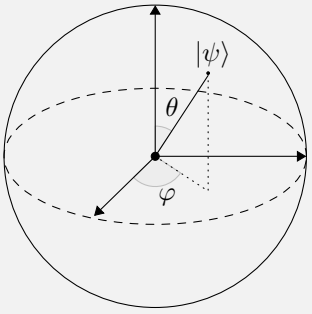
Solutions to Quantum Computation and Quantum Information (Vol. 2)



Arnab Adhikary & Rio Weil

This document was typeset on October 13, 2025

Introduction:

This document is the second (highly incomplete) volume to a collection of comprehensive solutions to the exercises and problems in Nielsen and Chuang's "Quantum Computation and Quantum Information". Each solution has the involved concepts (and hence rough pre-requisite knowledge) necessary for the problem in addition to the solution. Some exercises may contain additional remarks about implications. Starred exercises were considered to be more difficult (difficulty is assumed for the problems at the end of the chapter). When it is completed in the future, the second volume will contain solutions to all exercises/problems in Chapters 1 (Introduction and overview), 3 (Introduction to computer science), 5 (The quantum Fourier transform and its applications), 6 (Quantum search algorithms), 11 (Entropy and information), and 12 (Quantum information theory). It will also contain solutions to all exercies/problems in Appendices 1-6 (Notes on basic probability theory, Group theory, The Solovay-Kitaev theorem, Number Theory, Public key cryptography, and Proof of Lieb's theorem).

The most up-to-date version of this document, as well as a feedback form can be found at `https://nielsenandchuangsolutions.github.io`.

Contents

1 Introduction and overview

Exercise 1.1: Probabilistic Classical Algorithm

Suppose that the problem is not to distinguish between the constant and balanced functions with certainty, but rather, with some probability of error $\epsilon < 1/2$. What is the performance of the best classical algorithm for this problem?

Solution

Concepts Involved: Deutsch's Problem, Probability.
Recall that a Boolean function $f : \{0,1\}^n \to \{0,1\}$ is said to be *balanced* if $f(x) = 1$ for exactly half of all possible 2^n values of x.

A single evaluation tells us no information about whether f is constant or balanced, so our success rate/error rate after a single evaluation is $\epsilon = \frac{1}{2}$ (random guessing!). Therefore, consider the case where we do two evaluations. If we obtain two different results, then we immediately conclude that f is balanced. Suppose instead that we obtain two results that are the same. If f is balanced, then the probably that the first evaluation returns the given result is $\frac{1}{2}$, and the probability that the second evaluation returns the same result is $\frac{2^n/2-1}{2^n-1}$ (as there are $2^n/2$ of each result of 0 and 1, 2^n total results, $2^n/2 - 1$ of the given result left after the first evaluation, and $2^n - 1$ total uninvestigated cases after the first evaluation). Therefore, if f is balanced, this occurs with probability $\frac{1}{2} \cdot \frac{2^n/2-1}{2^n-1}$, which we can see is less than $\frac{1}{2}$ as:

$$2^n < 2^{n+1} \implies 2^n - 2 < 2^{n+1} - 2 \implies \frac{2^n/2 - 1}{2^n - 1} < 1 \implies \frac{1}{2}\frac{2^n/2 - 1}{2^n - 1} < \frac{1}{2}$$

Hence, if we get the same result in two evaluations, we can conclude that f is constant with error $\epsilon < \frac{1}{2}$. We conclude that only 2 evaluations are required for this algorithm. □

Exercise 1.2

Explain how a device which, upon input of one of two non-orthogonal quantum states $|\psi\rangle$ or $|\varphi\rangle$ correctly identified the state, could be used to build a device which cloned the states $|\psi\rangle$ and $|\varphi\rangle$, in violation of the no-cloning theorem. Conversely, explain how a device for cloning could be used to distinguish non-orthogonal quantum states.

Solution

Concepts Involved: Quantum Distinguishability, Quantum Measurement.

Given access to a device which can distinguish non-orthogonal quantum states $|\psi\rangle, |\varphi\rangle$ (without measurement), we can then design a quantum circuit that would map $|\psi\rangle \mapsto |\varphi\rangle$ (or vise versa), allowing us to clone the states as we like.
Conversely, given a cloning device, we could clone $|\psi\rangle$ and $|\varphi\rangle$ an arbitrary number of times. Then, performing repeated measurements of the two states in different measurement bases, we would (given enough measurements) be able to distinguish the two states based on the measurement statistics (there will of course be some error ϵ based on probabilistic considerations, but given that we have access to as many measurements of the states as we like, we are able to make this error arbitrarily low). □

Problem 1.1: (Feynman-Gates conversation)

Construct a friendly imaginary discussion of about 2000 words between Bill Gates and Richard Feynman, set in the present, on the future of computation (Comment: You might like to try waiting until you've heard the rest of the book before attempting this question. See 'History and further reading' below for pointers to one possible answer for this question).

Problem 1.2

What is the most significant discovery yet made in quantum computation and quantum information? Write an essay of about 2000 words to an educated lay audience about the discovery (Comment: As for the previous problem, you might like to try waiting until you've read the rest of the book before attempting this question.)

3 Introduction to computer science

Exercise 3.1: Non-computable processes in Nature

How might we recognize that a process in Nature computes a function not computable by a Turing machine?

Solution

Concepts Involved: Turing Machines.

One criteria is natural phenomena that appear to be truly random; Turing machines as defined in the text are deterministic (though there are probabilistic variations that would solve this issue) and hence would not be able to compute a random function. From a more direct point, if a process in Nature was to be found to compute a known non-computable problem (e.g. solve the Halting problem or the Tiling problem) then we may conclude (trivially) that the process would not be computable. However since the domain of inputs that we could provide top such a natural process would have to be finite, there would be no concrete method in which one could actually test if such a process was truly computing a non-Turing computable function (as a Turing machine that works on a finite subset of inputs for an uncomputable problem could be devised). □

Exercise 3.2: Turing numbers

$(*)$ Show that single-tape Turing machines can each be given a number from the list $1, 2, 3, \ldots$ in such a way that the number uniquely specifies the corresponding machine. We call this number the *Turing number* of the corresponding Turing machine. (Hint: Every positive integer has a unique prime factorization $p_1^{a_1} p_2^{a_2} \ldots p_k^{a_k}$, where p_i are distinct prime numbers, and $a_1, \ldots a_k$ are non-negative integers.)

Solution

Concepts Involved: Turing Machines, Cardinality.

Lemma 1. If $\{A_n\}_{n=1}^{\infty}$ is a sequence of sets that are countably infinite (that is, they can be put in bijection with the natural numbers $\mathbb{N}$) then their union $A = \bigcup_{n=1}^{\infty} A_n$ is also countably infinite.

Proof. Write $A_n = \{x_{n1}, x_{n2}, x_{n3}, \ldots\}$ (which we can do as each of the A_ns are countably infinite). Then, we form an array:

$$\begin{aligned} A_1 &= x_{11} \quad x_{12} \quad x_{13} \quad \cdots \\ A_2 &= x_{21} \quad x_{22} \quad x_{23} \quad \cdots \\ A_3 &= x_{31} \quad x_{32} \quad x_{33} \quad \cdots \\ &\cdots \end{aligned}$$

Then, we can re-number the elements along the diagonal lines (i.e. $x_{11}, x_{21}, x_{12}, x_{31}, x_{22}, x_{13}, \ldots$). This new enumeration corresponds to a countably infinite set. From there, we let $T \subset \mathbb{N}$ be the remaining labels in the enumeration after removing the repeated elements from the sequence. Then, $|T| = |A|$, and hence A is at most countably infinite. A cannot be finite as $A_1 \subset A$ and A_1 is not finite. Hence A is countably infinite. □

Lemma 2. The set of all finite sequences of elements from a countably infinite set B is also countably infinite.

Proof. Denote B_n the set of length n sequences consisting of elements from B. We show that B_n is countably infinite by induction.

Base Case: $|B| = |B_1|$, so B_1 is countably infinite.

Inductive Step: Suppose that B_k is countably infinite for $k \geq 1$. Then, define $B_{k,i}$ which is the set of $k+1$ length sequences, consisting of all sequences in B_k and then terminating with the ith element of B. $\left|B_{k,i}\right| = |B_k|$ so $B_{k,i}$ is countably infinite, and then $B_{k+1} = \bigcup_{i=1}^{\infty} B_{k,i}$ is countably infinite by Lemma 1. This concludes the argument by induction.

Finally, the set of all finite sequences of elements from B is simply $\bigcup_{k=1}^{\infty} B_k$, which is a union of countably infinite sets and therefore countably infinite again by Lemma '. □

Solution. A single-tape Turing machine can be uniquely specified by a finite integer sequence as follows. First, we state the number of states $|Q| = N$, then enumerate the states $q_s = -1, q_h = 0, q_1 = 1, \ldots, q_{N-2} = N-2$. We then state the size of the alphabet $|\Gamma| = L$ and enumerate the alphabet $\triangleright = -2, b = -1, 0 = 0, 1 = 1, \ldots, L-3 = L-3$. We then write the sequence of symbols appearing on the initial tape, starting from square 0, stopping when we encounter the last non-blank symbol. Finally, we list the program lines, each of which is a five-tuple of integers (as each of the states and symbols appearing in the program line we have associated to an integer, and the action s is one of $-1, 0, 1$). Between each part of this sequence we put a "space integer" -3.

As an example, the Turing machine in the text which computes the constant function (and let us say for the sake of example the tape starts with the number 7 on it, so 110 in binary followed by infinite blanks) has the associated finite integer sequence:

$$\underbrace{5}_{|Q|}, \underbrace{-1}_{q_s}, \underbrace{0}_{q_h}, \underbrace{1}_{q_1}, \underbrace{2}_{q_2}, \underbrace{3}_{q_3}, -3, \underbrace{4}_{|\Sigma|}, \underbrace{-2}_{\triangleright}, \underbrace{-1}_{b}, 0, 1, -3, \underbrace{1,1,0}_{7}, -3, \underbrace{-1,-2,1,-2,1}_{\langle q_s,\triangleright,q_1,\triangleright,+1\rangle},$$

$$\underbrace{1,0,1,-1,1}_{\langle q_1,0,q_1,b,+1\rangle}, \underbrace{1,1,1,-1,1}_{\langle q_1,1,q_1,b,+1\rangle}, \underbrace{1,-1,2,-1,-1}_{\langle q_1,b,q_2,b,-1\rangle}, \underbrace{2,-1,2,-1,-1}_{\langle q_2,b,q_2,b,-1\rangle}, \underbrace{2,-2,3,-2,1}_{\langle q_2,\triangleright,q_3,\triangleright,+1\rangle}, \underbrace{3,-1,0,1,0}_{\langle q_3,b,q_h,1,0\rangle}$$

Because the number of states is finite, the size of the alphabet is finite, the number of non-blank squares on the initial tape is finite, and there is a finite number of lines in the program, the above algorithm produces a finite sequence of integers for a given Turing machine.

Now, observe that the integers are countably infinite, as:

$$\begin{array}{rcccl} f & : & \mathbb{N} & \longrightarrow & \mathbb{Z} \\ & & n & \longmapsto & \begin{cases} \frac{n}{2} & n \text{ even} \\ -\frac{n-1}{2} & n \text{ odd} \end{cases} \end{array} \tag{1}$$

is a bijection. So too then is the set of all finite integer sequences by the Lemma 2. Since the finite integer sequences describing Turing machines as produced by the above prescription are a subset of all finite integer sequences, we must therefore be able to associate any single-tape Turing machine to a unique natural number, as claimed. □

Exercise 3.3: Turing machine to reverse a bit string

Describe a Turing machine which takes a binary number x as input, and outputs the bits of x in reverse order. (*Hint:* In this exercise and the next it may help to use a multi-tape Turing machine and/or symbols other than $\triangleright, 0, 1$ and the blank.)

Exercise 3.4: Turing machine to add modulo 2

Describe a Turing machine to add two binary numbers x and y modulo 2. The numbers are input on the Turing machine tape in binary, in the form x, followed by a single blank, followed by a y. If one number is not as long as the other then you may assume that it has been padded with leading 0s to make the two numbers the same length.

Exercise 3.5: Halting problem with no inputs

Show that given a Turing machine M there is no algorithm to determine whether M halts when the input to the machine is a blank tape.

Exercise 3.6: Probabilistic halting problem

Suppose we number the probabilistic Turing machines using a scheme similar to that found in Exercise 3.2 and define the probabilistic halting function $h_p(x)$ to be 1 if machine x halts on input of x with probability at least $1/2$ and 0 if machine x halts on input of x with probability less than $1/2$. Show that there is no probabilistic Turing machine which can output $h_p(x)$ with probability of correctness strictly greater than $1/2$ for all x.

Exercise 3.7: Halting oracle

Suppose a *black box* is made available to us which takes a non-negative integer x as input, and then outputs the value of $h(x)$, where $h(\cdot)$ is the halting function defined in Box 3.2 on page 130. This type of black box is sometimes known as an *oracle* for the halting problem. Suppose we have a regular Turing machine which is augmented by the power to call the oracle. One way of accomplishing this is to use a two-tape Turing machine, and add an extra program instruction to the Turing machine which results in the oracle being called, and the value of $h(x)$ being printed on the second tape, where x is the current contents of the second tape. It is clear that this model for computation is more powerful than the conventional Turing machine model, since it can be used to compute the halting function. Is the halting problem for this model of computation undecidable? That is, can a Turing machine aided by an oracle for the halting problem decide whether a program for the Turing machine with oracle will halt on a particular input?

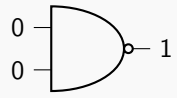
Exercise 3.8: Universality of `NAND`

Show that the `NAND` gate can be used to simulate the `AND`, `XOR`, and `NOT` gates, provides wires, ancilla bits and `FANOUT` are available.

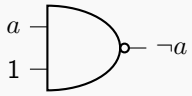
Solution

Concepts Involved: Logic Gates.

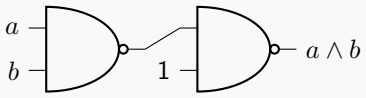
We start by showing how we can get a 1 qubit using two 0 ancilla bits and a `NAND` gate.



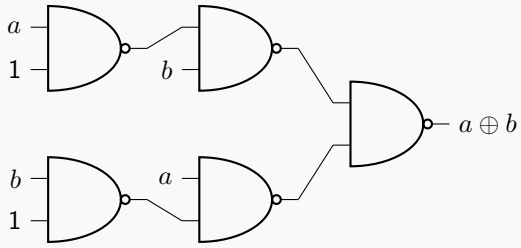
We will now show how to simulate the `NOT`, `AND`, and `XOR` gates. We note that we will use "1" to denote as shorthand a 1 bit constructed using two ancilla bits (as above). a/b represent the input bits. We start with the `NOT` gate.



Next, we simulate the `AND` gate.



For the `XOR` simulated gate, we note that we first use `FANOUT` twice to copy both input bits.



Having simulated the three gates using the `NAND` gate only, we conclude that the `NAND` is universal. □

Exercise 3.9

Prove that $f(n)$ is $O(g(n))$ if and only if $g(n)$ is $\Omega(f(n))$. Deduce that $f(n)$ is $\Theta(g(n))$ if and only if $g(n)$ is $\Theta(f(n))$.

Solution

Concepts Involved: Asymptotic Notation.

Suppose $f(n)$ is $O(g(n))$. Then, there exists $c > 0$ such that for all $n > n_0$, $f(n) \leq cg(n)$. Therefore, we have $\frac{1}{c} > 0$ such that for all $n > n_0$, $\frac{1}{c}f(n) \leq g(n)$. Hence, $g(n)$ is $\Omega(f(n))$. Conversely, if $g(n)$ is $\Omega(f(n))$, there exists $c > 0$ such that for all $n > n_0$, $cf(n) \leq g(n)$. Hence, we have $\frac{1}{c} > 0$ such that for all $n > n_0$, $f(n) \leq \frac{1}{c}g(n)$ and hence $f(n)$ is $O(g(n))$.
Therefore, if $f(n)$ is $\Theta(g(n))$ then $f(n)$ is $O(g(n))$ and $\Omega(g(n))$, and by the above argument, $g(n)$ is $O(f(n))$ and $\Omega(f(n))$ and hence $g(n)$ is $\Theta(f(n))$. The converse holds in the same way. □

Exercise 3.10

Suppose $g(n)$ is a polynomial of degree k. Show that $g(n)$ is $O(n^l)$ for any $l \geq k$.

Solution

Concepts Involved: Asymptotic Notation.

By assumption, $g(n) = a_0 + a_1 n^1 + a_2 n^2 + \ldots + a_k n^k$ with $a_k \neq 0$. For $n \geq 1$ we have that $n^l \geq n^k$ if $l \geq k$, and hence if $l \geq k$ we have that $a_i n^l \geq a_i n^i$ for all $i \in \{0, \ldots, k\}$. Therefore, we have that:

$$(a_0 + a_1 + \ldots + a_k) n^l \geq a_0 + a_1 n^1 + \ldots + a_k n^k = g(n)$$

for $n \geq 1$ and hence $g(n)$ is $O(n^l)$. □

Exercise 3.11

Show that $\log n$ is $O(n^k)$ for any $k > 0$.

Solution

Concepts Involved: Asymptotic Notation.

Let $k > 0$ and $c > 0$. By the definition of the exponential we have that:

$$\exp\left(cn^k\right) = \sum_{j=0}^{\infty} \frac{(an^k)^j}{j!} = \sum_{j=0}^{\infty} \frac{a^{kj} n^{kj}}{j!}$$

now, there exists some $j_0 \in \mathbb{Z}$ for which $kj_0 > 1$. Since for $n \geq 0$ the terms in the above sum are non-negative, we find:

$$\exp\left(cn^k\right) \geq \frac{c^{kj_0} n^{kj_0}}{j_0!}$$

Now, choose c sufficiently large such that $c^{kj_0} \geq j_0!$. We then find that:

$$\exp\left(cn^k\right) \geq \frac{c^{kj_0} n^{kj_0}}{j_0!} \geq n^{kj_0}$$

Then for $n \geq 1$ it follows that $n^{kj_0} \geq n$ as $kj_0 \geq 1$ and so:

$$\exp\left(cn^k\right) \geq n$$

Since the logarithm is monotonic, we may take the log of both sides and preserve the inequality:

$$cn^k \geq \log n$$

So we have shown that for any $k > 0$, there exists $c > 0$ such that for all $n > 1$, $cn^k \geq \log n$. Hence, $\log n$ is $O(n^k)$ for any $k > 0$. □

Exercise 3.12: $n^{\log n}$ is super-polynomial

Show that n^k is $O(n^{logn})$ for any k, but that $n^{\log n}$ is never $O(n^k)$.

Solution

Concepts Involved: Asymptotic Notation.

First, note that for any k, $e^k \leq n$ for sufficiently large $n > n_0$ and so $k \leq \log n$ by monotonicity of the logarithm. Therefore, for $n > n_0$ it follows by monotonicity (of exponentiation) that $n^k \leq n^{\log n}$ and so n^k is $O(n^{\log n})$.
Now, consider an arbitrary $a > 0$. It still follows for sufficiently large $n > n_0$ that $e^{ak} \leq n$ and so $ak \leq \log n$ and $n^a n^k \leq n^{\log n}$. But for any $c > 0$ $n^a > c$ for sufficiently large n and so:

$$cn^k \leq n^{\log n}$$

So since for any $c > 0$ there exists some n_0' for which $n > n_0'$ implies $cn^k \leq n^{\log n}$, it follows that $n^{\log n}$ is never $O(n^k)$. □

Exercise 3.13: $n^{\log n}$ is sub-exponential

Show that c^n is $\Omega(n^{\log n})$ for any $c > 1$, but that $n^{\log n}$ is never $\Omega(c^n)$.

