Quantum Algorithms 2/16/2023

Lecture 8: Phase Kickback

Instructor: Dieter van Melkebeek

In today's lecture, we cover our first quantum algorithms! Many of the computational problems for which we develop quantum algorithms in this course can be cast in the blackbox framework. We introduce the framework and present our first technique in it, the "phase kickback." We then apply the technique to two problems: determining whether the blackbox function is constant or balanced, and learning linear functions.

1 Solution to Exercise #6

The exercise was to show that for any two unitary matrices Q and $\tilde{Q}$ acting on k qubits,

$$\|\tilde{Q} \otimes I - Q \otimes I\|_2 = \|\tilde{Q} - Q\|_2,$$

where I denotes the identity matrix of dimension 2^{m-k} (corresponding to the remaining $m-k$ qubits of the system). This is equivalent to $\|(\tilde{Q} - Q) \otimes I\|_2 = \|\tilde{Q} - Q\|_2$. We'll more generally show that for any matrix A, $\|A \otimes I\|_2 = \|A\|_2$. We describe two solutions.

Via the operator norm definition. Consider any vector of 2-norm one in the domain of $A \otimes I$. We can write it as $|\psi\rangle = \sum_s \alpha_s |\phi_s\rangle |s\rangle$, where the sum is over all basis states in the domain of I and $\sum_a |\alpha_s|^2 = 1$. The terms $\alpha_s |\phi_s\rangle$ corresponds to the projection of $|\psi\rangle$ onto the subspace corresponding to $|s\rangle$. In our projection notation: $P_s |\psi\rangle = \alpha_s |\phi_s\rangle$. When we measure the last $m-k$ qubits, the probability of obtaining $|s\rangle$ equals $\alpha_s|^2$, and if so, the state of the first part becomes $|\phi_s\rangle$. We have that

$$\begin{aligned}
\|(A \otimes I) |\psi\rangle \|_2^2 &= \| \sum_s \alpha_s (A \otimes I) |\phi_s\rangle |s\rangle \|_2^2 && \text{(linearity)} \\
&= \| \sum_s \alpha_s A |\phi_s\rangle |s\rangle \|_2^2 && \text{(tensor product)} \\
&= \sum_s \|\alpha_s A |\phi_s\rangle |s\rangle \|_2^2 && \text{(Pythagorean theorem)} \\
&= \sum_s |\alpha_s|^2 \|A |\phi_s\rangle \|_2^2 && \text{(absolute homogeneity and definition)} \\
&\leq \sum_s |\alpha_s|^2 \|A\|_2^2 && \text{(definition operator norm)} \\
&= \|A\|_2^2 && \text{(state property)}
\end{aligned}$$

Moreover, the inequality becomes an equality in case all $|\phi_s\rangle$ with nonzero amplitude α_s are vectors for which $\|A |\phi_s\rangle \|_2 = \|A\|_2$, which exist by the definition of the operator norm. By applying the definition of the operator norm one more time, it follows that $\|A \otimes I\|_2 = \|A\|_2$.

Via the singular value decomposition. For ease of notation when working with tensor products, we consider $I \otimes A$ instead of $A \otimes I$. Since one can be obtained from the other by permuting rows and columns, $\|A \otimes I\|_2 = \|I \otimes A\|_2$.

Consider the singular value decomposition $A = U\Sigma V^*$. By matrix tensor and product properties we have that

$$\begin{aligned}
I \otimes A &= \begin{bmatrix} A & 0 & 0 & \dots & 0 \\ 0 & A & 0 & \dots & 0 \\ 0 & 0 & A & \dots & 0 \\ \vdots & \vdots & & \ddots & 0 \\ 0 & 0 & 0 & \dots & A \end{bmatrix} \\
&= \begin{bmatrix} U\Sigma V^* & 0 & 0 & \dots & 0 \\ 0 & U\Sigma V^* & 0 & \dots & 0 \\ 0 & 0 & U\Sigma V^* & \dots & 0 \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & 0 & \dots & U\Sigma V^* \end{bmatrix} \\
&= \begin{bmatrix} U & 0 & 0 & \dots & 0 \\ 0 & U & 0 & \dots & 0 \\ 0 & 0 & U & \dots & 0 \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & 0 & \dots & U \end{bmatrix} \begin{bmatrix} \Sigma & 0 & 0 & \dots & 0 \\ 0 & \Sigma & 0 & \dots & 0 \\ 0 & 0 & \Sigma & \dots & 0 \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & 0 & \dots & \Sigma \end{bmatrix} \begin{bmatrix} V^* & 0 & 0 & \dots & 0 \\ 0 & V^* & 0 & \dots & 0 \\ 0 & 0 & V^* & \dots & 0 \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & 0 & \dots & V^* \end{bmatrix} \\
&\doteq U'\Sigma'(V')^*
\end{aligned}$$

