Problem Set 1

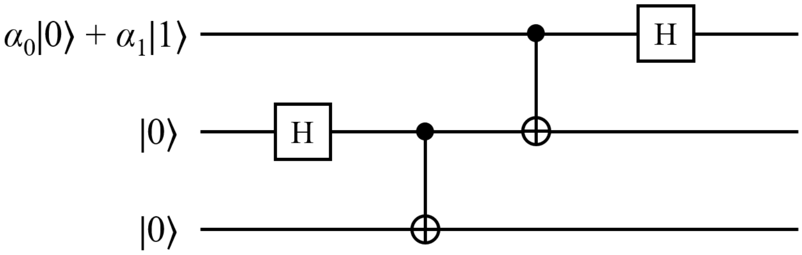
Due: Tuesday September 15, 11:59pm, email the PDF to pgarriso@andrew.cmu.edu

Homework policy: Please try to do the homework by yourself. If you get stuck, working in a group of two is okay; maybe three, max. Naturally, explicitly acknowledge sources you worked with. LaTeX typesetting with PDF output is highly preferred; very neatly handwritten material that has been scanned to PDF is acceptable. Questions about the homework or other course material can be asked on Piazza.

Solve any 5 out of 7

1. [**SWAP gate.**] A SWAP gate takes two inputs x_1 and x_2 and outputs x_2 and x_1; i.e., it swaps the values of two registers. Show how to build a SWAP gate using only CNOT gates. (Hint: you'll need 3 of them.)

2. [**Quantum circuit practice.**] Consider the following quantum circuit:



(Assume α_0 and α_1 are real numbers satisfying $\alpha_0^2 + \alpha_1^2 = 1$.)

An equivalent description of the circuit (calling the registers x_1, x_2, x_3) is:

1. Initialize x_1 to $\alpha_0 |0\rangle + \alpha_1 |1\rangle$
2. Initialize x_2 to $|0\rangle$
3. Initialize x_3 to $|0\rangle$
4. Hadamard(x_2)
5. CNOT(x_2, x_3)
6. CNOT(x_1, x_2)
7. Hadamard(x_1)

(a) Determine with proof the state of the three qubits at the end of the circuit's operation.

(b) If we then measure the three qubits, give the outcomes and their probabilities that arise.

3. **[Quantum gates preserve sum-of-squares and are reversible.]** Suppose we have n qubits in a quantum state

$$\sum_{x\in\{0,1\}^n} \alpha_x \left|x\right\rangle, \tag{1}$$

where as always, the sum of the squares of the amplitudes α_x is 1. Actually, in truth, quantum amplitudes are allowed to be complex numbers, so the true condition is that

$$\sum_{x\in\{0,1\}^n} |\alpha_x|^2 = 1.$$

(a) Verify that if a CCNOT (Toffoli) gate is applied to any 3 qubits, the resulting state still has the property that the sum of the squares of the (magnitudes) of the amplitudes is 1.

(b) Make the same verification supposing that a Hadamard H gate is applied to any 1 qubit.

(c) Suppose we begin with the n-qubit state (1) and apply a succession of gates $U_1, U_2, \ldots, U_t$ to it, where each U_i is either CCNOT applied to some 3 qubits or Hadamard applied to some 1 qubit. Show that if we subsequently apply the gates in reverse order, $U_t, U_{t-1}, \ldots, U_1$, the resulting state is again (1).

4. **[Uncomputing garbage.]** As described in class, any classical AND/OR/NOT circuit computing a function $f : \{0,1\}^n \to \{0,1\}^m$ can be efficiently converted to a reversible circuit using only CCNOT gates, assuming ancilla inputs and "garbage" outputs are allowed. In this problem we will also assume NOT and CNOT gates are allowed.[1] I.e., the circuit C will take some $n+a$ input bits and produce some $m+b$ output bits, where $n+a = m+b$; it has the property that

$$C\big(x_1, \ldots, x_n, \overbrace{\left|0\right\rangle, \ldots, \left|0\right\rangle}^{a \text{ ancillas}}\big) = \big(f(x)_1, \ldots, f(x)_m, \overbrace{g(x)_1, \ldots, g(x)_b}^{\text{garbage outputs}}\big).$$

For several reasons, it is undesirable for a the circuit to have the garbage $g(x)$; the main reason is that it makes it difficult to use the circuit as a subroutine. (Also, in some sense, computation with garbage is not really reversible; the garbage bits "leak" some entropy about x into the environment.) However in this problem you will show that it is possible to eliminate garbage.