Solution

Concepts Involved: Asymptotic Notation.

First note from Exercise 1.11 that $\log n$ is $O(n^k)$ for any $k > 0$. Specifically, take $k = 1/2$; then there exists $a > 0$ such that for $n > n_0$:

$$an^{1/2} \geq \log n$$

and therefore squaring both sides:

$$a^2 n \geq \log n \log n = \log n^{\log n}$$

Now for any $c > 1$, we can define $a' = \frac{a^2}{\log c} > 0$ and write:

$$na' \log c = \log c^{na'} \geq \log n^{\log n}$$

Exponentiating both sides preserves the inequality, and so:

$$c^{na'} = c^{a'} c^n \geq n^{\log n}$$

and so there exists a constant $\frac{1}{c^{a'}} > 0$ such that for $n > n_0$, $c^n \geq \frac{1}{c^{a'}} n^{\log n}$ and therefore c^n is $\Omega(n^{\log n})$.
Now, let $b > 0$ be some arbitrarily small constant. For sufficiently large n, we have that $na'logc + \log b > 0$

and so for sufficiently large n it further follows that:

$$\log b + na' \log c = \log\left(bc^{na'}\right) \geq \log n^{\log n}$$

where a' is defined as it was previously. Therefore exponentiating both sides:

$$bc^{na'} = bc^{a'}c^n \geq n^{\log n}$$

so for sufficiently large n, for any arbitrarily small constant b' it follows that $b'c^n \geq n^{\log n}$ and so $n^{\log n}$ is never $\Omega(c^n)$. □

Exercise 3.14

Suppose $e(n)$ is $O(f(n))$ and $g(n)$ is $O(h(n))$. Show that $e(n)g(n)$ is $O(f(n)h(n))$.

Solution

Concepts Involved: Asymptotic Notation.

By assumption, we have that $e(n) \leq c_1 f(n)$ for some $c_1 > 0$ and for all $n > n_1$ and that $g(n) \leq c_2 h(n)$ for some $c_2 > 0$ and for all $n > n_2$. Let $n_0 = \max n_1, n_2$. We then have that for $n > n_0$ that:

$$e(n)g(n) \leq c_1 f(n) c_2 h(n) = (c_1 c_2)(f(n)h(n))$$

so therefore $e(n)g(n)$ is $O(f(n)h(n))$. □

Exercise 3.15: Lower bound for compare-and-swap based sorts

Suppose an n element list is sorted by applying some sequence of compare-and-swap operations to the list. There are $n!$ possible initial orderings of the list. Show that after k of the compare-and-swap operations have been applied, at most 2^k of the possible initial orderings will have been sorted into the correct order. Conclude that $\Omega(n \log n)$ compare and swap operations are required to sort all possible initial orderings into the correct order.

Solution

Concepts Involved: Asymptotic Notation, Compare-and-Swap.

We prove the first statement by induction. After 0 steps, we have that $1 = 2^0$ out of the $n!$ possible orderings are already sorted. Let $k \in \mathbb{N}, k \geq 0$ and suppose that after k swaps, at most 2^k of the initial orderings have been sorted into the correct order. We now consider the state of the list after the $k+1$th swap. Each of the 2^k initial orderings from the previous step are correctly sorted already (so the swap does nothing), and there are a further 2^k initial orderings that are one swap away from the 2^k from the previous step, and hence the $k+1$th swap will put 2^k more initial orderings into the correct order. Therefore, after 2^{k+1} compare and swaps, there are at most $2^k + 2^k = 2^{k+1}$ possible initial orderings that are sorted into the correct order. This proves the claim.
Using the above fact, we have that in order to have all $n!$ possible initial orderings correct after k

steps that $2^k \geq n!$. Taking logarithms on both sides, we have that $\log(2^k) \geq \log(n!)$ and hence $k \geq \log(n!)$. Using Stirling's approximation for factorials (`https://en.wikipedia.org/wiki/Stirling%27s_approximation`), we have that:

$$k \geq n \log n - n \log e + O(\log n)$$

from which we conclude that k is $\Omega(n \log n)$ and hence $\Omega(n \log n)$ compare and swap operations are required to sort all possible initial orderings into the correct order. □

Exercise 3.16: Hard-to-compute functions exist

Show there exist Boolean functions on n inputs which require at least $2^n / \log n$ logic gates to compute.

Exercise 3.17

Prove that a polynomial-time algorithm for finding the factors of a number m exists if and ony if the factoring decision problem is in $\mathbf{P}$.

Exercise 3.18

Prove that if $\mathbf{coNP} \neq \mathbf{NP}$ then $\mathbf{P} \neq \mathbf{NP}$.

Exercise 3.19

The `Reachability` problem is to determine whether there is a path between two specified vertices in a graph. Show that `Reachability` can be solved using $O(n)$ operations if the graph has n vertices. Use the solution to `Reachability` to show that it is possible to decide whether a graph is connected in $O(n^2)$ operations.

Exercise 3.20: Euler's theorem

Prove Euler's theorem. In particular, if each vertex has an even number of incident edges, give a constructive procedure for finding an Euler cycle.

Exercise 3.21: Transitive property of reduction

Show that if a language L_1 is reducible to the language L_2 and the language L_2 is reducible to L_3 then the language L_1 is reducible to the language L_3.

Exercise 3.22

Suppose L is complete for a complexity class, and L' is another language in the complexity class such that L reduces to L'. Show that L' is complete for the complexity class.

Exercise 3.23

Show that `SAT` is $\mathbf{NP}$-complete by first showing that the `SAT` is in $\mathbf{NP}$, and then showing that `CSAT` reduces to `SAT`.

Exercise 3.24: `2SAT` has an efficient solution

Suppose φ is a Boolean formula in conjunctive normal form, in which each clause contains only two literals.

(1) Construct a (directed) graph $G(\varphi)$ with directed edges in the following way: the vertices of G correspond to variables x_k and their negations $\neg x_j$ in φ. There is a (directed) edge (α, β) in G if and only if the clause $(\neg\alpha \vee \beta)$ or the clause $(\beta \wedge \neg\alpha)$ is present in φ. Show that φ is not satisfiable if and only if there exists a variable x such that there are paths from x and $\neg x$ and from $\neg x$ to x in $G(\varphi)$.

(2) Show that given a directed graph G containing n vertices it is possible to determine whether two vertices v_1 and v_2 are connected in polynomial time.

(3) Find an efficient algorithm to solve `2SAT`.

Exercise 3.25: $\mathbf{PSPACE} \subseteq \mathbf{EXP}$

The complexity class $\mathbf{EXP}$ (for *exponential time*) contains all decision problems which may be decided by a Turing machine running in exponential time, that is time $O(2^{n^k})$, where k is any constant. Prove that $\mathbf{PSPACE} \subseteq \mathbf{EXP}$. (*Hint:* If a Turing machine has l internal states, an m letter alphabet, and uses space $p(n)$, argue that the machine can exist in one of at most $lm^{p(n)}$ different states, and that if the Turing machine is to avoid infinite loops then it must halt before revisiting a state.)

Exercise 3.26: $\mathbf{L} \subseteq \mathbf{P}$

The complexity class $\mathbf{L}$ (for *logarithmic space*) contains all decision problems which may be decided by a Turing machine running in logarithmic space, that is, in space $O(\log(n))$. More precisely, the class $\mathbf{L}$ is defined using a two-tape Turing machine. The first tape contains the problem instance, of size n, and is a read-only tape, in the sense that only program lines which don't change the contents of the first tape are allowed. The second tape is a working tape which initially contains only blanks. The logarithmic space requirement is imposed on the second, working tape only. Show that $\mathbf{L} \subseteq \mathbf{P}$.

Exercise 3.27: Approximation algorithm for `VERTEX COVER`

Let $G = (V, E)$ be an undirected graph. Prove that the following algorithm finds a vertex cover for G that is within a factor of two of being a minimial vertex cover.

$VC = \emptyset$
$E' = E$
while $E' = \emptyset$ **do**
  let (α, β) be any edge of E'
  $VC = VC \cup \{\alpha, \beta\}$
  remove from E' every edge incident on α or β
end
return *VC*

Exercise 3.28: Arbitrariness of the constant in the definition of BPP

Suppose k is a fixed constant, $1/2 < k \leq 1$. Suppose L is a language such that there exists a Turing machine M with the property that whenever $x \in L$, M accepts x with probability at least k, and whenever $x \notin L$, M rejects x with probability at least k. Show that $L \in \mathbf{BPP}$.

Exercise 3.29: Fredkin gate is self-inverse

Show that applying two consecutive Fredkin gates gives the same outputs as inputs.

Solution

Concepts Involved: Fredkin Gates. Recall the input/output table of the Fredkin gate:

Inputs			Outputs		
a	b	c	a'	b'	c'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	1	1	0
1	1	1	1	1	1

We check for all possible 8 input states that applying the Fredkin gate returns the original input state.

$$
\begin{aligned}
F[F[(0,0,0)]] &= F[(0,0,0)] = (0,0,0) \\
F[F[(0,0,1]] &= F[(0,0,1)] = (0,0,1) \\
F[F[(0,1,0]] &= F[(0,1,0)] = (0,1,0) \\
F[F[(0,1,1)]] &= F[(1,0,1)] = (0,1,1) \\
F[F[(1,0,0)]] &= F[(1,0,0)] = (1,0,0) \\
F[F[(1,0,1)]] &= F[(0,1,1)] = (1,0,1) \\
F[F[(1,1,0)]] &= F[(1,1,0)] = (1,1,0) \\
F[F[(1,1,1)]] &= F[(1,1,1)] = (1,1,1)
\end{aligned}
$$

We conclude that the Fredkin gate is self-inverse. □

Exercise 3.30

Verify that the billiard ball computer in Figure 3.14 computes the Fredkin gate.

Exercise 3.31: Reversible half-adder

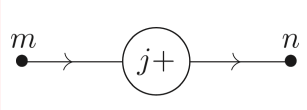
Construct a reversible circuit which, when two bits x and y are input, outputs $(x, y, c, x \oplus y)$, where c is the carry bit when x and y are odd.

Exercise 3.32: From Fredkin to Toffoli and back again

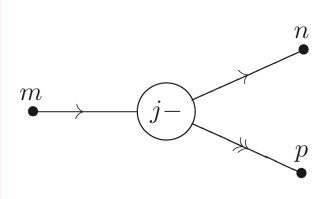
What is the smallest number of Fredkin gates needed to simulate a Toffoli gate? What is the smallest number of Toffoli gates needed to simulate a Fredkin gate?

Problem 3.1: Minsky machines

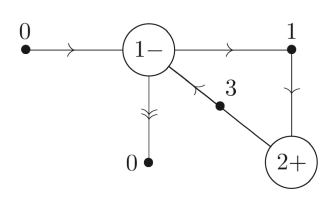
A *Minsky machine* consists of a finite set of *registers*, $r_1, r_2, \ldots, r_k$, each capable of holding an arbitrary non-negative integer, and a *program*, made up of *orders* of one of two types. The first type has the form:



The interpretation is that at point m in the program register rj is incremented by one, and execution proceeds to point n in the program. The second type of order has the form:



The interpretation is that at point m in the program, register r_j is decremented if it contains a positive integer, and execution proceeds to point n in the program. If register r_j is zero then execution simply proceeds to point p in the program. The *program* for the Minsky machine consists of a collection of such orders, of a form like:



The starting and all possible halting points for the program are conventionally labeled zero. This program takes the contents of register r_1 and adds them to register r_2, while decrementing r_1 to zero.

(1) Prove that all (Turing) computable functions can be computed on a Minsky machine, in the sense that given a computable function $f(\cdot)$ there is a Minsky machine program that when the registers start in the state $(n, 0, \ldots, 0)$ gives as output $(f(n), 0, \ldots, 0)$.

(2) Sketch a proof that any function which can be computed on a Minsky machine, in the sense just defined, can also be computed on a Turing machine.

Problem 3.2: Vector games

A *vector game* is specified by a finite list of vectors, all of the same dimension, and with integer co-ordinates. The game is to start with a vector x of non-negative integer co-ordinates and to add to x the first vector from the list which preserves the non-negativity of all the components, and to repeat this process until it is no longer possible. Prove that for any computable function $f(\cdot)$ there is a vector game which when started with the vector $(n, 0, \ldots, 0)$ reaches $(f(n), 0, \ldots, 0)$ (*Hint:* Show that a vector game in $k+2$ dimensions can simulate a Minsky machine containing k registers.)

Problem 3.3: Fractran

A *Fractran* program is defined by a list of positive rational numbers $q_1, \ldots, q_n$. It acts on a positve integer m by replacing it by $q_i m$ where i is the least number such that $q_i m$ is an integer. If there is ever a time when there is no i such that $q_i m$ is an integer, then execution stops. Prove that for any computable function $f(\cdot)$ there is a Fractran program which when started with 2^n reaches $2^{f(n)}$ without going through any intermediate powers of 2. (*Hint:* use the previous problem.)

Problem 3.4: Undecidability of dynamical systems

A Fractran program is essentially just a very simple dynamical system taking positive integers to positive integers. Prove that there is no algorithm to decide whether such a dynamical system ever reaches 1.

Problem 3.5: Non-universality of two bit reversible logic

Suppose we are trying to build circuits using only one and two bit reversible logic gates, and ancilla bits. Prove that there are Boolean functions which cannot be computed in this fashion. Deduce that the Toffoli gate cannot be simulated using one and two bit reversible gates, even with the aid of ancilla bits.

Problem 3.6: Hardness of approximation of TSP

Let $r \geq 1$ and suppose that there is an approximation algorithm for TSP which is guaranteed to find the shorted tour among n cities to within a factor r. Let $G = (V, E)$ be any graph on n vertices. Define an instance of TSP by identifying cities with vertices in V, and deefinign the distance between cities i and j to be 1 if (i, j) is an edge of G, and to be $\lceil r \rceil |V| + 1$ otherwise. Show that if the approximation algorithm is applied to this instance of TSP then it returns a Hamiltonian cycle for G if one exists, and otherwise returns a tour of length more than $\lceil r \rceil |V|$. From the **NP**-completeness of HC it follows that no such approximation algorithm can exist unless $\mathbf{P} = \mathbf{NP}$.

Problem 3.7: Reversible Turing machines

(1) Explain how to construct a reversible Turing machine that can compute the same class of functions as is computable on an ordinary Turing machine. (*Hint:* It may be helpful to use a multi-tape construction.)

(2) Give general space and time bounds for the operation of your reversible Turing machine, in terms of the time $t(x)$ and space $s(x)$ required on an ordinary single-tape Turing machine to compute a function $f(x)$.

Problem 3.8: Find a hard-to-compute class of functions (Research)

Find a natural class of functions on n inputs which requires a super-polynomial number of Boolean gates to compute.

Solution

Concepts Involved:

To the authors' knowledge, this remains an open problem. □

Problem 3.9: Reversible $\mathbf{PSPACE} = \mathbf{PSPACE}$

It can be shown that the problem 'quantified satisfiability', or QSAT, is $\mathbf{PSPACE}$-complete. That is, every other language in $\mathbf{PSPACE}$ can be reduced to QSAT in polynomial time. The language QSAT is defined to consist of all Boolean formulae φ in n variables $x_1, \ldots, x_n$, and in conjunctive normal form, such that:

$$\exists_{x_1} \forall_{x_2} \exists_{x_3} \ldots \forall_{x_n} \varphi \text{ if } n \text{ is even;}$$
$$\exists_{x_1} \forall_{x_2} \exists_{x_3} \ldots \exists_{x_n} \varphi \text{ if } n \text{ is odd.}$$

Prove that a reversible Turing machine operating in polynomial space can be used to solve QSAT. Thus, the class of languages decidable by a computer operating reversibly in polynomial space is equal to $\mathbf{PSPACE}$.

Problem 3.10: Ancilla bits and efficiency of reversible computation

Let p_m be the mth prime number. Outline the construction of a reversible circuit which, upon the input of m and n such that $n > m$, outputs the product $p_m p_n$, that is $(m, n) \mapsto (p_m p_n, g(m, n))$ where $g(m, n)$ is the final state of the ancilla bits used by the circuit. Estimate the number of ancilla qubits your circuit requires. Prove that if a polynomial (in $\log n$) size reversible circuit can be found that uses $O(\log(\log n))$ ancilla bits then the problem of factoring a product of two prime numbers is in $\mathbf{P}$.

5 The quantum Fourier transform and its applications

Exercise 5.1

Give a direct proof that the linear transformation defined by Equation (5.2) is unitary.

Solution

Concepts Involved: Unitary matrices, Roots of Unity

Let the linear transformation U be defined on the computational basis by

$$U_{kj} = \frac{1}{\sqrt{N}} e^{2\pi i jk/N}$$

To show that U is unitary, we compute the matrix product $U^\dagger U$. The matrix elements are

$$(U^\dagger U)_{j\ell} = \sum_{k=0}^{N-1} U_{kj}^* U_{k\ell} = \frac{1}{N} \sum_{k=0}^{N-1} e^{2\pi i k(\ell - j)/N}$$

This is a finite geometric series of N-th roots of unity. Its sum is

$$\sum_{k=0}^{N-1} e^{2\pi i k(\ell-j)/N} = \begin{cases} N & \text{if } j = \ell \\ 0 & \text{if } j \neq \ell \end{cases}$$

Therefore,

$$(U^\dagger U)_{j\ell} = \delta_{j\ell}$$

and so $U^\dagger U = I$, proving that U is unitary. □

Exercise 5.2

Explicitly compute the Fourier transform of the n qubit state $|00\ldots0\rangle$.

Solution

Concepts Involved: Quantum Fourier Transform (QFT), Computational Basis.

Let $N = 2^n$ and consider the n-qubit state $|0\rangle = |00\ldots0\rangle$. The QFT acts on computational basis states $|x\rangle$ as:

$$\mathrm{QFT}_N |x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i xy/N} |y\rangle$$

For $x = 0$, we have $e^{2\pi i \cdot 0 \cdot y/N} = 1$, so:

$$\mathrm{QFT}_N |0\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} |y\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} |y\rangle$$

□

Remark: This is identical to applying Hadamard gates to each qubit

$$H^{\otimes n}|0\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} |y\rangle$$

Exercise 5.3: (Classical fast Fourier transform)

Suppose we wish to perform a Fourier transform of a vector containing 2^n complex numbers on a classical computer. Verify that the straightforward method for performing the Fourier transform, based upon direct evaluation of Equation (5.1) requires $\Theta(2^{2n})$ elementary arithmetic operations. Find a method for reducing this to $\Theta(n2^n)$ operations, based upon Equation (5.4).

Solution

Concepts Involved: Discrete Fourier Transform (DFT); computational complexity; Fast Fourier Transform (FFT); divide-and-conquer; binary decomposition.

Direct evaluation of the DFT from Equation (5.1)

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi i j k/N}$$

requires $\Theta(N)$ operations per output y_k, and there are $N = 2^n$ outputs, so total cost is $\Theta(N^2) = \Theta(2^{2n})$. Using the FFT idea inspired by Equation (5.4), we recursively split the input into even/odd parts. This leads to the recurrence

$$T(N) = 2T(N/2) + \Theta(N)$$

which solves to $T(N) = \Theta(N \log N) = \Theta(n2^n)$. □

Remark: The FFT is a classical algorithm that reduces exponential DFT cost to quasi-linear by exploiting symmetry in complex exponentials. It is structurally analogous to the QFT circuit decomposition.

Exercise 5.4

Give a decomposition of the controlled-R_k gate into single qubit and `CNOT` gates.

Solution

Concepts Involved: Controlled phase gate CR_k; single-qubit phase rotation; gate conjugation; universal gate set.

