\rangle$$

Exercise 3.4 Let's assume a double tape now. We also assume, that the two numbers are the same length. Initially the tapes are in the following state.

$$\triangleright\texttt{111000b101010b}\ldots$$
$$\triangleright\texttt{bbbbbbbbbbbbbb}\ldots$$

First, let's copy the first number to the second tape. Then, we will move the first tape head to the second number's first digit, and the second tape head to the beginning of the second tape. Then, we add modulo 2 the digits one by one. This is a modulo 2 addition, there is no remainder (this is the xor operation).

$$\triangleright\texttt{111000b101010b}\ldots$$
$$\triangleright\texttt{111000bbbbbbbb}\ldots$$

$$\triangleright\texttt{111000b101010b}\ldots$$
$$\triangleright\texttt{010010bbbbbbbb}\ldots$$

Let's construct the program. First, we copy each bit one by one from the first number to the second tape.

$$\langle \text{start}, \triangleright, \triangleright, \text{first copy}, \triangleright, \triangleright, +1, +1\rangle$$
$$\langle \text{first copy}, \mathtt{0}, \mathtt{b}, \text{first copy}, \mathtt{0}, \mathtt{0}, +1, +1\rangle$$
$$\langle \text{first copy}, \mathtt{1}, \mathtt{b}, \text{first copy}, \mathtt{1}, \mathtt{1}, +1, +1\rangle$$
$$\langle \text{first copy}, \mathtt{b}, \mathtt{b}, \text{second head back}, \mathtt{b}, \mathtt{b}, 0, -1\rangle$$

Let's move the second tape head back, and then the first tape head to the right one, to the beginning of the second number.

$$\langle \text{second head back}, \mathtt{b}, \mathtt{0}, \text{second head back}, \mathtt{b}, \mathtt{0}, 0, -1\rangle$$
$$\langle \text{second head back}, \mathtt{b}, \mathtt{1}, \text{second head back}, \mathtt{b}, \mathtt{1}, 0, -1\rangle$$
$$\langle \text{second head back}, \mathtt{b}, \triangleright, \text{first head start to second number}, \mathtt{b}, \triangleright, 0, +1\rangle$$
$$\langle \text{first head start to second number}, \mathtt{b}, \mathtt{0}, \text{xor}, \mathtt{b}, \mathtt{0}, +1, 0\rangle$$
$$\langle \text{first head start to second number}, \mathtt{b}, \mathtt{1}, \text{xor}, \mathtt{b}, \mathtt{1}, +1, 0\rangle$$

Now, we can start the xor operation on both numbers one by one.

$$\langle \text{xor}, \mathtt{0}, \mathtt{0}, \text{xor}, \mathtt{0}, \mathtt{0}, +1, +1\rangle$$
$$\langle \text{xor}, \mathtt{0}, \mathtt{1}, \text{xor}, \mathtt{0}, \mathtt{1}, +1, +1\rangle$$
$$\langle \text{xor}, \mathtt{1}, \mathtt{0}, \text{xor}, \mathtt{0}, \mathtt{1}, +1, +1\rangle$$
$$\langle \text{xor}, \mathtt{1}, \mathtt{1}, \text{xor}, \mathtt{0}, \mathtt{0}, +1, +1\rangle$$
$$\langle \text{xor}, \mathtt{b}, \mathtt{b}, \text{halt}, \mathtt{b}, \mathtt{b}, 0, 0\rangle$$

Exercise 3.5 Let's assume, that there is an algorithm to decide whether M halts if M is given a blank tape $\texttt{HALT}(M)$. This algorithm will return 1 if it halts, and 0 when it does not halt. Then, we can create an algorithm $\texttt{TURING}(M)$ (similarly to the algorithm in Box 3.2), that will forever loop if M halts, and halts if M does not halt. Let's call denote this algorithm with T. Then, when we try to calculate $\texttt{TURING}(T)$, we get a contradiction.

Exercise 3.6 Let's suppose, that the algorithm which outputs $h_p(x)$ with probability ($p > 1/2$) of correctness greater that $1/2$ is called $\texttt{HALT}(x)$. Now, let's create an other algorithm (similar to the previous exercise and the algorithm in Box 3.2): $\texttt{T}(x)$ will create a loop forever, if $\texttt{HALT}(x)$ returns with 1, and $\texttt{T}(x)$ will halt, if $\texttt{HALT}(x)$ returns with 0.