(a) Give a straightforward transformation that converts C to a new reversible circuit C' with the following behavior: C' has $2n+a$ inputs, and when they are initialized to $(x, \left|0^n\right\rangle, \left|0^a\right\rangle)$, the output is $(x, f(x), g(x))$. (Hint: use CNOT gates to "copy" x.)

(b) Give a straightforward transformation that converts C' to a new reversible circuit C'' with the following behavior: C' has $2n+a+m$ inputs, and when they are initialized to $(x, \left|0^n\right\rangle, \left|0^a\right\rangle, y)$ (where y is any m-bit string), the output is $(x, \left|0^n\right\rangle, \left|0^a\right\rangle, y \oplus f(x))$, where $\oplus$ denotes bitwise XOR as usual. (Hint: use also ideas from Problem 3c.)

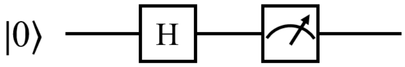
Remark: In the study of quantum (and reversible) computation we say that C'' *implements* the function f. Note that we can get the value of $f(x)$ in the final m registers by initializing y to $\left|0^m\right\rangle$, and that there is no garbage. We almost always insist on "implementing" functions, rather than just computing them with garbage. As the ancillas

[1] By allowing NOT gates, we may invoke the traditional assumption that ancilla bits are initialized to $\left|0\right\rangle$ rather than $\left|1\right\rangle$.

are perfectly restored to their initial state, we will sometimes be sloppy and not mention them, simply describing C'' as producing $(x, y \oplus f(x))$ on input (x, y).

(c) Discuss how the results of the previous problem extend from classical reversible computation to quantum computation. In particular, how/why do we not get unwanted "amplitude interference"?

5. **[Principle of deferred measurement.]** Given our definitions of randomized circuits (which allow AND, OR, NOT, and COIN$_{\frac{1}{2}}$) gates, and quantum circuits (which allow Hadamard, CNOT, CCNOT gates, plus measurement at the very end), it's not immediately obvious that quantum circuits are at least as powerful as randomized ones. We saw in class that CCNOT (together with ancillas) can simulate the AND, OR, and NOT gates, but it's not completely obvious how quantum circuits can simulate COIN$_{\frac{1}{2}}$ gates. A natural idea is the following: pass a $|0\rangle$ qubit through a Hadamard gate and then measure it —



On one hand, this *does* produce a (qu)bit which is $|0\rangle$ with probability $\frac{1}{2}$ and $|1\rangle$ with probability $\frac{1}{2}$, as desired. However, if we want to use this random bit within our circuit, we need to augment the quantum circuit model by allowing "intermediate measurements" (i.e., measuring some qubits prior to the end of the computation). While this is ultimately perfectly okay both theoretically and physically, it makes the model somewhat more complicated (more complicated than we want for introductory lectures!). On the other hand, there exists a not-very-hard-to-prove result called the *Principle of Deferred Measurement* that states that any computation done by a quantum circuit using intermediate measurements can be equivalently and nearly as efficiently done by a quantum circuit that only has a single measurement at the end. In this problem you won't quite prove this principle in full, but you'll get the essential idea, and in the meantime show that quantum circuits are indeed at least as powerful as randomized circuits.