The controlled-R_k gate acts as

$$CR_k = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes R_k \quad \text{where} \quad R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix}$$

To decompose this using only single-qubit and CNOT gates, observe that any controlled-unitary CU can be written as

$$CU = (I \otimes V) \cdot \text{CNOT} \cdot (I \otimes V^\dagger) \cdot \text{CNOT} \quad \text{if } U = VXV^\dagger X$$

For diagonal gates R_k, this identity holds with $V = R_k^{1/2}$. Therefore

$$CR_k = (I \otimes R_k^{1/2}) \cdot \text{CNOT} \cdot (I \otimes R_k^{-1/2}) \cdot \text{CNOT}$$

This construction ensures that the target qubit is unaffected when the control is $|0\rangle$, and acquires a phase R_k when the control is $|1\rangle$, as desired. □

Remark: This decomposition exploits the identity $R_k = R_k^{1/2} X R_k^{-1/2} X$, and uses only 2 CNOTs and 2 single-qubit phase gates. It generalizes to any controlled diagonal gate.

Exercise 5.5

Give a quantum circuit to perform the inverse quantum Fourier transform.

Solution

Concepts Involved: Inverse quantum Fourier transform (QFT^{-1}), Hermitian adjoint, qubit reversal.

The inverse QFT on n qubits is the adjoint of the QFT:

$$\text{QFT}^{-1}|k\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} e^{-2\pi i jk/2^n} |j\rangle$$

To implement this in a quantum circuit:

(1) Reverse the order of the QFT circuit.

(2) Replace each gate with its Hermitian conjugate:

- $H^\dagger = H$
- $CR_k^\dagger =$ controlled-R_k^{-1}, i.e., rotation by $e^{-2\pi i/2^k}$

(3) Add SWAP gates at the end if needed to reverse qubit order.

□

Remark: This circuit uses $\Theta(n^2)$ gates. For practical implementations, small-angle controlled rotations can often be omitted to obtain an approximate inverse QFT with depth $\Theta(n \log n)$.

Exercise 5.6: Approximate quantum Fourier transform

The quantum circuit construction of the quantum Fourier transform apparently requires gates of exponential precision in the number of qubits used. However, such precision is never required in any quantum circuit of polynomial size. For example, let U be the ideal quantum Fourier transform on n qubits, and V be the transform which results if the controlled-R_k gates are performed to a precision $\Delta = 1/p(n)$ for some polynomial $p(n)$. Show that the error $E(U,V) \equiv \max_{|\psi\rangle} \left\|(U-V)|\psi\rangle\right\|$ scales as $\Theta(n^2/p(n))$, and thus polynomial precision in each gate is sufficient to guarantee polynomial accuracy in the output state.

Solution

Concepts Involved: Quantum Fourier transform (QFT)operator norm, approximation error.

Let U be the ideal QFT on n qubits, and V an approximation where each controlled-R_k gate is implemented with precision $\Delta = 1/p(n)$ for some polynomial $p(n)$. The QFT uses $\Theta(n^2)$ such gates.
By Equation (4.63) in the book, the total error from replacing each U_j by V_j satisfies:

$$\|U - V\| \leq \sum_j \|U_j - V_j\|$$

If each gate contributes at most Δ error, then:

$$E(U,V) = \max_{|\psi\rangle} \|(U-V)|\psi\rangle\| = \Theta(n^2 \cdot \Delta) = \Theta\left(\frac{n^2}{p(n)}\right)$$

□

Remark: This shows that polynomial precision per gate yields overall polynomial accuracy. In practice, one may truncate small-angle controlled-R_k gates for large k, further simplifying the implementation without exceeding acceptable error bounds.

Exercise 5.7

Additional insight into the circuit in Figure 5.2 may be obtained by showing, as you should now do, that the effect of the sequence of controlled-U operations like that in Figure 5.2 is to take the state $|j\rangle|u\rangle$ to $|j\rangle U^j|u\rangle$. (Note that this does not depend on $|u\rangle$ being an eigenstate of U.)

Exercise 5.8

Suppose the phase estimation algorithm takes the state $|0\rangle|u\rangle$ to the state $|\tilde{\varphi_u}\rangle|u\rangle$, so that given the input $|0\rangle(\sum_u c_u|u\rangle)$, the algorithm outputs $\sum_u c_u|\tilde{\varphi_u}\rangle|u\rangle$. Show that if t is chosen according to (5.35), then the probability for measuring φ_u accurate to n bits at the conclusion of the phase estimation algorithm is at least $|c_u|^2(1-\epsilon)$.

Solution

Concepts Involved: Phase estimation, measurement accuracy, linearity of quantum circuits.

Given input $|0\rangle \sum_u c_u |u\rangle$, the phase estimation algorithm outputs $\sum_u c_u |\tilde{\varphi}_u\rangle |u\rangle$. For each eigenstate $|u\rangle$, the probability of measuring an estimate $\tilde{\varphi}_u$ accurate to n bits is at least $1 - \varepsilon$, provided $t = n + \lceil \log(2 + 1/2\varepsilon) \rceil$ (Equation 5.35). By linearity, the total probability of measuring such an estimate and collapsing to $|u\rangle$ is at least $|c_u|^2(1 - \varepsilon)$. □

Remark: This ensures that approximate phase estimation remains reliable even on superposition inputs, with per-eigenstate success scaling with $|c_u|^2$.

Exercise 5.9

Let U be a unitary transform with eigenvalues ± 1, which acts on a state $|\psi\rangle$. Using the phase estimation procedure, construct a quantum circuit to collapse $|\psi\rangle$ into one or the other of the two eigenspaces of U, giving also a classical indicator as to which space the final state is in. Compare your result to Exercise 4.34.

Exercise 5.10

Show that the order of $x = 5$ modulo $N = 21$ is 6.

Solution

Concepts Involved: Modular arithmetic, Euler's theorem.

We seek the smallest r such that $5^r \equiv 1 \mod 21$. Compute successive powers of 5 modulo 21:

$$\begin{aligned} 5^1 &\equiv 5 \mod 21 \\ 5^2 &\equiv 25 \equiv 4 \mod 21 \\ 5^3 &\equiv 20 \mod 21 \\ 5^4 &\equiv 100 \equiv 16 \mod 21 \\ 5^5 &\equiv 80 \equiv 17 \mod 21 \\ 5^6 &\equiv 85 \equiv 1 \mod 21 \end{aligned}$$

Since 6 is the smallest such integer, the order of $5 \mod 21$ is 6. □

Remark: This confirms that 5 generates a cyclic subgroup of order 6 in $\mathbb{Z}_{21}^*$, consistent with $\varphi(21) = 12$, since $r = 6 \mid \varphi(21)$.

Exercise 5.11

Show that the order of x satisfied $r \leq N$.

Solution

Concepts Involved: Modular arithmetic, Euler's theorem, Lagrange's theorem, group theory.

Let $x \in \mathbb{Z}_N^*$, and let r be the order of $x \mod N$, i.e., the smallest positive integer such that $x^r \equiv 1 \mod N$.

Then x lies in the multiplicative group $\mathbb{Z}_N^*$, which has $\varphi(N)$ elements. By Lagrange's theorem, the order r must divide $\varphi(N)$, so:

$$r \leq \varphi(N)$$

Since $\varphi(N) < N$ for all $N > 2$, it follows that:

$$r \leq \varphi(N) < N \Rightarrow r \leq N$$

□

Remark: This bound is useful when searching for the order in algorithms such as Shor's, as it ensures the period must lie in the finite range $1 \leq r \leq N$.

Exercise 5.12

Show that U is unitary (*Hint:* x is co-prime to N, and therefore has an inverse modulo N).

Exercise 5.13

Prove (5.44) (*Hint:* $\sum_{s=0}^{r-1} \exp\left(-2\pi isk/r\right) = r\delta_{k0}$.) In fact, prove that

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{2\pi isk/r} \left|u_s\right\rangle = \left|x^k \bmod N\right\rangle$$

Exercise 5.14

The quantum state produced in the order-finding algorithm, before the inverse Fourier transform, is

$$\left|\psi\right\rangle = \sum_{j=0}^{2^t-1} \left|j\right\rangle U^j \left|1\right\rangle = \sum_{j=0}^{2^t-1} \left|j\right\rangle \left|x^j \bmod N\right\rangle,$$

if we initialize the second register as $\left|1\right\rangle$. Show that the same state is obtained if we replace U^j with a *different* unitary transform V, which computes

$$V\left|j\right\rangle\left|k\right\rangle = \left|j\right\rangle\left|k + x^k \bmod N\right\rangle$$

and start the second register in the state $\left|0\right\rangle$. Also show how to construct V using $O(L^3)$ gates.

Exercise 5.15

Show that the least common multiple of positive integers x and y is $xy/\gcd(x, y)$, and thus may be computed in $O(L^2)$ operations if x and y are L bit numbers.

Exercise 5.16

For $x \geq 2$ prove that $\int_x^{x+1}(1/y^2)dy \geq 2/3x^2$. Show that

$$\sum_q \frac{1}{q^2} \leq \frac{3}{2}\int_2^\infty \frac{1}{y^2}dy = \frac{3}{4},$$

and thus that (5.58) holds.

Solution

Concepts Involved: Definite integrals; monotonic decreasing functions, Series convergence.

For $x \geq 2$,

$$\int_x^{x+1} \frac{1}{y^2}dy = \frac{1}{x} - \frac{1}{x+1} = \frac{1}{x(x+1)} \geq \frac{2}{3x^2}$$

since $\frac{1}{x(x+1)} \geq \frac{2}{3x^2}$ holds iff $x \geq 2$.
Thus, for all integers $q \geq 2$,

$$\frac{1}{q^2} \leq \frac{3}{2}\int_q^{q+1} \frac{1}{y^2}dy \Rightarrow \sum_{q=2}^\infty \frac{1}{q^2} \leq \frac{3}{2}\int_2^\infty \frac{1}{y^2}dy = \frac{3}{2}\cdot\frac{1}{2} = \frac{3}{4}$$

□

Remark: This confirms the bound in equation (5.58), ensuring that the tail of the $\sum 1/q^2$ series contributes a controlled, finite error in approximating quantum Fourier transforms.

Exercise 5.17

Suppose N is L bits long. The aim of this exercise is to find an efficient classical algorithm to determine whether $N = a^b$ for some integers $a \geq 1$ and $b \geq 2$. This may be done as follows:

(1) Show that b, if it exists, satisfies $b \leq L$.

(2) Show that if takes at most $O(L^2)$ operations to compute $\log_2 N$, $x = y/b$ for $b \leq L$, and the two integers u_1 and u_2 nearest to 2^x.

(3) Show that it takes at most $O(L^2)$ operations to compute u_1^b and u_2^b (use repeated squaring) and check to see if either is equal to N.

(4) Combine the previous results to give an $O(L^3)$ operations algorithm to determine whether $N = a^b$ for integers a and b.

Exercise 5.18: Factoring 91

Suppose we wish to factor $N = 91$. Confirm that steps 1 and 2 are passed. For step 3, suppose we choose $x = 4$, which is co-prime to 91. Compute the order r of x with respect to N, and show that $x^{r/2} \mod 91 = 64 \neq -1 \pmod{91}$, so the algorithm succeeds, giving $\gcd(64 - 1, 19) = 7$.
It is unlikely that this is the most efficient method you've seen for factoring 91. Indeed, if all computations had to be carried out on a classical computer, this reduction would not result in an efficient factoring algorithm, as no efficient method is known for solving the order-finding problem on a classical computer.

Exercise 5.19

Show that $N = 15$ is the smallest number for which the order-finding subroutine is required, that is, it is the smallest composite number that is not even or a power of some smaller integer.

Exercise 5.20

Suppose $f(x + r) = f(x)$, and $0 \leq x < N$, for N an integer multiple of r. Compute

$$\hat{f}(l) \equiv \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} e^{-2\pi i l x/N} f(x)$$

and relate the result to (5.63). You will need to use the fact that

$$\sum_{k \in \{0, r, 2r, \ldots, N-r\}} e^{2\pi i k l/N} = \begin{cases} \sqrt{N/r} & \text{if } l \text{ is an integer multiple of } N/r \\ 0 & \text{otherwise.} \end{cases}$$

Exercise 5.21: Period-finding and phase estimation

Suppose you are given a unitary opeprator U_y which performs the transformation $U_y |f(x)\rangle = |f(x + y)\rangle$, for the periodic function described above.

(1) Show that the eigenvectors of U_y are $|\hat{f}(l)\rangle$, and calculate their eigenvalues.

(2) Show taht given $|f(x_0)\rangle$ for some x_0, U_y can be used to realize a black box which is as useful as U in solving the period-finding problem.

Exercise 5.22

Show that

$$|\hat{f}(l_1, l_2)\rangle = \sum_{x_1=0}^{r-1} \sum_{x_2=0}^{r-1} e^{-2\pi i (l_1 x_1 + l_2 x_2)/r} |f(x_1, x_2)\rangle = \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} e^{-2\pi i l_2 j/r} |f(0, j)\rangle ,$$

and we are constrained to have $l_1/s - l_2$ be an integer multiple of r for this expression to be non-zero.

Exercise 5.23

Compute

$$\frac{1}{r}\sum_{l_1=0}^{r-1}\sum_{l_2=0}^{r-1} e^{-2\pi i(l_1x_1+l_2x_2)/r}\,|\hat{f}(l_1,l_2)\rangle$$

using (5.70), and show that the result is $f(x_1, x_2)$.

Exercise 5.24

Construct the generalized continued fractions algorithm needed in step 6 of the discrete logarithm algorithm to determine s from estimates of sl_2/r and l_2/r.

Exercise 5.25

Construct a quantum circuit for the black box U used in the quantum discrete logarithm algorithm, which takes a and b as parameters, and performs the unitary transform $|x_1\rangle|x_2\rangle|y\rangle \mapsto |x_1\rangle|x_2\rangle|y \oplus b^{x_1}a^{x_2}\rangle$. How many elementary operations are required?

Exercise 5.26

Since K is a subgroup of G, when we decompose G into a product of cyclic groups of prime power order, this also decomposes K. Re-express (5.77) to show that determining l_i' allows one to sample from the corresponding cyclic subgroup K_{p_i} of K

Exercise 5.27

Of course, the decomposition of a general finite Abelian group G into a product of cyclic groups of prime power order is usually a difficult problem (at least as hard as factoring integers, for example). Here, quantum algorithms come to the rescue again: explain how the algorithms in this chapter can be used to efficiently decompose G as desired.

Exercise 5.28

Write out a detailed specification of the quantum algorithm to solve the hidden subgroup problem, complete with runtime and success probability estimates, for finite Abelian groups.

Exercise 5.29

Give quantum algorithms to solve the Deutsch and Simon problems listed in Figure 5.5, using the framework of the hidden subgroup problem.

Problem 5.1

Construct a quantum circuit to perform the quantum Fourier transform

$$|j\rangle \mapsto \frac{1}{\sqrt{p}} \sum_{k=0}^{p-1} e^{2\pi i j k/p} |k\rangle$$

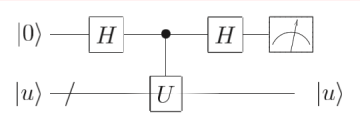
where p is prime.

Problem 5.2: Measured quantum Fourier transform

Suppose the quantum Fourier transform is performed as the last step of a quantum computation, followed by a measurement in the computational basis. Show that the combination of quantum Fourier transform and measurement is equivalent to a circuit consisting entirely of *one* qubit gates and measurement, with classical control, and no two qubit gates. You may find the discussion of Section 4.4 useful.

Problem 5.3: Kitaev's algorithm

Consider the quantum circuit



where $|u\rangle$ is an eigenstate of U with eigenvalue $e^{2\pi i \varphi}$. Show that the top qubit is measured to be 0 with probability $p \equiv \cos^2(\pi\varphi)$. Since the state $|u\rangle$ is unaffected by the circuit it may be reused; if U can be replaced by U^k, where k is an arbitrary integer under your control, show that by repeating this circuit and increasing k appropriately, you can efficiently obtain as many bits of p as desired, and thus, of φ. This is an alternative to the phase estimation algorithm.

Problem 5.4

The runtime bound $O(L^3)$ we have given for the factoring algorithm is not tight. Show that a better upper bound of $O(L^2 \log L \log\log L)$ operations can be achieved.

Problem 5.5: Non-Abelian hidden subgroups (Research)

Let f be a function on a finite group G to an arbitrary finite range X, which is promised to be constant and distinct on distinct left cosets of a subgroup K. State with the state

$$\frac{1}{\sqrt{|G|^m}} \sum_{g_1,\ldots,g_m} |g_1,\ldots,g_m\rangle\ |f(g_1),\ldots,f(g_m)\rangle$$

and prove that picking $m = 4\log|G| + 2$ allows K to be identifies with probability at least $1 - 1/|G|$. Note that G does not necessarily have to be Abelian, and being able to perform a Fourier transform over G is not required. This result shows that one can produce (using only $O(\log|G|)$ oracle calls) a final result in which the pure state outcomes corresponding to different possible hidden subgroups are nearly

orthogonal. However, it is unknown whether a POVM exists or not which allows the hidden subgroup to be identified *efficiently* (i.e. using $\mathrm{poly}(\log |G|)$ operations) from this final state.

Problem 5.6: Addition by Fourier transforms

Consider the task of constructing a quantum circuit to compute $|x\rangle \mapsto |x + y \mod 2^n\rangle$, where y is a fixed constant, and $0 \leq x < 2^n$. Show that one efficient way to do this, for values of y such as 1, is to first perform a quantum Fourier transform, then to apply single qubit phase shifts, then an inverse Fourier transform. What values of y can be added easily this way, and how many operations are required?

6 Quantum search algorithms

Exercise 6.1

Show that the unitary operator corresponding to the phase shift in the Grover iteration is $2\left|0\right\rangle\!\left\langle 0\right| - I$.

Solution

Concepts Involved: Unitary matrices, Grover Search

We wish to find the unitary phase shift operator U whose action on a computational basis state $\left|x\right\rangle$ is:

$$U\left|x\right\rangle = -(-1)^{\delta_{x0}}\left|x\right\rangle$$

Let's check that $2\left|0\right\rangle\!\left\langle 0\right| - I$ does the trick:

$$\left(2\left|0\right\rangle\!\left\langle 0\right| - I\right)\left|x\right\rangle = 2\left\langle 0|x\right\rangle\left|0\right\rangle - \left|x\right\rangle = 2\delta_{x0}\left|0\right\rangle - \left|x\right\rangle = \left(2\delta_{x0} - 1\right)\left|x\right\rangle = -(-1)^{\delta_{x0}}\left|x\right\rangle.$$

□

Exercise 6.2

Show that the operation $\left(2\left|\psi\right\rangle\!\left\langle\psi\right| - I\right)$ applied to a general state $\sum_k \alpha_k\left|k\right\rangle$ produces

$$\sum_k \left[-\alpha_k + 2\left\langle\alpha\right\rangle\right]\left|k\right\rangle$$

where $\left\langle\alpha\right\rangle \equiv \sum_k \alpha_k/N$ is the mean value of the α_k. For this reason, $\left(2\left|\psi\right\rangle\!\left\langle\psi\right| - I\right)$ is sometimes reffered to as the *inversion about mean* operation.