Let's try to find out what will happen if we apply $\texttt{T}$ to $\texttt{T}$. If the previous assumptions are correct, we will arrive at a contradiction.

First, let's assume, that $\texttt{T}(\texttt{T})$ will halt with a probability $q \geq 1/2$. This means, that $h_p(\texttt{T}) = 1$, and then $\texttt{HALT}(\texttt{T})$ should return 1 with probability $p > 1/2$, and return with 0 with probability $(1 - p) < 1/2$. Next, let's calculate the probability that $\texttt{T}(\texttt{T})$ halts using the given algorithm for it. It halts, if $\texttt{HALT}(\texttt{T})$ returns 0, and following the arguments in the last sentence, it happens with probability $(1 - p) < 1/2$. This is a contradiction: it cannot halt with probability $q \geq 1/2$, and with probability $(1 - p) < 1/2$ at the same time.

Next, let's assume that $\texttt{T}(\texttt{T})$ will halt with a probability $q < 1/2$. This means, that $h_p(\texttt{T}) = 0$, and then $\texttt{HALT}(\texttt{T})$ should return 0 with probability $p > 1/2$, and return with 1 with probability $(1 - p) < 1/2$. Following the same arguments as before, we can calculate, that $\texttt{T}(\texttt{T})$ halts with probability $p > 1/2$ (when $\texttt{HALT}(\texttt{T})$ is 0). This is again a contradiction, since it cannot halt with probability $q < 1/2$, and $p > 1/2$ at the same time.

Exercise 3.7 We can again, create a new `T` algorithm using this machine which will create a contradiction the same way as in previous exercises. Applying this `T` to the mentioned Turing machine aided by an oracle cannot give consistent answer.

3.1.2 Circuits

Exercise 3.8 `NOT` can be implemented by applying a `FANOUT` to the input wire, and plugging those wires in the `NAND`.

`AND` can be implemented by applying a `NAND` gate to the inputs, and then applying a `NOT` gate. This `NOT` gate can be implemented by `NAND` gate using the technique above.

`XOR` can be implemented by first applying a `NOT` gate to both A and B inputs to get $\overline{A}$ and $\overline{B}$. Next, let's apply a `NAND` to A and $\overline{B}$ to get C, and apply a `NAND` to B and $\overline{A}$ to get D. Clearly, C only 0 when $AB = 10$, otherwise it's 1, and D only 0 when $AB = 01$. Next, we apply a `NAND` to C and D. This output will be only 1 when C or D (or both) is 0, which only happens if $AB = 10$ or $AB = 01$.

3.2 The analysis of computation problems

3.2.1 How to quantify computational resources

Exercise 3.9 First, let's prove, that

$$f(n) \text{ is } O(g(n)) \Rightarrow g(n) \text{ is } \mathbf{\Omega}(f(n)). \tag{201}$$

If $f(n)$ is $O(g(n))$, then there exists an n_0, and c, such that for all $n > n_0$ it is true, that $f(n) \leq c \cdot g(n)$. This means, that there exist a $c' = 1/c$, and this same n_0, that for all $n > n_0$ it is true, that $c'f(n) \leq g(n)$, which means, that $g(n)$ is $\mathbf{\Omega}(f(n))$.

Now, let's prove the converse.

$$g(n) \text{ is } \mathbf{\Omega}(f(n)) \Rightarrow f(n) \text{ is } O(g(n)). \tag{202}$$

If $g(n)$ is $\mathbf{\Omega}(g(n))$, then there exists an n_0, and c, such that for all $n > n_0$ it is true, that $c \cdot f(n) \leq g(n)$. This means, that there exist a $c' = 1/c$, and this same n_0, that for all $n > n_0$ it is true, that $f(n) \leq c' \cdot g(n)$, which means, that $f(n)$ is $O(g(n))$.

