Quantum Computing (5314QUCO6Y), Final exam

Ronald de Wolf

May 29, 2015
9:00–12:00
SP F1.02

The exam is "open book", meaning you can bring any kind of paper you want but no electronic devices. Please answer in English. Use a black or blue pen, not a pencil. Write clear and explicitly. The total number of points adds up to 9; your exam grade will be your number of points +1.

1. **(1 point)** Show that unitaries cannot "delete" information: there is no one-qubit unitary U that maps $|\phi\rangle \mapsto |0\rangle$ for every one-qubit state $|\phi\rangle$.

2. **(2 points)**

   (a) Give a circuit that maps $|0^n, b\rangle \mapsto |0^n, 1-b\rangle$ for $b \in \{0,1\}$, and that maps $|i,b\rangle \mapsto |i,b\rangle$ whenever $i \in \{0,1\}^n \backslash \{0^n\}$. You are allowed to use every type of elementary gate mentioned in the lecture notes (incl. Toffoli gates), as well as auxiliary qubits that are initially $|0\rangle$ and that should be put back to $|0\rangle$ at the end of the computation.

   You can draw a Toffoli gate similar to a CNOT gate: a bold dot on each of the two control wires, and a '$\oplus$' on the target wire.

   (b) Suppose we can make queries of the type $|i,b\rangle \mapsto |i, b \oplus x_i\rangle$ to input $x \in \{0,1\}^N$, with $N = 2^n$. Let x' be the input x with its first bit flipped (e.g., if $x = 0110$ then $x' = 1110$). Give a circuit that implements a query to x'. Your circuit may use one query to x.

   (c) Give a circuit that implements a query to an input x'' that is obtained from x (analogously to (b)) by setting its first bit to 0. Your circuit may use one query to x.

3. **(2.5 points)** An informal pseudo-code description of algorithms is enough in this question, you don't have to write out circuits.

   Consider an undirected graph $G = (V, E)$, with vertex set $V = \{1, \ldots, n\}$ and edge-set E. We say G is *connected* if, for every pair of vertices $i, j \in V$, there is a path between i and j in the graph. The *adjacency matrix* of G is the $n \times n$ Boolean matrix M where $M_{ij} = 1$ iff $(i,j) \in E$ (note that M is a symmetric matrix because G is undirected). Suppose we are given input graph G in the form of a unitary that allows us to query whether an edge (i,j) is present in G or not:

   $$O_M : |i,j,b\rangle \mapsto |i,j,b \oplus M_{ij}\rangle.$$

(a) Assume G is connected. Suppose we have a set A of edges which we already know to be in the graph (so $A \subseteq E$; you can think of A as given classically, you don't have to query it). Let $G_A = (V, A)$ be the subgraph induced by only these edges, and suppose G_A is not connected, so it consists of $c > 1$ connected components. Call an edge $(i, j) \in E$ "good" if it connects two of these components. Give a quantum algorithm that finds a good edge with an *expected* number of $O(n/\sqrt{c-1})$ queries to M.

(b) Give a quantum algorithm that uses at most $O(n^{3/2})$ queries to M and that decides (with success probability at least $2/3$) whether G is connected or not.

(c) Show that classical algorithms for deciding (with success probability at least $2/3$) whether G is connected, need to make $\Omega(n^2)$ queries to M.

4. **(2 points)** The inner product problem in communication complexity is the function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ defined by $f(x, y) = \sum_{i=1}^{n} x_i y_i \bmod 2$. Suppose there exists a quantum protocol P for Alice and Bob that uses q qubits of communication (possibly using multiple messages between Alice and Bob) and computes the inner product function with success probability 1 (on every possible inputs x, y). The protocol does not assume any shared entangled state at the start.

   (a) Give a quantum protocol that uses $2q$ qubits of communication and maps $|x\rangle_A|y\rangle_B \mapsto (-1)^{x \cdot y}|x\rangle_A|y\rangle_B$ (possibly with some auxiliary qubits for each of Alice and Bob; these should start and end in state $|0\rangle$).

   (b) Give a quantum protocol where Alice transmits x to Bob using $2q$ qubits of communication. *Hint: run the (a)-protocol on an initial state where Bob has a superposition over many $|y\rangle$.*

   (c) Derive a lower bound on q from (b) and Holevo's theorem.

