Computational Complexity Assignment 4

Tyler Tracy

April 17, 2023

Problem 1

Question:
Give examples of qubit systems in normalized form that satisfy the following:
(a) A 1-qubit system that will contain a 0 with a 30% chance and a 1 with a 70% chance
(b) 2-qubit system that has no chance of being measured as —11⟩, but an equal chance of being measured in any other state,
(c) An n-qubit system where all possible measurements are equally likely.

Part a

$$\sqrt{\tfrac{3}{10}}\,|0\rangle + \sqrt{\tfrac{7}{10}}\,|1\rangle$$

Part b

$$\sqrt{\tfrac{1}{3}}\,|00\rangle + \sqrt{\tfrac{1}{3}}\,|01\rangle + \sqrt{\tfrac{1}{3}}\,|10\rangle$$

Part c

$\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$

Problem 2

Question:
Write a matrix for the following quantum operations:
(a) A 2×2 matrix that rotates a qubit by an angle of $\pi/3$,
(b) A 4×4 matrix that performs the XOR operation on two qubits and stores the result in the first qubit
(c) A 8×8 matrix that takes in two inputs and a scratch-pad bit (assumed to be initialized to 0) and performs the OR operation on the inputs, storing the result in the scratch-pad bit

Part a

I will use the rotation matrix we learned during lecture and plug in $\pi/6$ for θ.

$$\begin{pmatrix} \cos(\frac{\pi}{3}) & -\sin(\frac{\pi}{3}) \\ \sin(\frac{\pi}{3}) & \cos(\frac{\pi}{3}) \end{pmatrix} \tag{1}$$

Part b

I will leave the value of the second qubit unchanged so that this operation can be reversed.

Here is the table that describes the operation:

Input	Output
00	00
01	11
10	10
01	01

And here is the matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{2}$$

Part c

I will leave the values of the first and second qubits unchanged so that this operation can be reversed. This only works because the scratch-pad bit is assumed to be zero.

Input	Output
00X	000
01X	011
10X	101
11X	111

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \tag{3}$$

Problem 3

Question:

Let M be a 2×2 unitary matrix. Recall that a matrix is unitary exactly when $MM^T = I$, that is:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} a & c \\ b & d \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{4}$$

Prove that if you knew the value of a, you could determine the entire matrix.

We can write out a list of formulas of how a relates to the variables by writing out the equations you get if you multiply the matrices.

$a^2 + b^2 = 1$

$ac + bd = 0$

$c^2 + d^2 = 1$

If we solve for the variables in terms of a we get:

$b = \pm\sqrt{1 - a^2}$

$c = \pm\sqrt{1 - a^2}$

$d = \pm a$

So, if we know a, we can determine the entire matrix.

Problem 4

Question:

There are 16 possible classical logic gates that have 2 input bits and 1 output bit. Suppose that you wanted to implement some of these gates as quantum gates so that the result of the computation overwrote one of the input bits from your quantum register rather than using a scratch-pad bit. How many of these gates could you implement in this way and why wouldn't they require scratch-pad bits?

The only gates that you can implement without adding an extra scratch pad bit are gates that have equal numbers of 0 and 1 as output. So they would all have two states that yield a 0 and two states that yield a 1. This is to ensure that the operation is reversible. The first qubit would store the output of the operation and the second qubit represent which input yielded that output. This only works if there are 2 0s or 1s because if there was more you would need more than 1 bit to identify the input that gave that output.

This means there are 6 possible classical logic gates that can be implemented as quantum gates without a scratch pad bit. They are:

Input	Output
00	10
01	11
10	00
01	01

Input	Output
00	10
01	00
10	11
01	01

Input	Output
00	10
01	01
10	00
01	11

Input	Output
00	00
01	11
10	10
01	01

Input	Output
00	00
01	10
10	01
01	11

Input	Output
00	00
01	01
10	10
01	11