Note that U' and V' are unitaries, and Σ' a diagonal matrix with nonnegative entries. Hence, $U'\Sigma'V'$ represents a singular value decomposition of $I \otimes A$. By the uniqueness properties of singular value decompositions (namely that in any such decomposition the diagonal matrix needs to contain the singular values), it follows that the largest singular values of A and $I \otimes A$ are the same, and thus are the 2-norms of A and $I \otimes A$. In fact, the decomposition shows that the singular values of $I \otimes A$ are those of A taken the dimension of I many times.

2 Blackbox algorithms

In classical blackbox problems, there is an underlying function $f : \{0,1\}^n \to \{0,1\}^l$ and we are expected to determine some property of it. In doing so, we can make queries to the blackbox. In a single query, we select any input in $\{0,1\}^n$ and receive the result of applying f to our input. In probabilistic blackbox algorithms, the queries can depend on randomness (as well as on the outcomes of prior queries), which can be viewed as querying the blackbox in a probabilistic superposition.

In the quantum version, we are instead given an access to the "revisibilification" of f, namely $\tilde{f} : \{0,1\}^n \times \{0,1\}^l \to \{0,1\}^n \times \{0,1\}^l$ because only reversible deterministic computations can be performed in the quantum setting. We can query $\tilde{f}$ in quantum superposition, i.e., we can apply the following unitary operator on any pure state: $U_f : |x\rangle\,|y\rangle \mapsto |x\rangle\,|y \oplus f(x)\rangle$, where $x \in \{0,1|^n$ and $y \in \{0,1\}^l$. Note that U_f is its own inverse.

The *query complexity* is the minimum number of applications of the blackbox needed over all possible algorithms for the problem. The goal is to determine this quantity and to find an algorithm that achieves it.

A few caveats – Query complexity does not take into account non-query operations. In other words, we don't consider how fast the algorithm is including all other operations besides the calls to the blackbox. We also ignore how to implement U_f, which may not be trivial. Finally, when comparing quantum to classical query complexity numerically we are implicitly ignoring the difference in query mechanism.

That said, many tight bounds are known for both classical and quantum query complexity. Those allow us to demonstrate gaps between the power of deterministic, probabilistic, and quantum blackbox algorithms.

3 Phase kickback

Consider a single-output function $f : \{0,1\}^n \to \{0,1\}$, i.e., we set $l = 1$. We want to design a quantum circuit that realizes the unitary mapping $|x\rangle \mapsto (-1)^{f(x)} |x\rangle$ for every $x \in \{0,1\}^n$, i.e., we want to encode the value of f into the phase. Note that changing the phase on an a single basis state $|x\rangle$ has no physical effect, but it does in superposition. By linearity, a superposition $\sum_x \alpha_x |x\rangle$ is transformed into $\sum_x \alpha_x (-1)^{f(x)} |x\rangle$.

As the blackbox U_f needs one extra qubit for the output, we make use of a second register consisting of a single qubit. In order to achieve the intended transformation on the input register, the output register should be unentangled with the input register. We use the output qubit as an ancilla, whose state before after the operation is the same. That is, we effect the transformation from $\sum_x \alpha_x |x\rangle |\phi\rangle$ to $\sum_x \alpha_x (-1)^{f(x)} |x\rangle |\phi\rangle$, where $|\phi\rangle$ represents the state of the ancilla before and after the operation.