5. **(1.5 points)** Shor's 9-qubit code allows to *correct* a bit flip and/or a phase flip on one of its 9 qubits. Below we give a 4-qubit code which allows to *detect* a bitflip and/or a phaseflip. By this we mean that after the detection procedure we either have the original uncorrupted state back, or we know that an error occurred (though we do not know which one). The logical 0 and 1 are encoded as:

$$|\overline{0}\rangle = \tfrac{1}{2}(|00\rangle + |11\rangle) \otimes (|00\rangle + |11\rangle)$$
$$|\overline{1}\rangle = \tfrac{1}{2}(|00\rangle - |11\rangle) \otimes (|00\rangle - |11\rangle)$$

   (a) Give a procedure (either as a circuit or as sufficiently-detailed pseudo-code) that detects a bitflip error on one of the 4 qubits of $\alpha|\overline{0}\rangle + \beta|\overline{1}\rangle$.

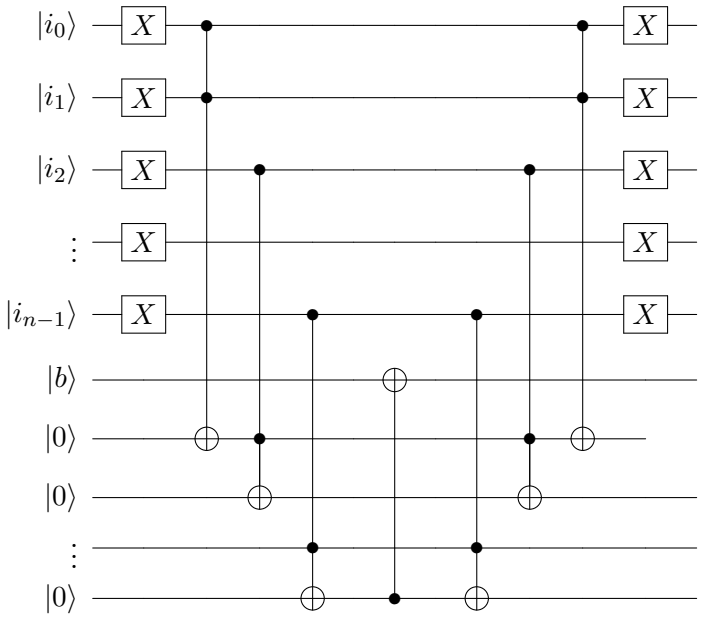
   (b) Give a procedure (either as a circuit or as sufficiently-detailed pseudo-code) that detects a phaseflip error on one of the 4 qubits of $\alpha|\overline{0}\rangle + \beta|\overline{1}\rangle$.

   (c) Does that mean that we can now detect *any* unitary 1-qubit error on one of the 4 qubits? Explain your answer.

Solutions

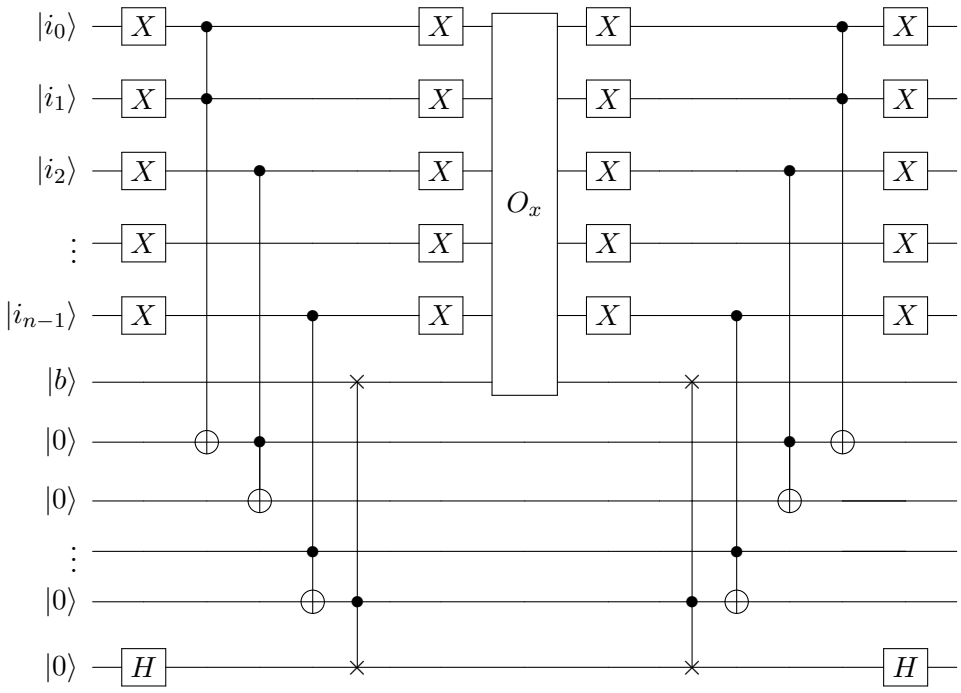
1. If $U|0\rangle = U|1\rangle = |0\rangle$ then U does not preserve the inner product between states $|0\rangle$ and $|1\rangle$, so then U cannot be unitary.