Let's prove the last statement. We know because of the definition of $\mathbf{\Theta}$ and from the previous arguments, that

$$\begin{gathered}
f(n) \text{ is } \mathbf{\Theta}(g(n)) \Leftrightarrow f(n) \text{ is } O(g(n)) \text{ and } f(n) \text{ is } \mathbf{\Omega}(g(n)), \\
\text{and} \\
g(n) \text{ is } \mathbf{\Theta}(f(n)) \Leftrightarrow g(n) \text{ is } O(f(n)) \text{ and } g(n) \text{ is } \mathbf{\Omega}(f(n)), \\
\text{and} \\
f(n) \text{ is } O(g(n)) \Leftrightarrow g(n) \text{ is } \mathbf{\Omega}(f(n)), \\
\text{and} \qquad (203) \\
g(n) \text{ is } O(f(n)) \Leftrightarrow f(n) \text{ is } \mathbf{\Omega}(g(n)). \qquad (204)
\end{gathered}$$

From these, it is true then, that $f(n)$ is $\mathbf{\Theta}(g(n))$ if and only if $g(n)$ is $\mathbf{\Theta}(f(n))$.

Exercise 3.10 $g(n)$ is a k degree polynomial, it can be written in the following form.

$$g(n) = a_k \cdot n^k + a_{k-1} \cdot n^{k-1} + \ldots + a_0 = \\ = a_k \cdot n^k \underbrace{\left(1 + \frac{a_{k-1}}{a_k} \cdot n^{-1} + \ldots + \frac{a_0}{a_k} n^{-k}\right)}_{1+\epsilon_n}. \tag{205}$$

For a sufficiently large n it's true, that $\epsilon_n \leq 1$, and so there is an n_0, so that for every $n > n_0$

$$g(n) \leq \underbrace{2a_k}_{c} n^k = c \cdot \underbrace{n^{k-l}}_{\leq 1} n^l \leq c \cdot n^l, \tag{206}$$

where I assumed, that $k \leq l$. This means, that $g(n)$ is $O(n^l)$.

Exercise 3.11 [3] First of all, let's calculate the following using L'hopital rule (nominator and denominator goes to infinity, so we can use this rule). $k > 0$.

$$\lim_{n\to\infty} \frac{\log(n)}{n^k} = \frac{1}{\ln(2)} \lim_{n\to\infty} \frac{\ln(n)}{n^k} = \frac{1}{k\ln(2)} \lim_{n\to\infty} \frac{\frac{1}{n}}{n^{k-1}} = \frac{1}{k\ln(2)} \lim_{n\to\infty} \frac{1}{n^k} = 0. \tag{207}$$

From this, we conclude, that there is an n_0 such that for every $n > n_0$ it's true, that $\log(n) \leq n^k$. Consequently, $\log(n)$ is $O(n^k)$ for $k > 0$.

Exercise 3.12 Because $\log(n)$ grows without bounds, there is an n_0 such that for every $n > n_0$ it's true, that $\log(n) > k$ for any prefixed k, and then for these n: $n^k < n^{\log(n)}$. This mean, that n^k is $O(n^{\log(n)})$.

Also, because $\log(n)$ grows without bounds, for a fixed k there cannot be any n_0 so that for every $n > n_0$ it was true, that $n^k \geq n^{\log(n)}$. $n^{\log(n)}$ cannot be $O(n^k)$ for any k.

Exercise 3.13 We already saw in **Exercise 3.11**, that for any k there is an n_0 such that for every $n > n_0$: $\log(n) < n^k$. Note, that using the same argument we can prove, that there is an n_0, so that for every $n > n_0$ the following is true: $\log(n) < (\log(c)n)^k$. Let's fix $k = 1/2$. Then

$$\log(n) < (\log(c)n)^{1/2}. \tag{208}$$

Squaring this gives

$$(\log(n))^2 < \log(c)n. \tag{209}$$

Let's put this into the exponent of 2.

$$n^{\log(n)} = 2^{(\log(n))^2} < 2^{\log(c)n} = c^n. \tag{210}$$

We just concluded, that c^n is $\mathbf{\Omega}(n^{\log(n)})$.

This last equation holds regardless of c, and it even holds if we multiply any side with a constant for sufficiently large n_0. This means, that $n^{\log(n)}$ is never $\mathbf{\Omega}(c^n)$.