Precisely, suppose C is a randomized circuit with n input bits, a ancilla bits, r COIN$_{\frac{1}{2}}$ gates, s CCNOT gates, and m output bits (possibly including garbage). Describe a straightforward transformation to a quantum circuit C' with n input bits, $a + 2r$ ancilla bits, $s + 2r$ CCNOT/CNOT/Hadamard gates, and $m + r$ output bits, such that when the output bits are measured at the end of $C'(x)$, the probability distribution on the first m of them is exactly the same as the probability distribution on the output bits of $C(x)$.

6. **[Simulating a biased coin.]** For $0 \leq p \leq 1$, let COIN$_p$ denote a gate that has no input and one output, the output being a random bit which is 1 with probability p and 0 with probability $1 - p$. The standard way to augment the basic circuit model[2] with randomness is to allow the use of COIN$_{\frac{1}{2}}$ gates. A more liberal model would be to allow COIN$_p$ gates for any rational value of p.

(a) In one sense, general COIN$_p$ gates are more powerful than COIN$_{\frac{1}{2}}$ gates. Show that if we only allow COIN$_{\frac{1}{2}}$ gates (as well as AND, OR, NOT, etc.), it is impossible to construct a circuit that *exactly* simulates a COIN$_{\frac{1}{3}}$ gate.

[2] Which has AND, OR, and NOT gates, ancillas, and allows arbitrary "fan-out" (equivalently, "DUPE" gates).

(b) However, in another sense, COIN_p gates are *not* fundamentally more powerful than $\mathrm{COIN}_{\frac{1}{2}}$. Show that for any $\epsilon > 0$, there is a circuit of $\mathrm{COIN}_{\frac{1}{2}}$ gates (and AND, OR, NOT etc.) of size $O(\log(1/\epsilon))$ that *almost* exactly simulates a $\mathrm{COIN}_{\frac{1}{3}}$ gate. Precisely, your circuit should have two output bits, called r and FAIL. The output bit FAIL should be 1 with probability at most ϵ. And the output bit r should have the property that $\mathbf{Pr}[r = 1 \mid \mathrm{FAIL} \neq 1] = \frac{1}{3}$ exactly.

(Part (b) is doable for any rational value of p, not just $\frac{1}{3}$; but we expect that once you solve it for $\frac{1}{3}$, you'll get the idea of how to do it for any p.)

7. **[Correctness amplification for randomized computation.]** Let $f : \{0,1\}^n \to \{0,1\}$ be a function we wish to compute, and let C be a circuit with n input bits $x_1, \dots, x_n$ and also one output bit (possibly with additional garbage bits). We write $C(x)$ for the output bit's value on input $x = (x_1, \dots, x_n)$. We assume either that C is a randomized circuit, or that a quantum circuit with a measurement at the end. In either case, for each input x we get a *random* output bit $C(x)$. A traditional definition in randomized computation is the following:

Definition 1. We say that C computes f if it holds that for *every* input $x \in \{0,1\}^n$,

$$\mathbf{Pr}[C(x) = f(x)] \geq \frac{2}{3}.$$

In a way, the point of this problem is to show that the arbitrary-looking constant $\frac{2}{3}$ doesn't matter too much (any constant strictly between $\frac{1}{2}$ and 1 is about the same). More precisely, suppose C computes f according to the above definition. Assume we now make $12n$ copies of C, call them $C_1, \dots, C_{12n}$, and use them as follows: On input $x \in \{0,1\}^n$, we feed x into all $12n$ copies and finally output the bit

$$z = \mathrm{Majority}\Big(C_1(x), C_2(x), \dots, C_{12n}(x)\Big).$$

(Observation: the circuit implementing this is not much bigger than that for C; its size is $O(n \cdot \mathrm{size}(C))$.) Prove the following superior correctness bound:

$$\mathbf{Pr}[z = f(x)] \geq 1 - 2^{-n}.$$

(Hint: if you know Chernoff bounds you may use them; just say the form you are citing. Otherwise, you can solve this problem using combinatorics; use the simplest possible upper bound for $\binom{12n}{i}$. By the way, it won't matter how ties are broken if the Majority splits $6n$ each.)