By definition, the blackbox U_f maps the basis state $|x\rangle |y\rangle$ to $|x\rangle |y \oplus f(x)\rangle$. As such, it leaves the first register untouched. The other operations in our quantum circuit will only involve the ancilla. The result will be a phase change from $|\phi\rangle$ to $(-1)^{f(x)} |\phi\rangle$. In reality, the phase change applies to the combined state $|x\rangle |\phi\rangle$. Instead of as a phase change to the ancilla $|\phi\rangle$, the effect also be interpreted as a phase change on the first register from basis stated $|x\rangle$ to $(-1)^{f(x)} |x\rangle$ while the ancilla remains totally unaffected (not even a phase change). In some sense the phase change is "kicked back" from the ancilla to the first register. Hence the name "phase kickback."

We present two algorithms, one with two blackbox queries and one with a single query. The state $|\phi\rangle$ of the ancilla differs between the two solutions.

Two query solution. We use $|\phi\rangle = |0\rangle$. The algorithm is built on the observation that $Z |0\rangle = |0\rangle$ and $Z |1\rangle = (-1) |1\rangle$ where Z is a Pauli-Z gate. Hence, applying Z to the ancilla qubit of the state $|x\rangle |f(x)\rangle$ results in $(-1)^{f(x)} |x\rangle |f(x)\rangle$.

The algorithm utilizes this observation as follows. Starting from a state $|x\rangle |0\rangle$, the state immediately after applying U_f is $|x\rangle |f(x)\rangle$. We then apply Z to the ancilla qubit to get $(-1)^{f(x)} |x\rangle |f(x)\rangle$. Finally, applying U_f again will revert the ancilla qubit making it be $(-1)^{f(x)} |x\rangle |0\rangle$, which is what we want our final state to be. See Figure 1

Single query solution. If we want to have a solution with a single application of U_f, then the state $|\phi\rangle$ of the ancilla should be such that $|x\rangle |\phi\rangle$ is an eigenstate of U_f for each $x \in \{0,1\}^n$. This

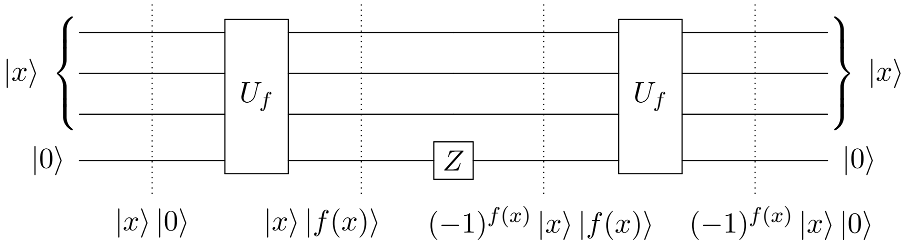

Figure 1: Two query phase kickback with $|\phi\rangle = |0\rangle$

is the case for $|+\rangle$ and $|-\rangle$. The eigenvalue corresponding to $|+\rangle$ is 1, which is not useful. The eigenvalue corresponding to $|-\rangle$ is the one we want.

Recall that $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, hence $|x\rangle|-\rangle = \frac{1}{\sqrt{2}}(|x\rangle|0\rangle - |x\rangle|1\rangle)$. Starting with this state, we apply U_f to it to get $\frac{1}{\sqrt{2}}(|x\rangle|f(x)\rangle - |x\rangle|\overline{f(x)}\rangle)$. The state reduces to $|x\rangle|-\rangle$ when $f(x) = 0$ and to $(-1)|x\rangle|-\rangle$ when $f(x) = 1$. Therefore, we achieve the state $(-1)^{f(x)}|x\rangle|-\rangle$ using only one query to U_f. See Figure 2.

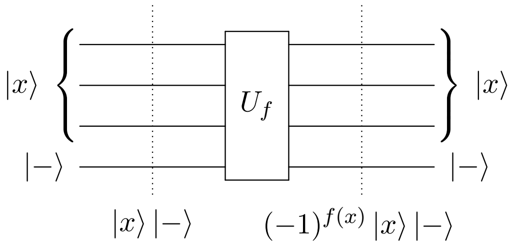

Figure 2: Single query phase kickback with $|\phi\rangle = |-\rangle$

4 Distinguishing constant from balanced

Consider a function $f : \{0,1\}^n \to \{0,1\}$ and we want to distinguish between the cases where f is "constant" and "balanced". We call f constant if it is identically 0 or 1. We call f balanced if the number of 0's and 1's that f maps to is equal.