Exercise 3.14 There is an n_0 such that for every $n > n_0$: $e(n) \leq f(n)$. There is an n_1 such that for every $n > n_1$: $g(n) \leq h(n)$. Let's say, that $N \geq n_0$ and $N \geq n_1$. Then, for every $n > N$: $e(n) \cdot g(n) \leq f(n) \cdot h(n)$. There is such an N, and so, $e(n) \cdot g(n)$ is $O(f(n) \cdot h(n))$.

Exercise 3.15 [5] The compare-and-swap operation can leave the list as it is, or swap the elements. Applying k compare-and-swap operations on in each step, can only result in 2^k different reorderings at most.

If we want to sort all of the $n!$ possible lists, in the best case scenario (when all of the reorderings are different) we have to make K steps. In the best case scenario then $2^K = n!$. According to the Stirling formula $\log(n!) = n\log(n) - n\log(e) + \mathbf{\Theta}(\log(n))$. Then, in the theoretical maximum best case scenario the lower bound for the number of steps is $A \cdot n\log(n)$, where A is any number less than 1. To see this, consider for which A it holds, that

$$An\log(n) \leq f(n) = n\log(n) - n\log(e) = n(\log(n) - \log(e)). \tag{211}$$

Dividing both sides by $n\log(n)$ we get, that $A \leq 1 - \frac{\log(e)}{\log(n)}$. For sufficiently large n, the right hand side can get arbitrarily close to 1. So, if $A < 1$, then this equation holds, consequently, there is an A and n_0 such that for every $n > n_0$: $An\log(n) \leq f(n)$, which means, that $\mathbf{\Omega}(n\log(n))$ operations are needed for this algorithm.

3.2.2 Computational complexity

Exercise 3.16 Unsolved.

3.2.3 Decision problems and the complexity classes P and NP

Exercise 3.17 If a polynomial-time algorithm for finding the factors of a number m exists, we can use that algorithm, and then check if this list contains elements less than l in linear time. This will be polynomial-time too, and this would be the algorithm for the factoring decision problem.

Let's prove the converse of the statement. Let's say, that we have a polynomial-time algorithm for the factoring decision problem, let's say it's $O(n^k)$ (n is the number of digits m has, $n \sim \log(m)$). Then, we do a logarithmic search to find the smallest non-trivial factor. It's done by first finding out whether there's any non-trivial factor less than $m/2$. If there is, we ask whether there's any non-trivial factor less than $m/2 + m/4$. If there isn't, we ask whether there's any less than $m/4$. If there isn't, we ask whether there's any less than $m/4 + m/8$. If there is, we ask whether there's any less than $m/8$. And so on, until we find only one number d. This proves is $O(\log(m)n^k) = O(n^{k+1})$. Then, we divide m by this d, which is a polynomial process, and start the process all over again on this new number. All in all, we find all smallest non-trivial divisor, which is the prime factorization of m. How many times should we divide m? We should divide by the number of prime divisors m has. Let's say, that m has l prime divisors $(p_1, \ldots p_l)$. The possible smallest p is 2, so $2^l \leq p_1 \cdot \ldots \cdot p_l = m$, and so

$$l \leq \log(m). \tag{212}$$

That means, that the division process should be done less than $\log(m)$ times, and so finding the prime factorization of m is $O(\log(m)n^{k+1}) = O(n^{k+2})$, which is polynomial.

Exercise 3.18 [1] Let's assume, that $\mathbf{P} = \mathbf{NP}$.

Then, if a problem is **NP**, then it is **P**, and so we can create an algorithm, such that for any w witness string it will answer q_Y if $x \in L$, and answer q_N if $x \notin L$ in polynomial time. That means, that it's true for this new algorithm, that if $x \notin L$, there exists a witness string w such that it will answer q_N and if $x \in L$, it will answer q_Y for all witness string w. And so, this problem is **coNP**.

If a problem is **coNP**, then there exists an algorithm (M), such that if $x \in L$, for every witness string w the algorithm returns q_Y, and if $x \notin L$, there exists a witness string w such that the algorithm will return q_N. Let's ask the negation of the problem. The negation of the problem is **NP** problem, because we can create an algorithm from the original M such that it answers q_N if the original answered q_Y, and answers q_Y if the original answered q_N, which will answer the negated problem correctly, and is **NP**. But this is **P** by assumption. By negating the answer again, we answer the original problem polynomial in time. But by assumption $\mathbf{P} = \mathbf{NP}$, which again means, that this original **coNP** problem is **NP**.