2. (a) High-level idea: first negate all address bits by X-gates, and then compute their logical AND using a sequence of $n-1$ Toffoli gates into $n-1$ auxiliary qubits that are initially $|0\rangle$. The last auxiliary qubit will then be 1 iff the initial i was 0^n. Now use a CNOT with the last auxiliary qubit as control and the $|b\rangle$-qubit as target, negating b iff $i = 0^n$. Then reverse the circuit (except for the CNOT) to set the auxiliary qubits back to 0. See the picture below.



   (b) First run the circuit of (a), and then apply O_x to the first $n+1$ qubits (or vice versa).

   (c) The idea is similar but more subtle than (b): controlled by the last auxiliary qubit (which is 1 iff $i = 0^n$) we swap $|b\rangle$ with a $|+\rangle$-qubit (prepared by applying H to another auxiliary $|0\rangle$), because O_x acts like identity on $|i\rangle|+\rangle$. In the circuit, the two 3-qubit gates with the two crosses are controlled-swaps. If you don't want to treat controlled-swap as an elementary gate, then by Exercise 2.4 you could construct them from Toffolis. See the following picture.



3. (a) Since G_A has c connected components but is a subgraph of the connected graph G, there are at least $c-1$ good edges (each good edge reduces the number of connected components by 1). Consider this as a search problem of size $N = \binom{n}{2}$ (because there are $\binom{n}{2}$ edges that could be present in G) with $t \geq c-1$ solutions. Since we know A classically, we can build a circuit that makes 2 queries to M and that maps $|i,j\rangle \mapsto s_{ij}|i,j\rangle$ with $s_{ij} = -1$ if (i,j) is a good edge, and $s_{ij} = 1$ if not. Now we can just run the expected-time version of Grover using the latter circuit for each query, and find a good edge using an expected number of $O(\sqrt{N/t}) = O(n/\sqrt{c-1})$ queries to M. If there is no good edge, this version of Grover won't terminate.

   (b) Consider the following algorithm:

   1. Set $A = \emptyset$
   2. If G_A is connected then output "connected" and halt
      else use the (a)-algorithm to find a good edge e and update $A := A \cup \{e\}$
   3. Goto 2

   First suppose G is connected. For the initial $A = \emptyset$, the graph G_A will consist of n isolated vertices, so it has n connected components. Each new good edge that we find reduces the number of connected components by 1, until we end with G_A connected, so then we know G is connected as well (note that the property $A \subseteq E$ remains valid throughout the computation). Using linearity of expectation, the algorithm will output "connected" using an expected number of

$$Q = \sum_{c=2}^{n} O(n/\sqrt{c-1}) = O\left(n \sum_{c=2}^{n} \frac{1}{\sqrt{c-1}}\right) = O\left(n \int_0^{n-1} \frac{1}{\sqrt{x}} dx\right) = O(n^{3/2})$$

queries to M. If G is *not* connected then this algorithm will not terminate. Now run this algorithm and output "not connected" if the algorithm hasn't terminated by itself after $3Q$ queries. So if G is *not* connected we will always give the correct output. If G *is* connected, the algorithm terminates with the correct answer within $3Q$ queries with probability at least $2/3$ (as otherwise the expected number of queries would be $> Q$).