Solution

Concepts Involved: Grover Search

Appplying the operator yields:

$$\begin{aligned}
\left(2\left|\psi\right\rangle\!\left\langle\psi\right| - I\right)\sum_k \alpha_k\left|k\right\rangle &= 2\sum_k \alpha_k\left\langle\psi|k\right\rangle\left|\psi\right\rangle - \sum_k \alpha_k\left|k\right\rangle \\
&= 2\sum_k \alpha_k \frac{1}{N^{1/2}}\left|\psi\right\rangle - \sum_k \alpha_k\left|k\right\rangle \\
&= 2\left\langle\alpha\right\rangle N^{1/2}\left|\psi\right\rangle - \sum_k \alpha_k\left|k\right\rangle \\
&= 2\left\langle\alpha\right\rangle N^{1/2}\frac{1}{N^{1/2}}\sum_k\left|k\right\rangle - \sum_k \alpha_k\left|k\right\rangle \\
&= \sum_k[-\alpha_k + 2\left\langle\alpha\right\rangle]\left|k\right\rangle
\end{aligned}$$

□

Exercise 6.3

Show that in the $|\alpha\rangle, |\beta\rangle$ basis, we may write the Grover iteration as

$$G = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix},$$

where θ is a real number in the range 0 to $\pi/2$ (assuming for simplicity that $M \leq N/2$; this limitation will be lifted shortly), chosen so that

$$\sin\theta = \frac{2\sqrt{M(N-M)}}{N}.$$

Solution

Concepts Involved: Grover Search

The action of the oracle O is to leave non-solutions invariant and attach a minus sign to the solutions, so $|\alpha\rangle \to |\alpha\rangle$ and $|\beta\rangle \to -|\beta\rangle$, hence in the $|\alpha\rangle, |\beta\rangle$ basis:

$$O \cong \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Now if we express:

$$|\psi\rangle = \sqrt{\frac{N-M}{N}}\,|\alpha\rangle + \sqrt{\frac{M}{N}}\,|\beta\rangle$$

We find:

$$|\psi\rangle\langle\psi| = \frac{N-M}{N}\,|\alpha\rangle\langle\alpha| + \frac{\sqrt{M(N-M)}}{N}(|\alpha\rangle\langle\beta| + |\beta\rangle\langle\alpha|) + \frac{M}{N}\,|\beta\rangle\langle\beta|$$

Hence:

$$2\,|\psi\rangle\langle\psi| - I \cong \begin{bmatrix} 2\frac{N-M}{N} - 1 & 2\frac{\sqrt{M(N-M)}}{N} \\ 2\frac{\sqrt{M(N-M)}}{N} & 2\frac{M}{N} - 1 \end{bmatrix} = \begin{bmatrix} \frac{N-2M}{N} & 2\frac{\sqrt{M(N-M)}}{N} \\ 2\frac{\sqrt{M(N-M)}}{N} & -\frac{N-2M}{N} \end{bmatrix}$$

Therefore:

$$G = (2\,|\psi\rangle\langle\psi| - I)O \cong \begin{bmatrix} \frac{N-2M}{N} & 2\frac{\sqrt{M(N-M)}}{N} \\ 2\frac{\sqrt{M(N-M)}}{N} & -\frac{N-2M}{N} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} \frac{N-2M}{N} & -2\frac{\sqrt{M(N-M)}}{N} \\ 2\frac{\sqrt{M(N-M)}}{N} & \frac{N-2M}{N} \end{bmatrix}$$

Choosing θ such that:

$$\sin\theta := 2\frac{\sqrt{M(N-M)}}{N}$$

We find the cosine to be:

$$\cos\theta = \sqrt{1-\sin^2\theta} = \sqrt{\frac{N^2}{N^2} - 4\frac{M(N-M)}{N^2}} = \sqrt{\frac{(N-2M)^2}{N^2}} = \frac{N-2M}{N}$$

Therefore we can conclude:

$$G \cong \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

□

Exercise 6.4

Give explicit steps for the quantum search algorithm, as above, but for the case of multiple solutions $(1 < M < N/2)$.

Solution

Concepts Involved: Grover Search

Algorithm: Quantum Search (Multiple Solutions)

- **Inputs:** (1) A black box oracle O which performs the transformation $O|x\rangle|q\rangle = |x\rangle|q \oplus f(x)\rangle$ where $f(x) = 0$ for all $0 \leq x < 2^n$ except for $x \in S = \{x_0, x_1, \ldots x_M\}$ $(1 < M < N/2)$ for which $f(x) = 1$. (2) $n+1$ qubits in the state $|0\rangle$.
- Outputs: $S = \{x_0, x_1, \ldots, x_M\}$
- Runtime: $O(M\sqrt{2^n})$ operations. Succeeds with probability $O(1)$.
- Procedure:
  (1) $|0\rangle^{\otimes n}|0\rangle$ (initial state)
  (2) $\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$ (apply $H^{\otimes n}$ to first n qubits, HX to the last qubit)
  (3) $[(2|\psi\rangle\langle\psi| - 1)O\prod_{i\in A}C_i^n(X)]^R\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] \approx |\beta\rangle\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$ (Apply the modified Grover iteration $R \approx \lceil\pi\sqrt{2^n}/4\rceil$ times. The modification from the usual Grover iteration is the $\prod_{i\in A}C_i^n(X)$. This corresponds to a controlled-operation over the set of already found solutions, such that the oracle does not toggle on solutions already found. The resulting state $|\beta\rangle$ is the uniform superposition over solutions still not found, in $S \setminus A$)
  (4) $\rightarrow x \in S \setminus A$ (measure the first n qubits - $|\beta\rangle$ is a uniform superposition of the (not yet found) solution computational basis states, and so we obtain one of the solutions in this way).
  (5) Add x to A and repeat from step (1) until all solutions are found.

□

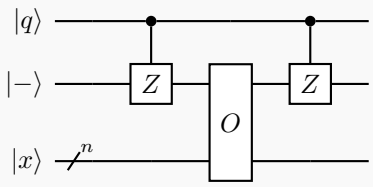
Exercise 6.5

Show that the augmented oracle O' may be constructed using one application of O, and elementary quantum gates, using the extra qubit $|q\rangle$.

Solution

Concepts Involved: Grover Search

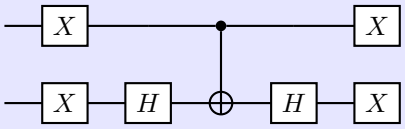
The circuit:



accomplishes the task of being the augmented oracle O'. Suppose $|q\rangle = |0\rangle$ - then the controlled-Z operations have no effect/drop out of the circuit, and the has the desired action of $O|x\rangle|-\rangle \rightarrow (-1)^{f(x)}|-\rangle$. Suppose instead that $|q\rangle = |1\rangle$. Then the controlled-Z flips $|-\rangle \rightarrow |+\rangle$, and therein the oracle has no effect $O|x\rangle|+\rangle = |x\rangle|+\rangle$ (with the $|+\rangle$ being flipped back to $|-\rangle$ by the final controlled Z). Hence, we have shown that the given O' marks an item only if x is a solution to the search problem and the extra bit q is set to zero. □

Exercise 6.6

Verify that the gates in the dotted box in the second figure of Box 6.1 perform the conditional phase shift operation $2|00\rangle\langle 00| - I$, up to an unimportant global phase factor.



Solution

Concepts Involved: Grover Search

Let us deduce the net unitary performed by the gates in the dotted box (we do the matrix multiplications

in block form):

$$\begin{aligned}(X_1 \otimes X_2)(I_1 \otimes H_2)\mathsf{CNOT}_{1,2}(I_1 \otimes H_2)(X_1 \otimes X_2) &\cong \begin{bmatrix} 0 & X \\ X & 0 \end{bmatrix}\begin{bmatrix} H & 0 \\ 0 & H \end{bmatrix}\begin{bmatrix} I & 0 \\ 0 & X \end{bmatrix}\begin{bmatrix} H & 0 \\ 0 & H \end{bmatrix}\begin{bmatrix} 0 & X \\ X & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & XH \\ XH & 0 \end{bmatrix}\begin{bmatrix} I & 0 \\ 0 & X \end{bmatrix}\begin{bmatrix} 0 & HX \\ HX & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & XH \\ XH & 0 \end{bmatrix}\begin{bmatrix} 0 & HX \\ XHX & 0 \end{bmatrix} \\ &= \begin{bmatrix} XHXHX & 0 \\ 0 & XHHX \end{bmatrix}\end{aligned}$$

Using that $X^2 = H^2 = I$ (Exercises **??**, **??**), the bottom right entry is:

$$XHHX = XIX = XX = I$$

And using that $HXH = Z$ (Exercise **??**) and $XZ = -ZX$ (Exercise **??**):

$$XHXHX = XZX = -ZXX = -Z$$

We thus find:

$$(X_1 \otimes X_2)(I_1 \otimes H_2)\mathsf{CNOT}_{1,2}(I_1 \otimes H_2)(X_1 \otimes X_2) \cong \begin{bmatrix} -Z & 0 \\ 0 & I \end{bmatrix} = -\mathsf{diag}(1,-1,-1,-1)$$

Which agrees with:

$$2\left|00\right\rangle\!\left\langle 00\right| - I \cong \mathsf{diag}(1,-1,-1,-1)$$

up to a global phase. □

Exercise 6.7

Verify that the circuits shown in Figures 6.4 and 6.5 implement the operations $\exp\big(-i\,|x\rangle\langle x|\,\Delta t\big)$ and $\exp\big(-i\,|\psi\rangle\langle\psi|\,\Delta t\big)$, respectively, with $|\psi\rangle$ as in (6.24).

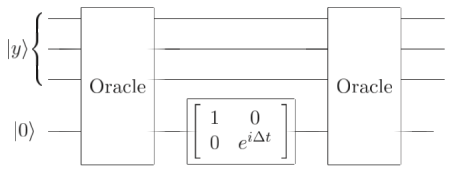


Figure 6.4. Circuit implementing the operation $\exp(-i|x\rangle\langle x|\Delta t)$ using two oracle calls.

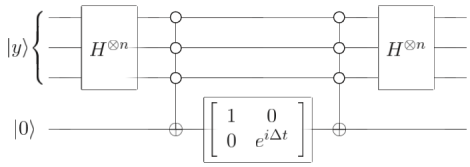


Figure 6.5. Circuit implementing the operation $\exp(-i|\psi\rangle\langle\psi|\Delta t)$, for $|\psi\rangle$ as in (6.24).

Exercise 6.8

Suppose the simulation step is performed to accuracy $O(\Delta t^r)$. Show that the number of oracle calls required to simulate H to reasonable accuracy is $O(N^{r/2(r-1)})$. Note that as r becomes large the exponent of N approaches $1/2$.

Solution

Concepts Involved: Grover Search, Accuracy

The number of total steps is:

$$O(t/\Delta t) = O(\sqrt{N}/\Delta t)$$

and hence the total error is:

$$O(\Delta t^r \cdot \sqrt{N}/\Delta t) = O(\Delta t^{r-1}\sqrt{N})$$

We want the error to be $O(1)$, and hence we choose $\Delta t = \Theta(N^{-\frac{1}{2(r-1)}})$. The total number of oracle calls is then $O(N^{\frac{r}{2(r-1)}})$, as claimed. □

Exercise 6.9

Verify Equation (6.25). (*Hint:* see Exercise 4.15.)

Exercise 6.10

Show that by choosing Δt appropriately we can obtain a quantum search algorithm which uses $O(\sqrt{N})$ queries, and for which the final state is $|x\rangle$ exactly, that is, the algorithm works with probability 1, rather than with some smaller probability.

Exercise 6.11: Multiple solution continuous quantum search

Guess a Hamiltonian with which one may solve the continuous time search problem in the case where the search problem has M solutions.

Exercise 6.12: Alternative Hamiltonian for quantum search

Suppose

$$H = |x\rangle\langle\psi| + |\psi\rangle\langle x|$$

(1) Show that it takes time $O(1)$ to rotate from the state $|\psi\rangle$ to the state $|x\rangle$, given an evolution according to the Hamiltonian H.

(2) Explain how a quantum simulation of the Hamiltonian H may be performed, and determine the number of oracle calls your simulation technique requires to obtain the solution with high probability.

Exercise 6.13

Consider a classical algorithm for the counting problem which samples uniformly and independently k times from the search space, and let $X_1, \ldots, X_k$ be the results of the oracle calls, that is, $X_j = 1$ if the j th oracle call revealed a solution to the problem, and $X_j = 0$ if the j th oracle call did not reveal a solution to the problem. This algorithm returns the estimate $S \equiv N \times \sum_j X_j/k$ for the number of solutions to the search problem. Show that the standard deviation in S is $\Delta S = \sqrt{M(N-M)/k}$. Prove that to obtain a probability at least $3/4$ of estimating M correctly to within an accuracy $\sqrt{M}$ for all values of M we must have $k = \Omega(N)$

Exercise 6.14

Prove that *any* classical counting algorithm with a probability at least $3/4$ for estimating M correctly to within an accuracy $c\sqrt{M}$ for some constant c and for all values of M must make $\Omega(N)$ oracle calls.

Exercise 6.15

Use the Cauchy-Schwarts inequality to show that for any normalized state vector $|\psi\rangle$ and set of N orthormal basis vectors $|x\rangle$,

$$\sum_x \||\psi\rangle - |x\rangle\|^2 \geq 2N - 2\sqrt{N}.$$

Exercise 6.16

Suppose we merely require that the probability of an error being made is less than $1/2$ when averaged uniformly over the possible values for x, instead of for all values of x. Show that $O(\sqrt{N})$ oracle calls are still required to solve the search problem.

Exercise 6.17: Optimality for multiple solutions

Suppose the search problem has M solutions. Show that $O(\sqrt{N/M})$ oracle applications are required to find a solution.

Exercise 6.18

Prove that the minimum degree polynomial representing a Boolean function $F(X)$ is unique.

Exercise 6.19

Show that $P(X) = 1 - (1 - X_0)(1 - X_1) \ldots (1 - X_{N-1})$ represents OR.

Exercise 6.20

Show that $Q_0(\text{OR}) \geq N$ by constructing a polynomial which represents the OR function from the output of a quantum circuit which computes OR with zero error.

Problem 6.1: Finding the minimum

Suppose $x_1, \ldots x_N$ is a databse of numbers held in memory, as in Section 6.5. Show that only $O(\log(N)\sqrt{N})$ accessed to the meory are required on a quantum computer, in order to find the smallest element on the list, with probability at least one-half.

Problem 6.2: Generalized quantum searching

Let $|\psi\rangle$ be a quantum state, and define $U_{|\psi\rangle} = I - 2|\psi\rangle\langle\psi|$. That is $U_{|\psi\rangle}$ gives the state $|\psi\rangle$ a -1 phase, and leaves states orthogonal to $|\psi\rangle$ invariant.

(1) Suppose we have a quantum circuit implementing a unitary operator U such that $U|0\rangle^{\otimes n} = |\psi\rangle$. Explain how to implement $U_{|\psi\rangle}$.

(2) Let $|\psi_1\rangle = |1\rangle$, $|\psi_2\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$, $|\psi_3\rangle = (|0\rangle - i|1\rangle)/\sqrt{2}$. Suppose an unknown oracle O is selected from the set $U_{|\psi_1\rangle}, U_{|\psi_2\rangle}, U_{|\psi_3\rangle}$. Give a quantum algorithm which identifies the oracle with just *one* application of the oracle. (*Hint:* consider superdense coding.)

(3) **(Research):** More generally, given k states $|\psi_1\rangle, \ldots, |\psi_k\rangle$, and an unknown oracle O selected from the set $U_{|\psi_1\rangle}, \ldots, U_{|\psi_k\rangle}$, how many oracle applications are required to identify the oracle, with high probability?

Problem 6.3: Database retrieval

Given a quantum oracle that returns $|k, y \oplus X_k\rangle$ given an n qubit query (and one scratchpad qubit) $|k, y\rangle$, show that with high probability, all $N = 2^n$ bits of X can be obtained using only $N/2 + \sqrt{N}$ queries. This implies the general upper bound $Q_2(F) \leq N/2 + \sqrt{N}$ for any F.

Problem 6.4: Quantum searching and cryptography

Quantum searching can, potentially, be used to speed up the search for cryptographic keys. The idea is to search through the space of all possible keys for decryption, in each case trying the key, and checking to see whether the decrypted message makes 'sense'. Explain why this idea doesn't work for the Vernam cipher (Section 12.6). When might it work for cryptosystems such as DES? (For a description of DES see, for example, [MvOV96] or [Sch96a].)

11 Entropy and information

Exercise 11.1: Simple calculations of entropy

What is the entropy associated with the toss of a fair coin? With the roll of a fair die? How would the entropy behave if the coin or die were unfair?

Solution

Concepts Involved:

The entropy associated with a fair coin toss X_{coin} (with $p_H = p_T = \frac{1}{2}$) is:

$$H(X_{\text{coin}}) = -\frac{1}{2}\log\left(\frac{1}{2}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right) = \log(2) = 1$$

and with a fair die X_{die} (with $p_i = \frac{1}{6}$ for $i \in \{1, 2, 3, 4, 5, 6\}$) is:

$$H(X_{\text{die}}) = 6 \cdot (-\frac{1}{6}\log\left(\frac{1}{6}\right)) = \log(6).$$

The entropy decreases if the coin/die are unfair; intuitively, this is because there is less uncertainty before we learn the result than in the fair case. For example in the limiting case, we can see that for a maximally unfair coin/die where one of the probabilities was one and the others were zero (wherein there is no uncertainty of the result whatsoever) that the entropy would be zero. For the case of the binary entropy (which we can view as a coin with weighted probability of heads p) we prove in Ex. 11.3 that indeed the fair coin of $p = 1/2$ maximizes the entropy. □

Exercise 11.2: Intuitive justification for the definition of entropy

Suppose we are trying to quantify how much information is provided by an event E which may occur in a probabilistic experiment. We do this using an 'information function' $I(E)$ whose value is determined by the event E. Suppose we make the following assumptions about this function:

- $I(E)$ is a function only of the probability of the event E, so we may write $I = I(p)$, where p is a probability in the range 0 to 1.
- I is a smooth function of probability.
- $I(pq) = I(p) + I(q)$ when $p, q > 0$. (*Interpretation:* The information gained when two independent events occur with individual probabilities p and q is the sum of the information gained from each event alone.)

Show that $I(p) = k \log p$, for some constant, k. It follows that the average information gain when one of a mutually exclusive set of events with probabilities $p_1, \ldots, p_n$ occurs is $k \sum_i p_i \log p_i$, which is just the Shanon entropy, up to a constant factor.

Solution

Concepts Involved:

Since $I(p)$ is a smooth function of p, it is differentiable (an arbitrary number of times) w.r.t. p. Now, let us consider $I(pq) = I(p) + I(q)$ and differentiate w.r.t. p, using the chain rule:

$$I'(pq) \cdot q = I'(p)$$

Now we differentiate the above w.r.t. q, using the product rule:

$$I'(pq) + I''(pq)pq = 0 \implies I''(pq) = -\frac{I'(pq)}{pq}$$

Defining $x = pq$, we obtain a differential equation for $I'(x)$:

$$I''(x) = -\frac{I'(x)}{x} \implies \frac{dI'(x)}{dx} = -\frac{I'(x)}{x} \implies \frac{dI'(x)}{I'(x)} = -\frac{dx}{x}$$

Integrating both sides, we obtain:

$$\log\big(I'(x)\big) = -\log(x) + c_1 \implies I'(x) = e^{c_1} e^{\log\left(\frac{1}{x}\right)} = \frac{e^{c_1}}{x}$$

Integrating once more:

$$I(x) = e^{c_1}\log(x) + c_2 = k\log(x) + c_2$$

where we have defined $e^{c_1} = k$. Now, using that $I(pq) = I(p) + I(q)$ for $p = q = 1$ gives us:

$$I(1) = I(1) + I(1) \implies k\log(1) + c_2 = k\log(1) + c_2 + k\log(1) + c_2 \implies c_2 = 2c_2$$

this sets $c_2 = 0$, so we conclude:

$$I(p) = k\log(p).$$

□

Exercise 11.3

Prove that the binary entropy $H_{bin}(p)$ attains its maximum value of one at $p = 1/2$

Solution

Concepts Involved:

Taking the derivative w.r.t. p using the product rule, we find:

$$\begin{aligned}\frac{\partial H_{\text{bin}}(p)}{\partial p} &= -\log(p) - p\frac{1}{p\ln(2)} - (-1)\log(1-p) - (1-p)\frac{-1}{(1-p)\ln(2)} \\ &= -(\log(p) - \log(1-p)) \\ &= -\log\left(\frac{p}{1-p}\right)\end{aligned}$$

We have an extremum when the derivative is zero, which occurs when the argument of the logarithm is one, which occurs when $p = 1/2$. To verify that this is indeed a maximum, we look at the second derivative:

$$\frac{\partial^2 H_{\text{bin}}(p)}{\partial p^2} = -\frac{1}{p(1-p)\ln(2)}$$

which evaluated at $p = 1/2$ yields $-\frac{4}{\ln(2)} < 0$, so indeed $p = 1/2$ is a maximum. For $p = 1/2$ we find that $H_{\text{bin}}(p = 1/2) = 1$ (see Ex. 11.1), thus completing the proof. □

Exercise 11.4: Concavity of the binary entropy

From Figure 11.1, it appears that the binary entropy is a concave function. Prove that this is so, that is:

$$H_{bin}(px_1 + (1-p)x_2) \geq pH_{bin}(x_1) + (1-p)H_{bin}(x_2),$$

where $0 \leq p, x_1, x_2 \leq 1$. Prove in addition that the binary entropy is strictly concave, that is, the above inequality is an equality only for the trivial cases $x_1 = x_2$ or $p = 0$ or $p = 1$.