Because of these

$$\mathbf{P} = \mathbf{NP} \Rightarrow \mathbf{coNP} = \mathbf{NP}. \tag{213}$$

We can use modus tollens to get

$$\mathbf{coNP} \neq \mathbf{NP} \Rightarrow \mathbf{P} \neq \mathbf{NP}. \tag{214}$$

Exercise 3.19 Let's start from vertex A, and go to any of it's adjacent vertex, and so on, until we find the specified endpoint vertex B, or until we find a vertex which only has edges which has not been in the current path. It the former is the case, vertex B is reachable. It the latter, we remove that vertex from the graph, and go back one step on the already existing path. Then, we use this same method until there is no vertex left, or B is found. At most, we go through n vertex, so this algorithm is $O(n)$.

The graph is connected if there is a path between all vertices, which is equivalent to the statement that any vertex is reachable for any specific – let's say A – vertex. (Because there if it's true, there is a path between any two vertices: B and C through A.) To see whether the graph is connected we can use the algorithm from the previous paragraph for A and every other $(n-1)$ vertex, which is $O(n^2)$.

Exercise 3.20 Let's define an Euler path as a path on the graph with vertices $v_1, v_2, \ldots, v_n$ $(v_i \neq v_{i+1})$, where every (v_i, v_{i+1}) is an edge, and each edge occurs only once.

Lemma: For any Euler path in a graph where every edge has an even number of vertices, and where $v_1 \neq v_n$, there is an other Euler path (an inverse path), for which $v_1' = v_n$ and $v_n' = v_1$, and the original path does not contain any of the edges of the inverse path.

To prove this, first, let's notice, that when I have an Euler path, v_1 and v_n has an odd number of edges in this path, and every other vertex (the middle ones) have an even number of edges in this path. When v_1 or v_2 occurs at the middle, it's "outgoing" and "ingoing" edges occur, which is even, and they are endpoints, which adds +1 edge to each of them.

Next, notice, that when I add a new vertex at the end of the path, $v_x \neq v_n$, this new vertex is either v_1 or this new vertex has an edge which is not part of the original path. This is because v_n has an odd number of edges in the path, and so it has at least one edge which is not part of the path, so we can add it. It leads to $v_x \neq v_n$. There are three possibilities: (1) $v_x = v_1$ or (2) v_x is part of the original path, but not v_1, or (3) v_x is a completely new vertex, not part of the original path. If (2) is the case, the original path contains even number of edges of v_x, but it has an other edge which we've just added: (v_n, v_x), and v_x cannot have odd number of edges, so there is an other edge we can use, which is not part of the original path, and is not the new edge. On the other hand, if (3) is the case, v_x must have at least one other edge apart from the one which we've just added, and because v_x is not part of the original path, it must be an edge not part of the original path.

Now, we can prove the lemma. Let's add new and new vertices to the original Euler path, such that the new edges are not part of the original path. Because of the previous arguments, we can do this until we reach v_1. This will not be an infinite process, because the graph is finite, so there must be a finite $v'_1, \dots v'_n$ Euler path back to v_1 from v_n. Let's define an Euler loop as a modified Euler path where $v_1 = v_n$.

Next, let's create an Euler cycle. We do that by creating an arbitrary Euler loop, and if any edge is not part of this loop, we enlarge this loop so that it contains any new edge. First step is to create an initial Euler loop. This is easy, because any edge by itself is an Euler path, and so it has an inverse, and together they form an Euler loop. Next, let's have any edge (v_b, v_c) which is not part of this loop. Because the graph is connected, there is a path between v_b and any vertex of the loop v_a. We can make this path an Euler path by cutting parts of the path until it's an Euler path. (If an edge is in the path multiple times, we shorten the path so that we do not use the path segment which is between the two mentions.) Also, we incorporate the edge (v_b, v_c) into this path: if it's not there, then we extend it so that the path ends at v_c. Next, we cut this $v_b/v_c \to v_a$ Euler path until it reaches any vertex in the loop instead of going until v_a. Let's say, that this vertex where this path reaches the loop is v'_a. This means, that we can create an Euler path which starts at v'_a, goes around with the original loop until it reaches v'_a, and then goes to v_b/v_c. Because of the lemma, we can add an inverse of this, extending this into an Euler loop which contains the original loop, and the new edge. We can do this iteratively until all edges are inside the loop, and this will be the Euler cycle.