(c) Let n be even and consider an n-vertex graph that consists of two disjoint cliques of $n/2$ vertices each (a "clique" is a graph where all possible edges are present). This graph is disconnected, but if we insert one or more of the $n^2/4$ possible edges between the two cliques, then it becomes connected. Hence if you can decide whether G is connected using T queries, you can also solve the OR problem on $n^2/4$ input bits with T queries: fix the edges inside the two cliques and run your T-query connectivity-algorithm using the $n^2/4$ input bits of the OR for the remaining edges. The graph will be connected iff the OR on those $n^2/4$ bits is 1. Since we know that computing OR on $n^2/4$ bits takes a classical randomized algorithm $\Omega(n^2)$ queries, we need at least that many queries to decide whether a given n-vertex graph is connected.

4. (a) Run the protocol on starting state $|x\rangle_A|y\rangle_B$, possibly with some auxiliary $|0\rangle$-qubits if the protocol needs those for workspace. This costs q qubits of communication and gives a state $|\phi_{xy}\rangle_{AB}|f(x,y)\rangle_B$, where the last qubit contains the answer bit, and the rest of the state is shared between Alice and Bob (and may be horribly entangled and complicated). Now Bob applies a Z gate to his last qubit, which multiplies the state with $(-1)^{f(x,y)} = (-1)^{x\cdot y}$. Finally, they reverse the protocol (everything except the last Z gate) at the cost of another q qubits of communication. The final state is $(-1)^{x\cdot y}|x\rangle_A|y\rangle_B$, possibly with some $|0\rangle$-qubits.

(b) Let Alice start with $x \in \{0,1\}^n$, and let Bob start with the uniform superposition over all y (which he can create for instance by applying Hadamard gates to $|0^n\rangle$). Applying the (a)-protocol gives the following change in the state:

$$|x\rangle_A \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} |y\rangle_B \mapsto |x\rangle_A \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} (-1)^{x\cdot y}|y\rangle_B$$

If Bob now applies Hadamard gates to his n qubits, his register becomes x! Hence n bits have been transmitted from Alice to Bob.

(c) Holevo's theorem implies that transmitting n bits of information (without prior entanglement) takes at least n qubits of communication. Since the protocol of (b) takes only $2q$ qubits of communication, we must have $2q \geq n$, hence $q \geq n/2$.

5. (a) Note that $|\overline{0}\rangle$ and $|\overline{1}\rangle$ are superpositions of 4-bit basis states of even parity (and hence so is $\alpha|\overline{0}\rangle + \beta|\overline{1}\rangle$). A bitflip on any one of the 4 code qubits makes all these parities odd. We can detect this as follows. Add an auxiliary qubit that's initially $|0\rangle$. Apply 4 CNOTs, with the control qubit being the 4 code qubits, respectively, and the auxiliary qubit as target each time. Measure the auxiliary qubit. If the outcome is 1 then we've detected a bitflip on one of the 4 qubits of the code (we don't know which one), and if the outcome is 0 we conclude that there was no bitflip.

Alternatively you could use one auxiliary qubit to detect a bitflip among the first 2 qubits of the code, and another to detect a bitflip among the last 2 qubits of the code.

The advantage is that we can now detect, for instance, a bitflip on the 1st qubit of the code and a simultaneous bitflip on the 3rd qubit, which we didn't detect before. The disadvantage is that we're using two auxiliary qubits instead of one.

(b) The idea is to compare the relative phase (± 1) in the first 2-qubit block of the code with that of the second 2-qubit block. Do a CNOT with the 1st qubit of the code as control and the 2nd as target, and then a Hadamard on the 1st qubit. Do the same with the 3rd and 4th qubits of the code. If no phaseflip happened then now the 1st and 3rd qubits are in state $\alpha|00\rangle + \beta|11\rangle$. Add an auxiliary $|0\rangle$-qubit, and apply a CNOT with the 1st qubit as control and the auxiliary qubit as target, and another CNOT with the 3rd qubit as control and again the auxiliary qubit as target. Measure the auxiliary qubit. If the outcome is 1 then we've detected a phaseflip on one of the 4 qubits of the code (we don't know which one), and if the outcome is 0 then we conclude there was no phaseflip, and we reverse the above CNOTs and Hadamards.

(c) Yes, because any unitary U is a linear combination of the 4 Pauli matrices I, X, Y, Z. I corresponds to no-error, X corresponds to bitflip error, Z to phaseflip error, and Y to bitflip followed by phaseflip error. Running the procedures from (a) and (b) will detect any such error, and hence will also detect U.