Exercise 11.5: Sub-additivity of the Shannon entropy

Show that $H(p(x,y)\|p(x)p(y)) = H(p(x)) + H(p(y)) - H(p(x,y))$. Deduce that $H(X,Y) \leq H(X) + H(Y)$, with equality if and only if X and Y are independent random variables.

Exercise 11.6: Proof of strong sub-additivity

Prove that $H(X,Y,Z) \leq H(X,Y) + H(Y,Z)$, with equality if and only if $Z \to Y \to X$ forms a Markov chain.

Exercise 11.7

In Exercise 11.5 you implicitly showed that the mutual information $H(X:Y)$ could be expressed as the relative entropy of two probability distributions, $H(X:Y) = H(p(x,y)\|p(x)p(y))$. Find an expression for the conditional entropy $H(Y \mid X)$ as a relative entropy between two probability distributions. Use this expression to deduce that $H(Y \mid X) \geq 0$ and to find the equality conditions.

Exercise 11.8: Mutual information is not always subadditive

Let X and Y be independent identically distributed random variables taking the values 0 and 1 with probability $1/2$. Let $Z \equiv X \oplus Y$, where $\oplus$ denotes addition modulo 2. Show that the mutual information in this case is not subadditive,

$$H(X, Y : Z) \not\leq H(X : Z) + H(Y : Z).$$

Exercise 11.9: Mutual information is not always superadditive

Let X_1 be a random variable taking values 0 or 1 with respective probabilities of $1/2$ and $X_2 \equiv Y_1 \equiv Y_2 \equiv X_1$. Show that the mutual information in this case is not superadditive,

$$H(X_1 : Y_1) + H(X_2 : Y_2) \not\leq H(X_1, X_2 : Y_1, Y_2).$$

Exercise 11.10

Show that if $X \to Y \to Z$ is a Markov chain then so is $Z \to Y \to X$.

Exercise 11.11: Example calculations of entropy

Calculate $S(\rho)$ for

$$\rho = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}; \; \rho = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}; \; \rho = \frac{1}{3}\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}.$$

Exercise 11.12: Comparison of quantum and classical entropies

Suppose $\rho = p\,|0\rangle\langle 0| + (1-p)\,|+\rangle\langle +|$. Evaluate $S(\rho)$. Compare the value of $S(\rho)$ to $H(p, 1-p)$.

Exercise 11.13: Entropy if a tensor product

Use the joint entropy theorem to show that $S(\rho \otimes \sigma) = S(\rho) + S(\sigma)$. Prove this result directly from the definition of the entropy.

Exercise 11.14: Entanglement and negative conditional entropy

Suppose $|AB\rangle$ is a pure state of a composite system belonging to Alice and Bob. Show that $|AB\rangle$ is entangled if and only if $S(B \mid A) < 0$.

Exercise 11.15: Generalised measurements can decrease entropy

Suppose a qubit is in the state ρ is measured using the measurement operators $M_1 = |0\rangle\langle 0|$ and $M_2 = |0\rangle\langle 1|$. If the result of the measurement is unknown to us then the state of the system afterwards is $M_1 \rho M_1^\dagger + M_2 \rho M_2^\dagger$. Show that this procedure can *decrease* the entropy of the qubit.

Exercise 11.16: Equality conditions for $S(A,B) \geq S(B) - S(A)$

Let $\rho^{AB} = \sum_i \lambda_i |i\rangle\langle i|$ is a spectral decomposition for ρ^{AB}. Show that $S(A,B) = S(B) - S(A)$ if and only if the operators $\rho^A \equiv \mathrm{tr}_B(|i\rangle\langle i|)$ have a common eigenbasis and the $\rho^B \equiv \mathrm{tr}_A(|i\rangle\langle i|)$ have orthogonal support.

Exercise 11.17

Find an explicit non-trivial example of a mixed state ρ for AB such that $S(A,B) = S(B) - S(A)$.

Exercise 11.18

Prove that equality holds in the concavity inequality (11.79) if and on,y if all the ρ_is are the same.

Exercise 11.19

Show that there exists a set of unitary matrices U_j and a probability distribution p_j such that for any matrix A,

$$\sum_i p_i U_i A U_i^\dagger = \mathrm{tr}(A)\frac{I}{d},$$

where d is the dimension of the Hilbert space A lives in. Use this observation and the strict concacity of the entropy to give an alternate proof that the completely mixed state I/d on a space of d dimensions is the unique state of maximal entropy.

Exercise 11.20

Let P be a projector and $Q = I - P$ the complementary projector. Prove that there are unitary operators U_1 and U_2 and a probability p such that for all ρ, $P\rho P + Q\rho Q = pU_1\rho U_1^\dagger + (1-p)U_2\rho U_2^\dagger$. Use this observation to give an alternate proof of Theorem 11.9 based on concavity.

Exercise 11.21: Concavity of the Shannon entropy

Use the concavity of the von Neumann entropy to deduce that the Shannon entropy is concave in probability distributions.

Exercise 11.22: Alternate proof of concavity

Define $f(p) \equiv S(p\rho + (1-p)\sigma)$. Argue that to show concavity it is sufficient to prove that $f''(p) \leq 0$. Prove that $f''(p) \leq 0$, first for the case where ρ and σ are invertible, and then for the case where they are not.

Exercise 11.23: Joint concavity implies concavity in each input

Let $f(A,B)$ be a jointly concave function. Show that $f(A,B)$ is concave in A, with B held fixed. Find a function of two variables that is concave in each of its inputs, but is not jointly concave.

Exercise 11.24

We obtained strong subadditivity as a consequence of the inequality $S(A) + S(B) \leq S(A, C) + S(B, C)$. Show that this inequality can be obtained as a consequence of strong subadditivity.

Exercise 11.25

We obtained strong subadditivity as a consequence of the concavity of the conditional information, $S(A|B)$. Show that the concavity of the conditional entropy may be deduced from strong subadditivity. (*Hint:* You may need to introduce an auxiliary system into the problem.)

Exercise 11.26

Prove that $S(A : B) + S(A : C) \leq 2S(A)$. Note that the corresponding inequality for Shannon entropies holds since $H(A : B) \leq H(A)$. Find an example where $S(A : B) > S(A)$.

Problem 11.1: Generalized Klein's inequality

Suppose $f(\cdot)$ is a convex functions from real numbers to real numbers. Then f induces a natural function $f(\cdot)$ on Hermitian operators, as described in Section 2.1.8 on page 75. Prove that

$$\operatorname{tr}\big(f(A) - F(B)\big) \geq \operatorname{tr}\big((A - B)f'(B)\big).$$

Use this result to show that the relative entropy is non-negative.

Problem 11.2: Generalized relative entropy

The definition of the relative entropy may be extended to apply to any two positive operators r and s,

$$S(r\|s) \equiv \operatorname{tr}(r \log r) - \operatorname{tr}(r \log s).$$

The earlier argument proving joint convexity of the relative entropy goes directly through for this generalized definition:

(1) For $\alpha, \beta > 0$ show that

$$S(\alpha r\|\beta s) = \alpha S(r\|s) + \alpha \operatorname{tr}(r) \log\big(\alpha/\beta\big).$$

(2) Prove that the joint convexity of the relative entropy implies the subadditivity of the relative entropy,

$$S(r_1 + r_2\|s_1 + s_2) \leq S(r_1|s_1) + S(r_2\|s_2).$$

(3) Prove that subadditivity of the relative entropy implies joint convexity of the relative entropy.

(4) Let p_i and q_i be probability distributions over the same set of indices. Show that

$$S\left(\sum_i p_i r_i | \sum_i q_i s_i\right) \leq \sum_i p_i S(r_i|s_i) + \sum_i p_i \operatorname{tr}(r_i) \log\big(p_i/q_i\big).$$

In the case where the r_i are density operators so $\mathrm{tr}(r_i) = 1$, this reduces to the pretty formula

$$S\left(\sum_i p_i r_i \| \sum_i q_i s_i\right) \leq \sum_i p_i S(r_i \| s_i) + H(p_i \| q_i),$$

where $H(\cdot\|\cdot)$ is the Shannon relative entropy.

Problem 11.3: Analogue of the triangle inequality for conditional entropy

(1) Show that $H(X, Y|Z) \geq H(X|Z)$.

(2) Show that it is not always true that $S(A, B|C) \geq S(A|C)$.

(3) Prove the conditional version of the triangle inequality,

$$S(A, B|C) \geq S(A|C) - S(B|C).$$

Problem 11.4: Conditional forms of strong subadditivity

(1) Prove that $S(A, B, C|D) + S(B|D) \leq S(A, B|D) + S(B, C|D)$.

(2) Show by explicit example that it is not always true that $H(D|A, B, C) + H(D, B) \leq H(D|A, B) + H(D|B, C)$.

Problem 11.5: Strong subadditivity – Research

Find a simple proof of the strong subadditivity inequality for quantum entropies.

Solution

Concepts Involved:

Has been done in D. Petz. Quasi-entropies for finite quantum systems. Rep. Math. Phys., 23(1):57–65, 1986. Discussed in https://arxiv.org/pdf/quant-ph/0408130v2. □

12 Quantum information theory

Exercise 12.1

Suppose $|\psi\rangle$ and $|\varphi\rangle$ are two orthogonal quantum states of a single qubit. Design a quantum circuit with two input qubits (the 'data' and the 'target' qubits), with the data qubit in either the state $|\psi\rangle$ or $|\varphi\rangle$, and the target qubit prepared in the standard state $|0\rangle$, which produces as output $|\psi\rangle|\psi\rangle$ or $|\varphi\rangle|\varphi\rangle$, depending on whether $|\psi\rangle$ or $|\varphi\rangle$ was input to the data qubit.

Exercise 12.2

Define U_y to be the unitary operator acting on system M whose action on a basis is $U_y|y'\rangle \equiv |y'+y\rangle$, where the addition is done modulo $n+1$. Show that $\left\{\sqrt{E_y}\otimes U_y\right\}$ is a set of operation elements defining a trace-preserving quantum operation $\mathcal{E}$ whose action on states of the form $\sigma\otimes|0\rangle\langle 0|$ agreed with (12.8).

Exercise 12.3

Use the Holevo bound to argue that n qubits can not be used to transmit more than n bits of classical information.

Exercise 12.4

Suppose Alice sends Bob an equal mixture of the four pure states

$$
\begin{aligned}
|X_1\rangle &= |0\rangle \\
|X_2\rangle &= \sqrt{\frac{1}{3}}\left[|0\rangle + \sqrt{2}|1\rangle\right] \\
|X_3\rangle &= \sqrt{\frac{1}{3}}\left[|0\rangle + \sqrt{2}e^{2\pi i/3}|1\rangle\right] \\
|X_4\rangle &= \sqrt{\frac{1}{3}}\left[|0\rangle + \sqrt{2}e^{4\pi i/3}|1\rangle\right]
\end{aligned}
$$

Show that the maximum mutual information between Bob's measurement and Alice's transmission is less than one bit. A POVM which achieves ≈ 0.415 bits us known. Can you construct this or better yet, one which achieves the Holevo bound?

Exercise 12.5: Variable-length zero error data compression

Consider the following heuristic for a variable length data compression scheme. Let $x_1,\ldots,x_n$ be the output from n uses of an i.i.d. source with entropy rate $H(X)$. If $x_1,\ldots,x_n$ is typical, then send a $H(X)$ bit index indicating which typical sequence it is. If $x_1,\ldots,x_n$ is atypical, send an uncompressed $\log d^n$ bit index for the sequence (recall that d is the alphabet size). Turn this heuristic into a rigorous argument that the source can be compressed to an average of R bits per source symbol, for any $R > H(X)$, with zero probability of error.

Exercise 12.6

In the notation of Box 12.4, give an explicit expression for C_X in terms of X. Also, describe how to construct a quantum circuit to perform U_n for arbitrary n. How many elementary operations do you require, as a function of n?

Exercise 12.7: Data compression circuit

Outline the construction of a circuit to reliably compress a qubit source with $\rho = p|0\rangle\langle 0| + (1-p)|1\rangle\langle 1|$ into nR qubits for any $R > S(\rho) = H(\rho)$.

Exercise 12.8: Compression of an ensemble of quantum states

Suppose that instead of adopting the definition of a quantum source based on a single density matrix ρ and the entanglement fidelity, we instead adopted the following *ensemble* definition, that an (i.i.d.) quantum source is specified by an ensemble $\{p_j, |\psi_j\rangle\}$ of quantum states, and that consecutive used of the source are independent and produce a state $|\psi_j\rangle$ with probability p_j. A compression-decompression scheme $(\mathcal{C}^n, \mathcal{D}^n)$ is said to be reliable in this definition if the *ensemble average fidelity* approaches 1 as $n \to \infty$:

$$\bar{F} \equiv \sum_J p_{j_1} \dots p_{j_n} F(\rho_J, (\mathcal{D}^n \circ \mathcal{C}^n)(\rho_J))^2,$$

Where $J = (j_1, \dots, j_n)$ and $\rho_J \equiv |\psi_{j_1}\rangle\langle\psi_{j_1}| \otimes \dots \otimes |\psi_{j_n}\rangle\langle\psi_{j_n}|$. Define $\rho \equiv \sum_j p_j |\psi_j\rangle\langle\psi_j|$ and show that provided $R > S(\rho)$ there exists a reliable compression scheme of rate R with respect to this definition of fidelity.

Exercise 12.9

The *erasure channel* has two inputs, 0 and 1, and three outputs, $0, 1$ and e. With probability $1-p$. the input is left alone. With probability p the input is 'erased', and replaced by e.

(1) Show that the capacity of the erasure channel is $1-p$.

(2) Prove that the capacity of the erasure channel is greater than the capacity of the binary symmetric channel. Why is this result intuitively plausible?

Exercise 12.10

Suppose $\mathcal{N}_1$ and $\mathcal{N}_2$ are two discrete memoryless channels such that the input alphabet of $\mathcal{N}_2$ is equal to the output alphabet of $\mathcal{N}_1$. Show that

$$C(\mathcal{N}_1 \circ \mathcal{N}_2) \leq \min(C(\mathcal{N}_1), C(\mathcal{N}_2)).$$

Find an example where the inequality is strict.

Exercise 12.11

Show that the maximum in the expression (12.71) may be achieved using an ensemble of pure states. Show further that it suffices to consider only ensembles of at most d^2 pure states, where d is the dimension of the input to the channel.

Exercise 12.12

Adapt the proof of the HSW theorem to find a proof of Shannon's noisy channel coding theorem, simplifying the proof wherever possible.

Exercise 12.13

Show that the entropy exchange is concave in the quantum operation $\mathcal{E}$.

Exercise 12.14

Show that the condition $\rho^{RQ_1} = \rho^R \otimes \rho^{Q_1}$ is also *sufficient* to be able to correct errors on the subsystem Q_1.

Exercise 12.15

Apply all possible combinations of the subadditivity and strong subadditivity inequalities to deduce other inequalities for the two stage quantum process $\rho \mapsto \rho' = \mathcal{E}_1(\rho) \mapsto \rho'' = (\mathcal{E}_2 \circ \mathcal{E}_1)(\rho)$, expressing the results whenver possible in terms of entropy excahnges and the entropies $S(\rho), S(\rho'), S(\rho'')$. When it is not possible to express a quantity appearing in such an inequality in these terms, give a prescription for calculating the quantity only using a knowledge of ρ and operation elements $\{E_j\}$ for $\mathcal{E}_1$ and $\{F_k\}$ for $\mathcal{E}_2$.

Exercise 12.16

Show that in the case where $\mathcal{R}$ perfectly corrects $\mathcal{E}$ for the input ρ, the inequality

$$S(\rho) - S(\rho') + S(\rho', \mathcal{R}) \geq 0$$

must actually be satisfied with equality.

Exercise 12.17

Show that $x \prec y$ if and only if for all real t, $\sum_{j=1}^d \max(x_j - t, 0) \leq \sum_{j=1}^d \max(y_j - t, 0)$, and $\sum_{j=1}^d x_j = \sum_{j=1}^d y_j$.

Exercise 12.18

Use the previous exercise to show that the set of x such that $x \prec y$ is convex.

Exercise 12.19

Verify that $x' \prec y'$.

Exercise 12.20

Show that the assumption that ρ_ψ is invertible may be removed from the proof of the converse part of Theorem 12.15.

Exercise 12.21: Entanglement catalysis

Suppose Alice and Bob share a pair of four level systems in the state $|\psi\rangle = \sqrt{0.4}\,|00\rangle + \sqrt{0.4}\,|11\rangle + \sqrt{0.1}\,|22\rangle + \sqrt{0.1}\,|33\rangle$. Show that it is not possible for them to convert this state by LOCC to the state $|\varphi\rangle = \sqrt{0.5}\,|00\rangle + \sqrt{0.25}\,|11\rangle + \sqrt{0.25}\,|22\rangle$. Imagine, however, that a friendly bank is willing to offer them the loan of a *catalyst*, an entangled pair of qubits in the state $|c\rangle = \sqrt{0.6}\,|00\rangle + \sqrt{0.4}\,|11\rangle$. Show that it is possible for Alice and Bob to convert the state $|\psi\rangle|c\rangle$ to $|\varphi\rangle|c\rangle$ by local operations and classical communication, returning the catalyst $|c\rangle$ to the bank after the transformation is complete.

Exercise 12.22: Entanglement conversion without communication

Suppose Alice and Bob are trying to convert a pure state $|\psi\rangle$ into a pure state $|\varphi\rangle$ using local operations only - no classical communciation. Show that this is possible if and only if $\lambda_\psi \cong \lambda_\psi \otimes x$ where x is some real vector with non-negative entries summing to 1, and '$\cong$' means that the vectors on the left and right have identical non-zero entries.

Exercise 12.23

Prove that the procedure for entanglement distillation we have described is optimal.

Exercise 12.24

Recall that the Schmidt number of a bi-partite pure state is the number of non-zero Schmidt components. Prove that the Schmidt number of a pure quantum state cannot be increased by local operations and classical communication. Use this result to argue that the number of Bell states shared between Alice and Bob cannot be increased by local operations and classical communication.

Exercise 12.25

Consider a system with n users, any pair of which would like to be able to communicate privately. Using public key cryptography how many keys are required? Using private key cryptography how many keys are required?