Proving that if an Euler cycle exist, that each vertex has even edges is obvious. If it wasn't the case, there was one vertex where the path ended, which is impossible because it is a loop.

Exercise 3.21 If L_1 is reducible to L_2, there is a Turing machine operating polynomial in time which will map $x \mapsto R(x)$, and $x \in L_1$ if and only if $R(x) \in L_2$. If L_2

is reducible to L_3, there is an other Turing machine operating polynomial in time which maps $R(x)$ to $Q(R(x))$ if and only if $R(x) \in L_3$. Then, there exists a Turing machine polynomial in time such that $x \in L_1$ if and only if $Q(R(x)) \in L_3$.

Exercise 3.22 We can use the transitivity property. Because any L is complete, every L'' can be reduced to L. And also, L is reducible to L', which means, that any L'' can be reduced to L', hence, L' is complete.

Exercise 3.23 We have to create a machine M which has inputs (φ, x), which evaluates in polynomial time the formula φ to $x = (x_1, \dots x_n)$ to prove, that `SAT` is **NP**. We can for example create the M machine so that it first substitutes the values of x into the formula φ, and then evaluates it. The substitution process is linear, $O(n)$. In the worst case scenario, each evaluation substitutes the evaluation of one connective, and then shortens the substituted formula. This one evaluation process is $O(n)$ in time. And in the worst case scanario, the number of connectives is proportional to the length of the formula, so the whole evaluation is $O(n^2)$, which is polynomial.

Next, we have to show, that `CSAT` is reducible to `SAT`. If it is, then `SAT` is **NP** complete. It is not a very explicit proof, because it depends on how we represent the circuit of `CSAT`, and the book does not talk about it. But essentially, each input wire is mapped to a variable x_i in the formula, and each outgoing wire is represented by the conjunction/disjunction/negation of the input wire of a gate, which is also a formula (either one or two x_i, or a formula containing multiple x_i. We can construct this complex formula by expanding it when we are reading the gates of the `CSAT`. This expansion process will also take $O(n^2)$ in time, because if let's say the input of `CSAT` take up linear space in the number of gates, each new expansion of the formula (by adding a new gate) takes $O(n)$ in time.

Exercise 3.24 [6]

(1) Let's rephrase the exercise, because if it's true, that there is a directed edge (α, β) if and only if $(\neg\alpha \vee \beta)$ or $(\beta \vee \neg\alpha)$ is present in φ (as a formula, not as a truth statement, as the exercise says), then there cannot be any directed edge to or from $\neg x$, hence, there would be no graphs which has path to or from $\neg x$, and any formula is satisfiable (according to the exercise), which is not true.

Let's say, that there there are only these three rules to create edges.

(i) If $(\alpha \vee \beta)$ is present, then we add the directed edges $(\neg\alpha, \beta)$ and $(\neg\beta, \alpha)$,

(ii) if $(\neg\alpha \vee \beta)$ or $(\beta \vee \neg\alpha)$, add the directed edges (α, β) and $(\neg\beta, \neg\alpha)$,

(iii) if $(\neg\alpha \vee \neg\beta)$, add the directed edges $(\alpha, \neg\beta)$ and $(\beta, \neg\alpha)$.