Exercise 12.26

Let a'_k be Bob's measurement result of qubit $|\psi_{a_k b_k}\rangle$, assuming a noiseless channel with no eavesdropping. Show that when $b'_k \neq b_k$, a'_k is random and completely uncorrelated with a_k. But when $b'_k = b_k$, $a'_k = a_k$.

Exercise 12.27: Randomized sampling tests

The random test of n of $2n$ check bits allows Alice and Bob to place an upper bound on the number of errors in their untested bits, with high probability. Specifically, for any $\delta > 0$, the probability of obtainining less than δn erros on the check bits, and more than $(\delta + \epsilon)n$ errors on the remanining n bits is asymptotically less than $\exp\left[-O(\epsilon^2 n)\right]$, for large n. We prove this claim here.

(1) Without loss of generality, you may assume that there are μn errors in the $2n$ bits, where $0 \leq \mu \leq 2$. Now, if there are δn errors on the check bits, and $(\delta + \epsilon)n$ errors on the rest, then $\delta = (\mu - \epsilon)/2$. The two conditional statements in the claim thus imply the following:

$$\begin{aligned} &< \delta n \text{ errors on check bits} &&\implies < \delta n \text{ errors on check bits} \\ &> (\delta + \epsilon)n \text{ errors on rest} &&\implies > (\mu - \delta)n \text{ errors on rest,} \end{aligned}$$

and in fact, the top claim on the right implies the bottom one on the right. Using this, show that the probability p which we would like to bound satisfies

$$p < \binom{2n}{n}^{-1} \binom{\mu n}{\delta n} \binom{(2-\mu)n}{(1-\delta)n} \delta n.$$

(2) Show that for large n, you can bound

$$\frac{1}{an+1} 2^{anH(b/a)} \leq \binom{an}{bn} \leq 2^{anH(b/a)}$$

where $H(\cdot)$ is the binary entropy function, Equation (11.8). Apply this to the above bound for p.

(3) Apply the bound $H(x) < 1 - 2(x - 1/2)^2$ to obtain the final result, $p < \exp\left[-O(\epsilon^2 n)\right]$. You may replace μ by a constant which expresses the worst possible case.

(4) Compare this result with the Chernoff bound, Box 3.4. Can you come up with a different way to derive an upper bound on p?

Exercise 12.28

Show that when $b = 1$, then a and a' are perfectly correlated with each other.

Exercise 12.29

Give a protocol using six states, the eigenstates of X, Y and Z, and argue why it is also secure. Discuss the sensitivity of this protocol to noise and eavesdropping, in comparison with that of BB84 and B92.

Exercise 12.30

Simplify (12.199) to obtain the expression for $S(\rho)$ given in the statement of the Lemma.

Exercise 12.31

It may be unclear why $S(\rho)$ bounds Eve's mutual information with Alice and Bob's measurement results. Show that this follows from assuming the worst about Eve, giving her *all* the control over the channel.

Exercise 12.32

Note that the local measurements that Alice and Bob perform, such as $I \otimes X$ and $X \otimes I$, do *not* commute with the Bell basis. Show that despite this, the statistics which Alice and Bob compile from their measurements are the same as those which they would have obtained had they actually measured Π_{bf} and Π_{pf}.

Exercise 12.33

Let $\{M_1, M_2, \ldots, M_n\}$ be a set of measurement observables which produce respective results X_i when an input state ρ is measured. Argue that the random variables X_i obey classical probability arguments if $[M_i, M_j] = 0$, that is, they commute with each other.

Exercise 12.34: Entanglement distillation by error-correction

In Section 10.5.8, we saw that codewords of an $[n, m]$ qubit stabilizer code can be constructed by measuring its generators $g_1, \ldots, g_{n-m}$ on an *arbitrary* n qubit quantum state, then applying Pauli operations to change the result to be a simultaneous $+1$ eigenstate of the generators. Using that idea, show that if we start out with n EPR pairs in the state $|\beta_{00}\rangle^{\otimes n}$, and perform identical generator measurements on the two n qubit halves of the pairs, followed by Pauli operations to correct for *differences* in the measurement results between the pairs, then we obtain an encoded $|\beta_{00}\rangle^{\otimes m}$ state. Also show that if the stabilizer code corrects up to δn errors, then even if δn errors are suffered by an n qubit half, we still obtain $|\beta_{00}\rangle^{\otimes m}$.

Exercise 12.35

Show that the states $|\xi_{v_k,z,x}\rangle$ defined in (12.202) form an orthonormal basis for a 2^n-dimensional Hilbert space, that is,

$$\sum_{v_k,z,x} |\xi_{v_k,z,x}\rangle\langle\xi_{v_k,z,x}| = I.$$

Hint: for C_1 an $[n, k_1]$ code, C_2 an $[n, k_2]$ code, and $m = k_1 - k_2$, note that there are 2^m distinct values of v_k, 2^{n-k_1} distinct x, and 2^{k_1} distinct z.

Exercise 12.36

Verify Equation (12.203).

Exercise 12.37

This is an alternative way to understand why Alice's measurements in steps 9 and 10 collapse EPR pairs into random qubits encoded in a random quantum code. Suppose Alice has an EPR pair $(|00\rangle + |11\rangle)/\sqrt{2}$. Show that if she measures the first qubit in the X basis, then the second qubit collapses into an eigenstate of X determined by the measurement result. Similarly, show that if she measures in the Z basis, then the second qubit is left in a Z eigenstate labeled by the measurement result. Using this observation and the results of section 10.5.8, conclude that Alice's measurements of $H_1, H_2^{\perp}$, and $\bar{Z}$ on her EPR pair halves result in a random codeword of $CSS_{z,x}(C_1, C_2)$ determined by her measurement results.

Exercise 12.38

Show that if you had the ability to distinguish non-orthogonal states, then it would be possible to compromise the security of BB84, and indeed, all of the QKD protocols we have described.

Problem 12.1

In this problem, we will work through an alternate proof of the Holevo bound. Define the *Holevo chi quantity*,

$$\chi \equiv S(\rho) - \sum_x p_x S(p_x)$$

(1) Suppose the quantum system consists of two parts, A and B. Show that

$$\chi_A \leq \chi_{AB}.$$

(*Hint:* Introduce an extra system which is correlated with AB, and apply strong subadditivity.)

(2) Let $\mathcal{E}$ be a quantum operation. Use the previous result to show that

$$\chi' \equiv S(\mathcal{E}(\rho)) - \sum_x p_x S(\mathcal{E}(p_x)) \leq \chi \equiv S(\rho) - \sum_x p_x S(p_x)$$

That is, the Holevo chi quantity decreases under quantum operations. This is an important and useful fact in its own right.

(3) Let E_y be a set of POVM elements. Augment the quantum system under consideration with an 'apparatus' system, M, with an orthonormal basis $|y\rangle$. Define a quantum operation by

$$\mathcal{E}(\rho \otimes |0\rangle\langle 0|) \equiv \sum_y \sqrt{E_y}\rho\sqrt{E_y} \otimes |y\rangle\langle y|$$

where $|0\rangle$ is some standard pure state of M. Prove that after the action of $\mathcal{E}$, $\chi_M = H(X:Y)$. Use this and the previous two results to show that

$$H(X:Y) \leq S(\rho) - \sum_x p_x S(p_x)$$

which is the Holevo bound.

Problem 12.2

This result is an extension of the previous problem. Provide a proof of the no-cloning theorem by showing that a cloning process for non-orthogonal pure states would necessarily increase χ.

Problem 12.3

For a fixed quantum source and rate $R > S(\rho)$, design a quantum circuit implementing a rate R compression scheme.

Problem 12.4: Linearity forbids cloning

Supose we have a quantum machine with two slots, A and B. Slot A, the *data slot*, starts out in an unknown quantum state ρ. This is the state to be copied. Slot B, the *target slot*, starts out in some standard quantum state, σ. We will assume that any candidate copying procedure is *linear* in the initial state,

$$\rho \otimes \sigma \mapsto \mathcal{E}(\rho \otimes \sigma) = \rho \otimes \rho$$

where $\mathcal{E}$ is some linear function. Show that if $\rho_1 \neq \rho_2$ are density operations such that

$$\mathcal{E}(\rho_1 \otimes \sigma) = \rho_1 \otimes \rho_1$$
$$\mathcal{E}(\rho_2 \otimes \sigma) = \rho_2 \otimes \rho_2$$

then any mixture of ρ_1 and ρ_2 is not copied correctly by this procedure.

Solution

Concepts Involved:

Denote $\rho = c_1\rho_1 + c_2\rho_2$ an arbitrary mixture of ρ_1, ρ_2. By the linearity of $\mathcal{E}$, we have:

$$\begin{aligned}\mathcal{E}(\rho \otimes \sigma) &= \mathcal{E}((c_1\rho_1 + c_2\rho_2) \otimes \sigma) \\ &= c_1\mathcal{E}(\rho_1 \otimes \sigma) + c_2\mathcal{E}(\rho_2 \otimes \sigma) \\ &= c_1\rho_1 \otimes \rho_1 + c_2\rho_2 \otimes \rho_2\end{aligned}$$

Let us compare the above to $\rho \otimes \rho$:

$$\begin{aligned}\rho \otimes \rho &= (c_1\rho_1 + c_2\rho_2) \otimes (c_1\rho_1 + c_2\rho_2) \\ &= c_1^2\rho_1 \otimes \rho_1 + c_1c_2(\rho_1 \otimes \rho_2 + \rho_2 \otimes \rho_1) + c_2^2\rho_2 \otimes \rho_2\end{aligned}$$

Unless one of c_1, c_2 is zero (and the other is one) then the two expressions are not equal, and we conclude that ρ is not correctly cloned. □

Problem 12.5: Classical capacity of a quantum channel (Research)

Is the product state capacity (12.71) the true capacity of a noisy quantum channel for classical information, that is, the capacity when entangled inputs to the channel are allowed?

Problem 12.6: Methods for achieving capacity (Research)

Find an efficient construction for codes achieving rates near the product state capacity (12.71) of a noisy quantum channel for classical information.

Problem 12.7: Quantum channel capacity (Research)

Find a method to evaluate the capacity of a given quantum channel $\mathcal{E}$ for the transmission of quantum information.

A1 Notes on basic probability theory

Exercise A1.1

Prove Bayes' rule.

Solution

Concepts Involved: Probability, Conditional Probability.
Recall that conditional probabilities were defined as:

$$P(Y = y|X = x) = \frac{P(X = x, Y = y)}{P(X = x)}$$

and also recall that Bayes' rule is given by:

$$p(x|y) = p(y|x)\frac{p(x)}{p(y)}$$

By the definition of conditional probability:

$$p(y|x)\frac{p(x)}{p(y)} = \frac{p(X = x, Y = y)}{p(x)}\frac{p(x)}{p(y)} = \frac{P(X = x, Y = y)}{p(y)} = p(x|y)$$

□

Exercise A1.2

Prove the law of total probability.

Solution

Concepts Involved: Probability, Conditional Probability.
Recall that the law of total probability is given by:

$$p(y) = \sum_x p(y|x)p(x)$$

Using the identity $p(Y = y) = \sum_x p(X = x, Y = y)$ and Bayes' rule, we have

$$p(y) = \sum_x p(x, y) = \sum_x p(y|x)p(x)$$

□

Exercise A1.3

Prove that there exists a value of $x \geq \mathbf{E}(X)$ such that $p(x) > 0$.

Solution

Concepts Involved: Probability, Expectation.
Recall that the expectation of a random variable X is defined by:

$$\mathbf{E}(X) = \sum_x p(x)x$$

Let $\tilde{x} = \max\{x : x \text{ is a possible value of } X\}$. This maximum exists as we assume X can only take on a finite set of values. We therefore have that:

$$\mathbf{E}(X) = \sum_x p(x)x \leq \sum_x p(x)\tilde{x} = \tilde{x}\sum_x p(x) = \tilde{x}$$

Where in the last equality we use that the sum over all probabilities must be 1. □

Exercise A1.4

Prove that $\mathbf{E}(X)$ is linear in X

Solution

Concepts Involved: Probability, Expectation.

Let $a, b \in \mathbb{R}$ and X, Y be random variables. We then have that:

$$\begin{aligned}
\mathbf{E}(aX + bY) &= \sum_x \sum_y p(x,y)(ax + by) \\
&= \sum_x \sum_y p(x,y)ax + \sum_x \sum_y p(x,y)by \\
&= a\sum_x \left(\sum_y p(x,y)\right) x + b\sum_y \left(\sum_x p(x,y)\right) y \\
&= a\sum_x p(x)x + b\sum_y p(y)y \\
&= a\mathbf{E}(X) + b\mathbf{E}(y)
\end{aligned}$$

which shows that expectation is linear. □

Exercise A1.5

Prove that for independent random variables X and Y, $\mathbf{E}(XY) = \mathbf{E}(X)\mathbf{E}(Y)$.

Solution

Concepts Involved: Probability, Expectation, Independent Random Variables.

Recall two random variables X, Y are independent if

$$p(X = x, Y = y) = p(X = x)p(Y = y)$$

We have that:

$$\mathbf{E}(XY) = \sum_x \sum_y xyp(x, y) = \sum_x \sum_y xyp(x)p(y) = \left(\sum_x p(x)x\right)\left(\sum_y p(y)y\right) = \mathbf{E}(x)\mathbf{E}(y)$$

□

Exercise A1.6

$(*)$ Prove Chebyshev's inequality.

Solution

Concepts Involved: Probability, Expectation, Variance.
Recall the definition of the variance and standard deviaiton of a random variable X:

$$\mathrm{Var}(X) = \mathbf{E}[(X - \mathbf{E}(X))^2], \quad \Delta(X) = \sqrt{\mathrm{Var}(X)}$$

Also, recall that Chebyshev's inequality reads:

$$p(\big|X - \mathbf{E}(X)\big| \geq \lambda\Delta(X)) \leq \frac{1}{\lambda^2}$$

where $\lambda > 0$.

We first establish Markov's inequality for the expectation value $\mathbf{E}(X)$. Let $a > 0$, and then we have that:

$$\mathbf{E}(X) = \sum_x xp(x) = \sum_{x \geq a} xp(x) + \sum_{x < a} xp(x) \geq \sum_{x \geq a} ap(x) + 0 = ap(X \geq a)$$

Therefore, we obtain that:

$$p(X \geq a) \leq \frac{\mathbf{E}(X)}{a}$$

for any random variable X and $a > 0$. Next, substitute X with $(X - \mathbf{E}(X))^2$ and let $a = \lambda^2 \mathrm{Var}(X)$ for $\lambda > 0$. Markov's inequality then states that:

$$p((X - \mathbf{E}(X))^2 \geq \lambda^2 \mathrm{Var}(X)) \leq \frac{\mathbf{E}(X - \mathbf{E}(X))^2}{\lambda^2 \mathrm{Var}(X)}$$

Since $\mathbf{E}(X - \mathbf{E}(X))^2 = \mathrm{Var}(X)$, we have that:

$$p((X - \mathbf{E}(X))^2 \geq \lambda^2 \mathrm{Var}(X)) \leq \frac{1}{\lambda^2}$$

If $\lambda > 0$, then $p((X - \mathbf{E}(X))^2 \geq \lambda^2 \operatorname{Var}(X)) = p(\big|X - \mathbf{E}(X)\big| \geq \lambda\Delta(X))$ by taking square roots, so we obtain:

$$p(\big|X - \mathbf{E}(X)\big| \geq \lambda\Delta(X)) \leq \frac{1}{\lambda^2}$$

as desired. □

A2 Group theory

Exercise A2.1

Prove that for any element of g of a finite group, there always exists a positive integer r such that $g^r = e$. That is, every element of such a group has an order.

Solution

Concepts Involved: Group Axioms, Order.

Suppose G is a finite group, and $g \in G$. Then, there exists some $r_1, r_2 \in \mathbb{N}$ such that $r_1 \neq r_2$ and $g^{r_1} = g^{r_2}$. If this was not the case, then g^n would be unique for each $n \in \mathbb{N}$, contradicting the finiteness of G. WLOG take $r_1 < r_2$, and let $r = r_2 - r_1 \in \mathbb{N}$. Using associativity, we then have that:

$$g^{r_1} = g^{r_2} = g^{r_1 + r} = g^{r_1} g^r$$

from which we conclude that $g^r = e$. □

Exercise A2.2

Prove Lagrange's Theorem.

Solution

Concepts Involved: Group Axioms, Subgroups, Order, Equivalence Relations.

Let H be a subgroup of a group G and define the relation $\sim$ by $a \sim b$ iff $a = bh$ for some $h \in H$. $\sim$ is reflexive as $a = ae$ (so $a \sim a$) where $e \in H$ is the identity element. $\sim$ is symmetric as if $a \sim b$, then $a = bh$ for some $h \in H$ so $b = ah^{-1}$ (so $b \sim a$) where $h^{-1} \in H$ as H is closed under inverses. Finally $\sim$ is transitive as if $a \sim b$ and $b \sim c$, there exist $h_1, h_2 \in H$ such that $a = bh_1$ and $b = ch_2$ so $a = ch_2h_1$. As $h_2h_1 \in H$ (H is closed under multiplication) it follows that $a \sim c$. Having shown $\sim$ to have these three properties, we conclude it is an equivalence relation. Then, the equivalence classes of $\sim$ partition G, where the equivalence class of $g \in G$ is $[g] = \{gh | h \in H\}$.
Now, let $g \in G$ and define the map $\varphi_g : H \to [g]$ as $\varphi_g(h) = gh$. φ_g is injective as if $\varphi_g(h_1) = \varphi_g(h_2)$ then $gh_1 = gh_2$ and multiplying by g^{-1} on both sides $h_1 = h_2$. φ_g is surjective as if $k \in [g]$, then there exists some $h \in H$ such that $k = gh$ by the definition of $\sim$. Hence φ_g is bijective.
As per our prior observation, the equivalence classes of $\sim$ partition G, so $G = \bigcup_{i=1}^n [g_i]$ and $|G| = \left|\bigcup_{i=1}^n [g_i]\right| = \sum_{i=1}^n \left|[g_i]\right|$. Further, there is a bijection φ_{g_i} from each equivalence class to H, so $\left|[g_i]\right| = |H|$ for all i. Thus $|G| = \sum_{i=1}^n |H| = nH$ and hence $|H|$ divides $|G|$, as desired. □

Exercise A2.3

Show that the order of an element $g \in G$ divides $|G|$.

Solution

Concepts Involved: Group Axioms, Subgroups, Order, Lagrange's Theorem.

Let $g \in G$ with order r. Then, define $H = \{g^n | n \in \mathbb{N}\}$. We claim that H is a subgroup of G. First, $g^n \in G$ for any n as G is closed under multiplicaton, so $H \subset G$. Next, if $g^{n_1}, g^{n_2} \in H$ then $g^{n_1} \cdot g^{n_2} = g^{n_1+n_2} \in H$. Associativity is inherited from the associativity of multiplication in G. Since $g^r = e \in H$, H contains the identity. Finally, for $g^k \in H$ we have $g^{r-k} \in H$ such that $g^k g^{r-k} = g^{r-k} g^k = g^r = e$ so H is closed under inverses. Hence the claim is proven.
Next, we observe that $|H| = r$ as H contains the r elements $e, g, g^2, \ldots g^{r-1}$. Hence by Lagrange's Theorem r divides $|G|$. □

Exercise A2.4

Show that if $y \in G_x$ then $G_y = G_x$

Solution

Concepts Involved: Group Axioms, Conjugacy Classes

Suppose $y \in G_x$. Then there exists some $g \in G$ such that $g^{-1}xg = y$. Multiplying both sides on the left by g and on the right by g^{-1} we find that $x = gyg^{-1}$. We now show the two inclusions.
$\boxed{\subseteq}$ Suppose that $k \in G_x$. Then there exists some $g' \in G$ such that $k = g'^{-1}xg'$. Then using $x = gyg^{-1}$ we find $k = g'^{-1}gyg^{-1}g'$. Now, $g^{-1}g' \in G$ (by closure) and it has inverse $g'^{-1}g$, and hence $k = g'^{-1}gyg^{-1}g' \in G_y$. So, $G_x \subseteq G_y$.
$\boxed{\supseteq}$ Suppose that $l \in G_y$. Then there exists some $g'' \in G$ such that $l = g''^{-1}yg''$. Then with $g^{-1}xg = y$ we find $l = g''^{-1}g^{-1}xgg''$. Much like before, $gg'' \in G$ (by closure) with inverse $g''^{-1}g^{-1}$ so $l \in G_x$. So, $G_y \subseteq G_x$.
We conclude that $G_y = G_x$. □

Exercise A2.5

Show that if x is an element of an Abelian group G, then $G_x = \{x\}$.

Solution

Concepts Involved: Abelian Groups, Conjugacy Classes.

Evidently $x = e^{-1}xe \in G_x$ so $\{x\} \subseteq G_x$. Next, if $k \in G_x$ then $k = g^{-1}xg$ for some $g \in G$, but since G is abelian, $g^{-1}x = xg^{-1}$ so $k = xg^{-1}g = xe = x$ so $k \in \{x\}$ and hence $G_x \subseteq \{x\}$. We conclude that $G_x = \{x\}$. □

Exercise A2.6

Show that any group of prime order is cyclic.

Solution

Concepts Involved: Order, Cyclic Groups.

Suppose $|G| = p$ where p is prime. Since G is finite, every element of G has an order by Exercise A2.1. Since the order of any element $g \in G$ divides $|G| = p$ by Exercise A2.3, and since p is prime, the order of g is either 1 or p. Since $|G| > 1$, there exists at least one $g \in G$ with order p, and this g is a generator of G (with $g^1 = g, g^2, g^3, \ldots, g^p = e$ distinct and comprising all the elements of G. In fact this is true of any non-identity g). Hence G is cyclic. □

Exercise A2.7

Show that every subgroup of a cyclic group is cyclic.

Solution

Concepts Involved: Group Axioms, Subgroups, Cyclic Groups, Euclid's Division Algorithm.

First we prove a necessary Lemma, namely that any nonempty subset of the natural numbers contains a least element. We show this by proving the contrapositive. Suppose that $A \subseteq \mathbb{N}$ has no least element. Then $1 \notin A$ as then 1 would be the least element. Suppose then that $1, \ldots k-1 \notin A$; then $k \notin A$ as then k would be the least element. By strong induction, there exists no $k \in \mathbb{N}$ such that $k \in A$, i.e. A is empty. This concludes the proof of the lemma.
Let $G = \langle a \rangle$ be a cyclic group and H a subgroup of G. If $H = \{e\}$, it is trivially cyclic and we are done. If $H \neq \{e\}$, then there exists some $a^n \in H$ with $n \neq 0$. Since H is closed under inverses, $(a^n)^{-1} = a^{-n} \in H$ as well which ensures that H contains some positive power of a. Then consider the set $A = \{k \in \mathbb{N} | a^k \in H\}$. Any nonempty subset of the naturals has a minimum element; therefore let $d = \min A$. It is immediate that $\langle a^d \rangle$ is a subgroup of H as $a^d \in H$ and H is a group. To show the reverse containment, suppose that $g \in H$. Since H is a subgroup of the cyclic G, it follows that $g = a^p$ for some $p \in \mathbb{Z}$. We can then write $p = qd + r$ for $0 \leq r < d$ by Euclid's Division algorithm (see Appendix 4). We then have that $a^r = a^{p-qd} = a^p (a^d)^{-q} \in H$ by closure. Now, since d is the least positive integer for which $a^d \in H$ and $0 \leq r < d$, it must follow that $r = 0$. Therefore, $p = qd$ and hence $a^{qd} = (a^d)^q \in \langle a^d \rangle$. So, H is a subgroup of $\langle a^d \rangle$. We conclude that $H = \langle a^d \rangle$ and hence H is cyclic.
□

Exercise A2.8

Show that if $g \in G$ has finite order r, then $g^m = g^n$ if and only if $m = n \pmod{r}$.

Solution

Concepts Involved: Order, Modular Arithmetic, Euclid's Division Algorithm

Suppose $g \in G$ has finite order r.
$\boxed{\Longleftarrow}$ First suppose that $m = n \pmod{r}$. Then $m - n = kr$ for some $k \in \mathbb{N}$. Therefore $g^{m-n} = g^{kr}$. But $g^{kr} = (g^r)^k = e^k = e$, so $g^{m-n} = g^m g^{-n} = e$, and multiplying both sides by g^n we find $g^m = g^n$.
$\boxed{\Longrightarrow}$ Suppose $g^m = g^n$. Then multiplying both sides by g^{-n} we find $g^{m-n} = e$. By Euclid's Division algorithm there exist integers q, p such that $m - n = qr + p$ with $0 \leq p < r$. We then have that

$g^{m-n} = g^{qr+p} = g^{qr}g^p = e$. Furthermore, $g^{qr} = (g^r)^q = e^q = e$ so $g^p = e$. But since g has order r and $0 \leq p < r$, it follows that $p = 0$. Hence $m - n = qr$ and so $m \equiv n \pmod{r}$. □

Exercise A2.9

Cosets define an equivalence relation between elements. Show that $g_1, g_2 \in G$ are in the same coset of H in G if and only if there exists some $h \in H$ such that $g_2 = g_1 h$.

Solution

Concepts Involved: Equivalence Relations, Cosets

In Exercise A2.2 we showed that the relation $\sim$ on a group G defined by $g_1 \sim g_2$ iff $g_1 = g_2 h$ for some $h \in H$ was an equivalence relation. The equivalence classes of this equivalence relation were $\{gh | h \in H\}$, i.e. precisely the left cosets of H in G. So, g_1, g_2 are in the same coset of H in G if and only if $g_1 = g_2 h$ for some $h \in H$, which is exactly what we wished to show. □

Exercise A2.10

How many cosets of H are there in G?

Solution

Concepts Involved: Equivalence Relations, Cosets

We observe that the map $\varphi_g : H \to [g]$ defined in the solution of Exercise A2.2 is a map from H to a right coset of H in G defined by g. Since we showed that this map was bijective, this shows that $|H| = |Hg|$ for any $g \in G$. Furthermore, since the cosets define an equivalence relation between elements of G, the cosets of H in G partition G. So, we conclude that there are $|G|/|H|$ cosets of H in G, each of cardinality $|H|$. □

Exercise A2.11: Characters

Prove the properties of characters given above.

Solution

Concepts Involved: Matrix Groups, Character (Trace)
Recall that the character of a matrix group $G \subset M_n$ is a function on the group defined by $\chi(g) = \mathrm{tr}(g)$ where tr is the trace function. It has the properties that (1) $\chi(I) = n$, (2) $\left|\chi(g)\right| \leq n$, (3) $\left|\chi(g)\right| = n$ implies $g = e^{i\theta} I$, (4) χ is constant on any given conjugacy class of G, (5) $\chi(g^{-1}) = \chi^*(g)$ and (6) $\chi(g)$ is an algebraic number for all g.

The six properties are proven below.

(1) $\chi(I) = \mathrm{tr}(I) = \sum_{k=1}^{n} 1 = n$.

(2) Let $g \in G$. Since G is finite, by Exercise A2.1 it follows that g has order r such that $g^r = I$. So, g may be diagonalized with roots of unity $e^{2\pi i j/r}$, $j \in \{0, 1, \ldots, r-1\}$ on the diagonal. We then

find using the triangle inequality that:

$$|\chi(g)| = |\mathrm{tr}(g)| = \left|\sum_{k=1}^{n} e^{2\pi i j_k/r}\right| \leq \sum_{k=1}^{n} \left|e^{2\pi i j_k/r}\right| = \sum_{i} 1 = n$$

which proves the claim.

(3) The (complex) triangle inequality $|z_1 + z_2| \leq |z_1| + |z_2|$ is saturated when $z_1 = kz_2$ for some $k \geq 0$. This can only occur in the above equation when every λ_i in the sum is identical (as distinct roots of unity are not related by a non-negative constant). If the λ_is are identical, then g is diagonal with diagonal entries of unit modulus, so $g = e^{i\theta} I$ as claimed.

(4) Let $G_x = \left\{g^{-1}xg | g \in G\right\}$ be the conjugacy class of x in G. We then have for any $h \in G_x$ that $\chi(h) = \chi(g^{-1}xg) = \mathrm{tr}\left(g^{-1}xg\right) = \mathrm{tr}\left(xgg^{-1}\right) = \mathrm{tr}(xI) = \mathrm{tr}(x)$, using the cyclicity of the trace. We conclude that χ is constant on the conjugacy class.

(5) By the same argument as (2), $g \in G$ can be diagonalized with roots of unity $e^{2\pi ij/r}$ on the diagonal:

$$g = \begin{bmatrix} e^{2\pi i j_1/r} & & & \\ & e^{2\pi i j_2/r} & & \\ & & \ddots & \\ & & & e^{2\pi i j_n/r} \end{bmatrix}$$

It then follows that g^{-1} is:

$$g^{-1} = \begin{bmatrix} e^{-2\pi i j_1/r} & & & \\ & e^{-2\pi i j_2/r} & & \\ & & \ddots & \\ & & & e^{-2\pi i j_n/r} \end{bmatrix}$$

So we have that:

$$\chi(g^{-1}) = \mathrm{tr}\left(g^{-1}\right) = \sum_{j=1}^{n} e^{-2\pi i j_k/r} = \sum_{j=1}^{n} (e^{2\pi i j_k/r})^* = \left(\sum_{j=1}^{n} e^{2\pi i j_k/r}\right)^* = (\mathrm{tr}(g))^* = \chi^*(g).$$

which proves the claim.

(6) $\chi(g)$ is the sum of r-th roots of unity, which are algebraic; hence $\chi(g)$ is algebraic as the sum of algebraic numbers.

□

Exercise A2.12: Unitary matrix groups

$(*)$ A unitary matrix group is comprised solely of unitary matrices (those who which satisfy $U^\dagger U = I$). Show that every matrix group is equivalent to a unitary matrix group. If a representation of a group consists entirely of unitary matrices, we may refer to it as being a *unitary representation*.

Solution

Concepts Involved: Matrix Groups, Character (Trace), Equivalence, Unitary Operators.
Recall that two groups are equivalent if they are isomorphic (i.e. there is a bijection between the groups that respects the group multiplication) and the isomorphic element have the same character.

Let $G = \{A_1, \dots A_n\}$ be a finite matrix group. Then define

$$A = \sum_{i=1}^{n} A_i^\dagger A_i.$$

By Ex. **??** each term of the above sum is positive, and by Ex. **??** each term is Hermitian. The sum of Hermitian operators is Hermitian, so A is Hermitian. By Ex. **??**, A is diagonalizable. Let U be the unitary matrix that diagonalizes A. We then have that D is a diagonal matrix, with:

$$D = UAU^\dagger.$$

Let $D^{1/2}$ be the matrix obtained by taking the square root of the diagonal entries of D. Then define $T = D^{1/2}U$. We then claim that $G_U = \{V_1, \dots V_n\}$ is a unitary matrix group equivalent to G, where:

$$V_i = TA_iT^{-1}.$$

We have three points to verify; (i) That the V_is are unitary, (ii) That $\varphi : G \to G_u$ defined by $\varphi(A_i) = TA_iT^{-1} = V_i$ is an isomorphism, and (iii) that the characters of A_i and V_i are equivalent.

(i) For any V_i, we have:

$$\begin{aligned}
V_i^\dagger V_i &= (TA_iT^{-1})^\dagger (TA_iT^{-1}) \\
&= (D^{1/2}UA_iU^\dagger D^{-1/2})^\dagger (D^{1/2}UA_iU^\dagger D^{-1/2}) \\
&= (D^{-1/2}UA_i^\dagger U^\dagger D^{1/2})(D^{1/2}UA_iU^\dagger D^{-1/2}) \\
&= (D^{-1/2}UA_i^\dagger U^\dagger) D (UA_iU^\dagger D^{-1/2}) \\
&= (D^{-1/2}UA_i^\dagger U^\dagger)(UAU^\dagger)(UA_iU^\dagger D^{-1/2}) \\
&= (D^{-1/2}UA_i^\dagger) A (A_i^\dagger U^\dagger D^{-1/2}) \\
&= (D^{-1/2}UA_i^\dagger) \left(\sum_{j=1}^{n} A_j^\dagger A_j \right) (A_iU^\dagger D^{-1/2}) \\
&= D^{-1/2}U \left(\sum_{j=1}^{n} (A_jA_i)^\dagger (A_jA_i) \right) U^\dagger D^{-1/2} \\
&= D^{-1/2}U \left(\sum_{k=1}^{n} A_k^\dagger A_k \right) U^\dagger D^{-1/2} \\
&= D^{-1/2}UAU^\dagger D^{-1/2} \\
&= D^{-1/2}DD^{-1/2} \\
&= I
\end{aligned}$$

Where in the sixth equality we use the unitarity of U, and in the ninth equality we use that $A_j A_i = A_k$ iterates over all the group elements as A_j iterates over all the group elements. To see that this is the case, it suffices to show that the map $\psi_i : M_n \to M_n$ defined by $\psi_i(A_j) = A_j A_i$ is a bijection. To see that it is injective, suppose that $\psi_i(A_{j_1}) = \psi_i(A_{j_2})$. Then it follows that $A_{j_1} A_i = A_{j_2} A_i$, and multiplying on the left by A_i^{-1} (which exists) we find that $A_{j_1} = A_{j_2}$. To see that it is surjective, suppose that $A_{j'} \in M_n$. Then, there exists $A_{j'} A_i^{-1} \in M_n$ such that $\psi_i(A_{j'} A_i^{-1}) = A_{j'} A_i^{-1} A_i = A_{j'}$. We conclude that ψ_i is bijective.

(ii) Firstly, φ is a homomorphism as for any A_i, A_j we have:

$$\varphi(A_i)\varphi(A_j) = V_i V_j = T A_i T^{-1} T A_j T^{-1} = T A_i A_j T^{-1} = \varphi(A_i A_j).$$

Next, φ is surjective by construction. Finally, it is injective; suppose that $V_i = V_j$. Then we have that:

$$T A_i T^{-1} = T A_j T^{-1}$$

And multiplying both sides on the left by T^{-1} on the left and T on the right we find that $A_i = A_j$. Hence we conclude that φ is a bijective homomorphism and hence an isomorphism.

(iii) This is immediate from the cyclicity of the trace:

$$\chi(V_i) = \mathrm{tr}\Big(T A_i T^{-1}\Big) = \mathrm{tr}\Big(T^{-1} T A_i\Big) = \mathrm{tr}(A_i) = \chi(A_i).$$

The claim is therefore proven. □

Exercise A2.13

Show that every irreducible Abelian matrix group is one dimensional.

Exercise A2.14

Show that if ρ is an irreducible representation of G, then $|G|/d_\rho$ is an integer.

Exercise A2.15

Using the Fundamental Theorem, prove that characters are orthogonal, that is:

$$\sum_{i=1}^{r} r_i (\chi_i^p)^* \chi_i^q = |G| \delta_{pq} \text{ and } \sum_{p=1}^{r} (\chi_i^p)^* \chi_j^q = \frac{|G|}{r_i} \delta_{ij}$$

where p, q, and δ_{pq} have the same meaning as in the theorem and χ_i^p is the value the character of the pth irreducible representation takes on the ith conjugacy class of G and r_i is the size of the ith conjugacy class of G and r_i is the size of the ith conjugacy class.

Exercise A2.16

S_3 is the group of permutations of three elements. Suppose we order these as mapping 123 to: 123;231;312;213;132, and 321, respectively. Show that there exist two one-dimensional irreducible representations of S_3, one of which is trivial, and the other of which is 1,1,1,-1,-1,-1, corresponding in order to the six permutations given earlier. Also show that there exists a two dimensional irreducible representation, with the matrices

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \frac{1}{2}\begin{bmatrix} -1 & -\sqrt{3} \\ \sqrt{3} & -1 \end{bmatrix}, \quad \frac{1}{2}\begin{bmatrix} -1 & \sqrt{3} \\ -\sqrt{3} & 1 \end{bmatrix},$$
$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \frac{1}{2}\begin{bmatrix} 1 & \sqrt{3} \\ \sqrt{3} & -1 \end{bmatrix}, \quad \frac{1}{2}\begin{bmatrix} 1 & -\sqrt{3} \\ -\sqrt{3} & 1 \end{bmatrix}$$

Verify that the representations are orthogonal.

Exercise A2.17

Prove that the regular representation is faithful.

Exercise A2.18

Show that the character of the regular representation is zero except on the representation of the identity element, for which $\chi(I) = |G|$.

Exercise A2.19

Use Theorem A2.5 to show that the regular representation contains d_{ρ^p} instances of each irreducible representation ρ^p. Thus, if R denotes the regular representation, and $\hat{G}$ denotes the set of all inequivalent irreducible representations, then:

$$\chi_i^R = \sum_{\rho \in G} d_\rho \chi_i^\rho$$

Exercise A2.20

The character of the regular representation is zero except for the conjugacy class i containing e, the identity element in G. Show, therefore, that

$$\sum_{\rho \in G} d_\rho \chi^\rho(g) = N\delta_{ge}.$$

Exercise A2.21

Show that $\sum_{\rho \in \hat{G}} d_\rho^2 = |G|$.

Exercise A2.22

Substitute (A2.10) into (A2.9) and prove that $\hat{f}(\rho)$ is obtained.

Exercise A2.23

Let us represent an Abelian group G by $g \in [0, N-1]$, with addition as the group operation, and define $\rho_h(g) = \exp\left[-2\pi igh/N\right]$ at the h representation of g. This representation is one-dimensional, so $d_\rho = 1$. Show that the Fourier transform relations for G are

$$\hat{f}(h) = \frac{1}{\sqrt{N}} \sum_{g=0}^{N-1} f(g) e^{-2\pi igh/N} \text{ and } f(h) = \frac{1}{\sqrt{N}} \sum_{g=0}^{N-1} \hat{f}(g) e^{2\pi igh/N}$$

Exercise A2.24

Using the results of Exercise A2.16, construct the Fourier transform over S_3 and express it as a 6x6 unitary matrix.

A3 The Solovay-Kitaev theorem

Exercise A3.1

In Chapter 4 we made use of the distance measure $E(U,V) = \max_{|\psi\rangle} \big\|(U-V)|\psi\rangle\big\|$, where the maximum is over all pure states $|\psi\rangle$. Show that when U and V are single qubit rotations, $U = R_{\hat{m}}(\theta), V = R_{\hat{n}}(\varphi)$, $D(U,V) = 2E(U,V)$, and thus it does not matter whether we use the trace distance of the measure $E(\cdot,\cdot)$ for the Solovay-Kitaev theorem.

Exercise A3.2

Suppose A and B are Hermitian matrices such that $\mathrm{tr}\,|A|, \mathrm{tr}\{B\} \leq \epsilon$. Prove that for sufficiently small ϵ,

$$D([e^{-iA}, e^{-iB}]_{gp}, e^{-[A,B]}) \leq d\epsilon^3$$

for some constant d, estavlishing Equation (A3.6). (*Comment:* for practical purposes it may be interesting to obtain good bounds on d.)