We know, that each of these clauses has to be true, because they are connected with AND. Let's analyze one of the clauses. If $(\alpha \vee \beta)$, then it's also true, that if α is not true, then β must be true. Also, if β is not true, then α must be true. We can write these as $\neg\alpha \to \beta$ and $\neg\beta \to \alpha$. The same is true for the

other two rules mentioned. Also, the converse is true. We can summarize this as

$$(\alpha \vee \beta) \Leftrightarrow (\neg\alpha \to \beta) \text{ and } (\neg\beta \to \alpha),$$
$$(\neg\alpha \vee \beta) \Leftrightarrow (\alpha \to \beta) \text{ and } (\neg\beta \to \neg\alpha), \tag{215}$$
$$(\neg\alpha \vee \neg\beta) \Leftrightarrow (\alpha \to \neg\beta) \text{ and } (\beta \to \neg\alpha). \tag{216}$$

That means, that when a directed path x to $\neg x$ is found, then the statement $x \to \neg x$ can be derived from the original φ formula. Similarly to $\neg x \to x$. If both of the paths are found, then $x \leftrightarrow \neg x$, which is impossible, because x can only be true or false, and not both. The formula can never be satisfied.

Let's prove the converse: if φ is not satisfiable, then there is a path which goes from x to $\neg x$ and a path going from $\neg x$ to x. We only have to prove, that the formula is satisfiable if there is not path from x to $\neg x$ while it's also true that there is a path from $\neg x$ to x for every x. Let's find the values of x_i. First, let's set any x to true, if there is any path from $\neg x$ to x, and set x to false if there is a path from x to $\neg x$. If there is any path from these mentioned $x/\neg x$ to any x_i, then we set x_i to true, and if there is a path to any $\neg x_i$, we set x_i to false. If there is any other x_j which has not been set, we set it to any value, and follow all of it's path (or $\neg x_j$'s paths), and set each other x_k appropriately. We do this process iteratively. We constructed an x which satisfies the formula, so we just proved, the converse.

(2) We can use the same method as in **Exercise 3.19**, with the small modification that we take into account the direction of edges.

(3) The algorithm first translates the formula to a graph, which is linear in time in the number of clauses, because each clause is translated to two directed edges. Next, for each x it is checked whether there is a path from x to $\neg x$, and from $\neg x$ to x. This is done in $O(n^2)$. The output is true if there are no paths $x \to \neg x$ and $\neg x \to x$ for every x, otherwise false.

3.2.4 A plethora of complexity classes

Exercise 3.25 The hint answers the question. If the machine is PSPACE, then it uses $p(n)$ space, where $p(n)$ is some polynomial. The machine can be in any of the l internal states, and for any of the internal states the first square of the tape can be in m states, the second too, and so until, until the $p(n)$ square. The internal state and all of the usable squares can be in $lm^{p(n)}$ different number of states. Also, the tape head can be anywhere from 1 to $p(n)$. All in all, the whole machine can be in $lp(n)m^p(n)$ states. If from any of the state S it goes back to state S, then the machine never halts, it's going to be an infinite loop, so it cannot go back to any of the states it's already visited. In the worst case it's going through $lp(n)m^p(n)$ states, which is $O(2^{n^q})$, if $q > k$.

Exercise 3.26 We can use the same method as the previous exercise. The internal state can be in any of the l states, the first tape head can be in any of the n squares,

the second tape head can be in any of the $\log(n)$ squares, and the second tape can be in any of the $m^{\log(n)}$ states. If the machine does not halt, the worst case scenario is that it goes through all the $ln\log(n)m^{\log(n)}$ states, which is $O(n^q)$, if $q > log(m)+1$.

Exercise 3.27 [7] The minimal vertex cover contains only one or two of the vertices of each edge. The important thing is that it contains at least one of the vertices. The algorithm clearly produces a cover, because for each edge it removes, it adds vertices which are connected to these edges, and every edge is removed. Next, we can prove, that the number of vertices it produces is at most 2 times the possible minimal amount of vertices. Let's say, that algorithm picks n number of edges. In each cycle it removes a set of edges too, but the minimal cover contains at least one edge for all of those edges removed in the cycle, which is n. Each cycle the algorithm adds 2 vertices to the cover, it returns with $2n$ vertices, and the minimal cover contains at least n vertices.

Exercise 3.28 There is an algorithm, which runs M multiple times (n times), and calculates whether the majority of the results were 0 or 1, and returns 0 if the majority was 0, and 1 otherwise. If $n \to \infty$, the probability can be arbitrarily close to 1, that this new machine returns 1 if $x \in L$, and returns 0 if $x \notin L$. See Box 3.4.