Exercise A3.3

Let $\mathbf{x}$ and $\mathbf{y}$ be any two real vectors. Show that:

$$D(u(\mathbf{x}), u(\mathbf{y})) = 2\sqrt{2}\sqrt{1 - \cos\big(x/2\big)\cos\big(y/2\big) - \sin\big(x/2\big)\sin\big(y/2\big)\mathbf{x}\cdot\mathbf{y}},$$

where $x \equiv \|\mathbf{x}\|, y \equiv \|\mathbf{y}\|$, and $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ are the unit vectors in the $\mathbf{x}$ and $\mathbf{y}$ directions, respectively.

Exercise A3.4

Show that in the case of $\mathbf{y} = \mathbf{0}$ the formula for $D(u(\mathbf{x}), u(\mathbf{y}))$ reduces to

$$D(u(\mathbf{x}), I) = 4\sin\left|\frac{x}{4}\right|$$

Exercise A3.5

Show that when $x, y \leq \epsilon$,

$$D(u(\mathbf{x}), u(\mathbf{y})) = \|\mathbf{x} - \mathbf{y}\| + O(\epsilon^3)$$

Exercise A3.6

Fixing the set of $\mathcal{G}$ of elementary gates, describe an algorithm which, given a description of a single qubit unitary gate U and a desired accuracy $\epsilon > 0$, efficiently computes a sequence of gates from $\mathcal{G}$ such that ϵ-approximates U.

Problem A3.1

The following problem outlines a more elaborate construction that achieves $O(\log^2(1/\epsilon)\log^c(\log(1/\epsilon)))$ bound on the number of gates required to approximate to within ϵ of a desired target, for any $c > 2$.

(1) Suppose $\mathcal{N}$ is a δ-net in S_ϵ, for $0 < \delta < \epsilon < \epsilon_0$, ϵ_0 sufficiently small. Show that $[\mathcal{N},\mathcal{N}]_{gp}$ is a $d\delta\epsilon$-net in S_{ϵ^2}, for some constant d.

(2) Suppose G_l is a δ-net in S_ϵ, for $0 < \delta < \epsilon \leq \epsilon_0$. Show that $\mathcal{G}_{4^k l}$ is a $d^k \delta \epsilon^{2^k - 1}$-net is $S_{\epsilon^{2^k}}$.

(3) Suppose we define k by

$$k \equiv \left\lceil \log\left(\frac{\log(1/\epsilon)}{\log(1/\epsilon_0)} \right) \right\rceil,$$

and suppose we can find l such that $\mathcal{G}_l$ is a δ_0-net for S_{ϵ_0}, where

$$d^k \delta_0 = \epsilon_0.$$

Show that $\mathcal{G}_{4^k l}$ is an ϵ-net for $S_{\epsilon_0^{2^k}}$.

Problem A3.2: (Research)

If it exists, find an approximation procedure asymptotically faster than the result found in the previous problem. Ideally, a procedure would (a) saturate the $\Omega(\log(1/\epsilon))$ lower bound on the number of gates required to perform the approximation, and (b) provide an efficient algorithm for constructing such an approximation sequences of gates.

Problem A3.3: (Research)

Fix a finite set of single qubit gates $\mathcal{G}$ which can be performed fault-tolerantly and which generate a set dense in the single qubit gates; say the $\pi/8$ gate and the Hadamard gate. Develop an elegant, efficient, and reasonable tight method which, given an arbitrary single qubit gate U and some $\epsilon > 0$, produces a sequence of gates from the fault-tolerant set giving a ϵ-approximation to U, up to a global phase.

A4 Number theory

Exercise A4.1: Transitivity

Show that if $a|b$ and $b|c$ then $a|c$.

Solution

Concepts Involved:

We have $b = ak_1$ and $c = bk_2$ for $k_1, k_2 \in \mathbb{Z}$, so $c = bk_2 = (ak_1)k_2 = ak_1k_2$ for $k_1k_2 \in \mathbb{Z}$ and so $a|c$. □

Exercise A4.2

Show that if $d|a$ and $d|b$ then d also divides linear combinations of a and b, $ax + by$, where x and y are integers.

Solution

Concepts Involved:

We have $a = dk_1$ and $b = dk_2$ for $k_1, k_2 \in \mathbb{Z}$, so for $x, y \in \mathbb{Z}$ we have $ax + by = a(dk_1) + b(dk_2) = (ak_1 + bk_2)d$ with $ak_1 + bk_2 \in \mathbb{Z}$ so $d|ax + by$. □

Exercise A4.3

Suppose a and b are positive integers. Show that if $a|b$ then $a \leq b$. Conclude that if $a|b$ and $b|a$ then $a = b$.

Solution

Concepts Involved:

We have $b = ak$ for $k \in \mathbb{Z}$ and since $a, b > 0$ it must follow that $k > 0$ and hence $k \geq 1$. Thus $a \leq ak = b$. If $b|a$ also then $b \leq a$ and so combining the two relations $a = b$. □

Exercise A4.4

Find the prime factorizations of 697 and 36300.

Solution

Concepts Involved:

By brute-force division (starting from the smallest primes until we find a clean factor) we find:

$$697 = 17 \cdot 41$$

$$36300 = 2^2 \cdot 3 \cdot 5^2 \cdot 11^2$$

□

Exercise A4.5

For p a prime prove that all integers in the range 1 to $p-1$ have multiplicative inverses modulo p. Which integers in the range 1 to p^2-1 do not have multiplicative inverses modulo p^2?

Exercise A4.6

Find the multiplicative inverses of 17 modulo 24.

Exercise A4.7

Find the multiplicative inverse of $n+1$ modulo n^2, where n is any integer greater than 1.

Exercise A4.8: Uniqueness of the Inverse

Suppose b and b' multiplicative inverses of a, modulo n. Prove that $b = b' \pmod{n}$

Exercise A4.9

Explain how to find $\gcd(a, b)$ if the prime factorizations of a and b are known. Find the prime factorizations of 6825 and 1430, and use them to compute $\gcd(6825, 1430)$.

Exercise A4.10

What is $\varphi(187)$?

Exercise A4.11

Prove that

$$n = \sum_{d|n} \varphi(d)$$

where the sum is over all positive divisors d of n, including 1 and n. (*Hint:* Prove th result for $n = p^\alpha$ first, then use the multiplicative property (A4.22) of φ to complete the proof.)

Exercise A4.12

Verify that $\mathbb{Z}_n^*$ forms a group of size $\varphi(n)$ under the operation of multiplication modulo n.

Exercise A4.13

Let a be an arbitrary element of $\mathbb{Z}_n^*$. Show that $S \equiv \{1, a, a^2, \ldots\}$ forms a subgroup of $\mathbb{Z}_n^*$, and that the size of S is the least value of r such that $a^r = \pmod{n}$.

Exercise A4.14

Suppose g is generator for $\mathbb{Z}_n^*$. Show that g must have order $\varphi(n)$.

Exercise A4.15

Lagrange's theorem (Theorem A2.1 on page 610) is an elementary result of group theory stating that the size of a subgroup must divide the order of the group. Use Lagrange's theorem to provide an alternate proof of Theorem A4.9, that is, show that $a^{\varphi(n)} = 1 \pmod{n}$ for any $a \in \mathbb{Z}_n^*$.

Exercise A4.16

Use Theorem A4.9 to show that the order of x modulo N must divide $\varphi(N)$.

Exercise A4.17: Reduction of order-finding to factoring

We have seen that an efficient order-finding algorithm allows us to factor efficiently. Show that an efficient factoring algorithm would allow us to efficiently find the order modulo N of any x co-prime to N.

Exercise A4.18

Find the continued fraction expansion for $x = 19/17$ and $x = 77/65$.

Exercise A4.19

Show that $q_n p_{n-1} - p_n q_{n-1} = (-1)^n$ for $n \geq 1$. Use this fact to conclude that $\gcd(p_n, q_n) = 1$. (*Hint:* Induct on n.)

Problem A4.1: Prime number estimate

Let $\pi(n)$ be the number of prime numbers which are less than n. A difficult-to-prove result known as the *prime number theorem* asserts that $\lim_{n\to\infty} \pi(n)\log(n)/n = 1$ and this $\pi(n) \approx n/\log(n)$. This problem gives poor man's version of prime number theorem which gives a pretty good lower bound on the distribution of prime numbers.

(1) Prove that $n \leq \log \binom{2n}{n}$

(2) Show that

$$\log \binom{2n}{n} \leq \sum_{p \leq 2n} \left\lfloor \frac{\log(2n)}{\log p} \right\rfloor \log p$$

where the sum is over all primes p less than or equal to $2n$.

(3) Use the previous two results to show that

$$\pi(2n) \geq \frac{n}{\log(2n)}$$

A5 Public key cryptography and the RSA cryptosystem

Exercise A5.1

Written examples of the application of RSA tend to be rather opaque. It's better to work through an example yourself. Encode the word 'QUANTUM' (or at least the first letters!), one letter at a time, using $p = 3$ and $q = 11$. Choose appropriate values for e and d, and use a representation of English text involving 5 bits per letter.

Exercise A5.2

Show that d is also an inverse of e modulo r, and thus $d = d' \pmod{r}$

Problem A5.1

Write a computer program for performing encryption and decryption using the RSA algorithm. Find a pair 20 bit prime numbers and use them to encrypt a 40 bit message.

A6 Proof of Lieb's theorem

Exercise A6.1: $\leq$ is preserved under conjugation

If $A \leq B$, show that $XAX^\dagger \leq XBX^\dagger$ for all matrices X.

Solution

Concepts Involved: Positive Operators

If $A \leq B$, then $\langle\psi| (B-A) |\psi\rangle \geq$ for all ψ. It then follows that:

$$\langle\psi| (XBX^\dagger - XAX^\dagger) |\psi\rangle = \langle\psi| X(B-A)X^\dagger |\psi\rangle = (\langle\psi| X)(B-A)(X^\dagger |\psi\rangle) \geq 0$$

for all matrices X. Hence $XAX^\dagger \leq XBX^\dagger$. □

Exercise A6.2

Prove that $A \geq 0$ if and only if A is a positive operator.

Solution

Concepts Involved: Positive Operators

If A is a positive operator, then $0 \leq \langle\psi| A |\psi\rangle = \langle\psi| (A-0) |\psi\rangle$ for all $|\psi\rangle$ so $A \geq 0$. Suppose instead $A \geq 0$. Then $(A-0) = A$ is positive. □

Exercise A6.3: $\leq$ is a partial order

Show that the relation $\leq$ is a partial order on operators - that is, it is transitive ($A \leq B$ and $B \leq C$ implies $A \leq C$), asymmetric ($A \leq B$ and $B \leq A$ implies $A = B$), and reflexive ($A \leq A$).

Solution

Concepts Involved: Positive Operators

- (Transitivity) If $A \leq B$ and $B \leq C$ then $\langle\psi| (B-A) |\psi\rangle \geq 0$ and $\langle\psi| (C-B) |\psi\rangle \geq 0$, and hence:

  $$\langle\psi| A |\psi\rangle \leq \langle\psi| B |\psi\rangle \leq \langle\psi| C |\psi\rangle$$

  and so $\langle\psi| (C-A) |\psi\rangle \geq 0$ and $A \leq C$.

- (Asymmetry) If $A \leq B$ and $B \leq A$ then $\langle\psi| (B-A) |\psi\rangle \geq 0$ and $\langle\psi| (A-B) |\psi\rangle \geq 0$ for all $|\psi\rangle$. But this implies that $\langle\psi| B |\psi\rangle \geq \langle\psi| A |\psi\rangle$ and $\langle\psi| B |\psi\rangle \leq \langle\psi| A |\psi\rangle$ and hence $\langle\psi| A |\psi\rangle = \langle\psi| B |\psi\rangle$ for all $|\psi\rangle$, and hence $A = B$.

- (Reflexivity) $A - A = 0$ is positive (as $\langle\psi| 0 |\psi\rangle = 0 \geq 0$ for all $|\psi\rangle$ so $A \leq A$.

□

Exercise A6.4

Suppose A has eigenvalues λ_i. Define λ to be the maximum of the set $|\lambda_i|$. Prove that:

1. $\|A\| \geq \lambda$
2. When A is Hermitian, $\|A\| = \lambda$.
3. When

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix},$$

$\|A\| = 3/2 > 1 = \lambda$.

Solution

Concepts Involved: Eigenvalues, Operator Norms

1. Let $|v\rangle$ be the unit eigenvector with maximal magnitude eigenvalue λ_i, i.e. such that $|\lambda_i| = \lambda$. Then:

$$\|A\| = \max_{\langle u|u\rangle=1} \left|\langle u| A |u\rangle\right| \geq \left|\langle v| A |v\rangle\right| = |\lambda_i| = \lambda$$

2. By Ex. **??** (Spectral decomposition), there exists an orthonormal basis $\left\{|i\rangle\right\}_i$ such that A is diagonal. Therein:

$$A = \sum_i \lambda_i |i\rangle\langle i|$$

with λ_i real. Let us write any normalized $|u\rangle$ in this basis such that $|u\rangle = \sum_i c_i |i\rangle$ with $\sum_i |c_i|^2 = 1$. Then:

$$\begin{aligned}
\|A\| &= \max_{\langle u|u\rangle=1} \left|\langle u| A |u\rangle\right| \\
&= \max_{\{c_i\}} \left| \left(\sum_i c_i^* \langle i|\right) \left(\sum_j \lambda_j |j\rangle\langle j|\right) \left(\sum_k c_k |k\rangle\right) \right| \\
&= \max_{\{c_i\}} \left| \sum_i |c_i|^2 \lambda_i \right| \\
&\leq \max_{\{c_i\}} \left| \sum_i |c_i|^2 \right| \lambda \\
&= \lambda
\end{aligned}$$

where in the third equality we use the orthonormality of the basis and in the fourth equality we use that λ is the magnitude of the largest eigenvector. Thus $\|A\| \leq \lambda$ for Hermitian A, and combining

this with the general previous result that $\|A\| \leq \lambda$ we conclude $\|A\| = \lambda$.

3. Note that A is not Hermitian, and hence we do not expect the equality of the previous part to hold. First, solving for the eigenvalues we have:

$$0 = \det(A - I\lambda_i) = \det\left(\begin{bmatrix} 1-\lambda_i & 0 \\ 1 & 1-\lambda_i \end{bmatrix}\right) = (1-\lambda_i)^2 - 0 = (1-\lambda_i)^2$$

Hence A has eigenvalues $\lambda_1 = \lambda_2 = 1$ and so $\lambda = \max\left\{|\lambda_i|\right\} = 1$.
To solve for the operator norm, we observe that in the $\left\{|+\rangle, |-\rangle\right\}$ basis A can be represented as:

$$A \cong \begin{matrix} \\ |+\rangle \\ |-\rangle \end{matrix}\begin{matrix} \begin{matrix} \langle +| & \langle -| \end{matrix} \\ \begin{bmatrix} 3/2 & -1/2 \\ 1/2 & 1/2 \end{bmatrix} \end{matrix}$$

wherein it becomes clear that $|+\rangle$ would maximize $\langle u| A |u\rangle$, and so $\|A\| = 3/2$.

□

Exercise A6.5: AB and BA have the same eigenvalues

Prove that AB and BA have the same eigenvalues. (Hint: For invertible A, show that $\det(\lambda I - AB) = \det(\lambda I - BA)$, and thus the eigenvalues of AB and BA are the same. By continuity this holds even when A is not invertible).

Solution

Concepts Involved:

We recall that that the determinant is multiplicative, i.e. $\det(AB) = \det(A)\det(B)$, and as a consequence that $\det\left(A^{-1}\right) = \frac{1}{\det(A)}$ for invertible A. Then:

$$\begin{aligned}
\det(\lambda I - AB) &= 1 \cdot \det(\lambda I - AB) \cdot 1 \\
&= \det(I)\det(\lambda I - AB)\det(I) \\
&= \det\left(AA^{-1}\right)\det(\lambda I - AB)\det\left(AA^{-1}\right) \\
&= \det(A)\det\left(A^{-1}\right)\det(\lambda I - AB)\det(A)\det\left(A^{-1}\right) \\
&= \det(A)\det\left(\lambda A^{-1}A - A^{-1}ABA\right)\det\left(A^{-1}\right) \\
&= \det(A)\det(\lambda I - BA)\det\left(A^{-1}\right) \\
&= \det(A)\det(\lambda I - BA)\frac{1}{\det(A)} \\
&= \det(\lambda I - BA)
\end{aligned}$$

Thus $\det(\lambda I - AB) = \det(\lambda I - BA)$ for invertible A. Since the eigenvalues are the roots of the charac-

teristic polynomial given by the determinant, this tells us that AB, BA have the same eigenvalues for all invertible A, and by continuity all A, B. □

Exercise A6.6

Suppose A and B are such that AB is Hermitian. Using the previous two observations, show that $\|AB\| \leq \|BA\|$.

Solution

Concepts Involved:

By Ex. A6.5 AB and BA have the same eigenvalues, so $\lambda = \max\left\{\left|\lambda_{i,AB}\right|\right\} = \max\left\{\left|\lambda_{i,BA}\right|\right\}$. By part 1 of A6.4 we then have that $\|BA\| \geq \lambda$ and by part 2 we have that $\|AB\| = \lambda$ (as AB is Hermitian). Combining these relations $\|AB\| \leq \|BA\|$. □

Exercise A6.7

Suppose A is positive. Show that $\|A\| \leq 1$ if and only if $A \leq I$.

Solution

Concepts Involved:

Since $A \geq 0$, by the spectral theorem

$$A = \sum_i \lambda_i \, |v_i\rangle\langle v_i|, \qquad \lambda_i \geq 0,$$

and for positive (Hermitian) A,

$$\|A\| = \sup_{\|x\|=1} \langle x, Ax\rangle = \lambda_{\max}(A).$$

($\Rightarrow$) If $\|A\| \leq 1$, then $\lambda_{\max}(A) \leq 1$, hence $\lambda_i \leq 1$ for all i, and

$$\langle x, (I - A)x\rangle = \sum_i (1 - \lambda_i) \, |\langle v_i|x\rangle|^2 \geq 0,$$

so $I - A \geq 0$, i.e. $A \leq I$.
($\Leftarrow$) If $A \leq I$ (equivalently $I - A \geq 0$), then for any unit vector x,

$$\langle x, Ax\rangle \leq \langle x, Ix\rangle = 1.$$

Thus, $\|A\| = \sup_{\|x\|=1}\langle x, Ax\rangle \leq 1$.
Overall we have, $\|A\| \leq 1 \iff A \leq I$. □

Exercise A6.8

Let A be a positive matrix. Define a superoperator (linear operator on matrices) by the same equation $\mathcal{A}(X) \equiv AX$. Show that $\mathcal{A}$ is positive with respect to the Hilbert-Schmidt inner product. That is, for all X, $\mathrm{tr}\big(X^\dagger \mathcal{A}(X)\big) \geq 0$. Similarly, show that the superoperator defined by $\mathcal{A}(X) \equiv XA$ is positive with respect to the Hilbert-Schmidt inner product on matrices.

Solution

Concepts Involved:

For any X, it must be that $X^\dagger X$ and $XX^\dagger$ are positive (Ex. **??**). Further, any positive operator A satisfies $\mathrm{Tr}(A) \geq 0$ as its eigenvalues are all positive and so $\mathrm{Tr}(A) = \sum_i \lambda_i \geq 0$. We then observe that for any X:

$$\mathrm{Tr}\Big(X^\dagger \mathcal{A}(X)\Big) = \mathrm{Tr}\Big(X^\dagger AX\Big) = \mathrm{Tr}\Big(XX^\dagger A\Big) \geq 0$$

$$\mathrm{Tr}\Big(X^\dagger \mathcal{A}(X)\Big) = \mathrm{Tr}\Big(X^\dagger XA\Big) \geq 0$$

where we use that $XX^\dagger A$ and $X^\dagger XA$ are positive, being products of two positive operators. □