3.2.5 Energy and computation

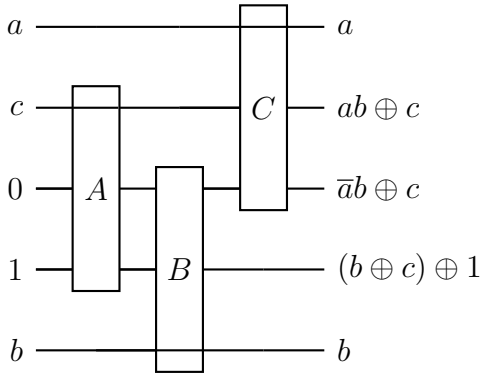
Exercise 3.29 There are two possibilities: $c = 0$ and $c = 1$. Note, that $c = c' = c''$. When $c = c' = 0$, the two gates will not change the input at all. When $c = c' = 1$, the first gate will flip a and b: $a' = b$ and $b' = a$. The second gate will flip these again: $a'' = b' = a$ and $b'' = a' = b$. Both cases will result in $a'' = a$ and $b'' = b$.

Exercise 3.30 It can be verified on the Figure 3.14.

Exercise 3.31 There are 4 wires, which are (w_1, w_2, w_3, w_4). Initially the wires are $(x, y, 0, 0)$. Next, apply a Fredkin gate to the wires (w_1, w_2, w_4) as (c, b, a). The state of the 4 wires will be $(x, \overline{x}y, 0, xy)$ (the explanation is on Figure 3.16 in the book). Next, we apply a `CNOT` gate to (w_3, w_4) is $(w_3 \oplus w_4, w_4)$. The result on the 4 wires is $(x, \overline{x}y, xy, xy)$. Next, we apply a Fredkin gate on the wires (w_1, w_2, w_4) exactly as before, so that it reverses those wires, so that the wires are now in the state $(x, y, xy, 0)$. Now, apply a `CNOT` to (w_2, w_4) so that it's $(w_2, w_2 \oplus w_4)$, and then an other `CNOT` to (w_1, w_4) so that it's $(w_1, w_1 \oplus w_4)$. The result on the 4 wires is $(x, y, xy, x \oplus y)$.

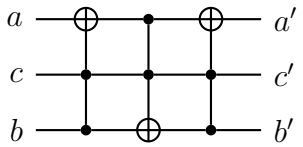
Exercise 3.32 I created a small python script which checks the effect of all the possible gates and different input ancilla bits on inputs a, b, c. When it found outputs which match the expected, it outputted the gate configuration together with the ancilla bits, and which output corresponds to the expected.

To create a Toffoli gate from Fredkin gates, the least amount of gates needed is 3, and the least amount of ancilla bits needed is 2.



As we can see, the A, B and C Fredkin gates all in all create a Toffoli gate on the first wire as a, last wire as b and second wire as $ab \oplus c$.

To create a Fredkin gate from Toffoli gates, the least amount of gates needed is 3, and the least amount of ancilla bits needed is 0.



We can check, that this really is a Fredkin gate by calculating each case.

References

[1] Karl R. Abrahamson. CSCI 6420 Lecture Notes East Carolina University. www.cs.ecu.edu/karl/6420/spr16/Notes/CoNP/relationsships.html.

[2] Goropikari. Solutions by Goropikari. https://github.com/goropikari/SolutionQCQINielsenChuang.

[3] ASKASK (https://math.stackexchange.com/users/136368/askask). Does the logarithm function grow slower than any polynomial? https://math.stackexchange.com/q/1663823.

[4] Alastair Kay. Tutorial on the quantikz package. *arXiv preprint arXiv:1809.03842*, 2018.

[5] D.E. Knuth, Addison-Wesley, and Pearson Education. *The Art of Computer Programming*. Addison-Wesley series in computer science and information processing. Addison-Wesley, 1997. Section 5.3.1: Minimum-Comparison Sorting.

[6] Ivanov Maksim. E-Maxx Algorithms in English. https://cp-algorithms.com/graph/2SAT.html.

[7] Tandy Warnow. CS 105: Algorithms Lecture Notes University of Illinois. http://tandy.cs.illinois.edu/dartmouth-cs-approx.pdf.