Promise problems. Note that the problem we are considering here is only partially specified. In particular, the output is not specified in cases where f is neither constant nor balanced. For $n = 1$ there are no such cases, but there are for $n > 1$. Such partially specified problems are often called "promise problems," where the term "promise" refers to the fact that the input to the problems is guaranteed to satisfy some property that not all possible inputs (necessarily) satisfy. In this case the promise is that "f is either constant or balanced". For $n = 1$ this includes all possible functions, but not for $n > 1$.

The difference between promise problems and fully specified problems is important in query complexity. For promise problems exponential gaps are known between quantum query complexity and classical query complexity. For fully specified problems only polynomial gaps are possible. In modern lingo, for promise problems quantum supremacy is possible in terms of query complexity, whereas for fully specified problems only quantum advantage can be achieved.

4.1 Deutsch algorithm: $n = 1$

Classically, we need two queries to solve the problem, even when using randomness. This is because a single query only gives us information about the value at one point, from which we can draw no information about whether f is constant or balanced. Two queries, namely $f(0)$ and $f(1)$ are trivially sufficient.

Quantumly, we can do it with a single query. Recall that, when we apply phase kickback to states $|0\rangle|-\rangle$ and $|1\rangle|-\rangle$, we get $(-1)^{f(0)}|0\rangle|-\rangle$ and $(-1)^{f(1)}|1\rangle|-\rangle$ respectively. Thus, if we apply this phase kickback technique to the superposition state $|+\rangle|-\rangle$, we have

$$U_f|+\rangle|-\rangle = \frac{1}{\sqrt{2}}(U_f|0\rangle|-\rangle + U_f|1\rangle|-\rangle) = \frac{1}{\sqrt{2}}((-1)^{f(0)}|0\rangle|-\rangle + (-1)^{f(1)}|1\rangle|-\rangle)$$

Notice that, if $f(0) = f(1)$, the state ends up being $(-1)^{f(0)}|+\rangle|-\rangle = \pm|+\rangle|-\rangle$. On the other hand, if $f(0) \neq f(1)$, the state reduces to $(-1)^{f(0)}|-\rangle|-\rangle = \pm|-\rangle|-\rangle$. Consider the two possible states of the non-ancilla qubit, $|+\rangle$ and $|-\rangle$; these states are orthogonal. Hence, we can find a transformation that maps one of them to $|0\rangle$ and the other to $|1\rangle$. Such transformation is the Hadamard gate which maps $|+\rangle$ to $|0\rangle$ and $|-\rangle$ to $|1\rangle$. This mapping allows us to determine the property of f by measuring the non-ancilla qubit which can either be $|0\rangle$ or $|1\rangle$ with no error.

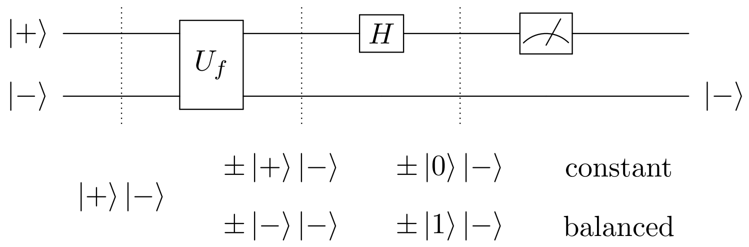

Figure 3: Deutsch algorithm

4.2 Deutsch-Jozsa algorithm: $n > 1$

This algorithm also utilizes the phase kickback technique like we did in the case with $n = 1$. Now, we start with the state $|\psi\rangle = |+\rangle|+\rangle \dots |+\rangle|-\rangle$, which results in the uniform superposition for the first register:

$$|\psi\rangle = \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \dots \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right)|-\rangle = \frac{1}{\sqrt{2^n}}\left(\sum_{x\in\{0,1\}^n}|x\rangle\right)|-\rangle$$

Applying a phase kickback technique U_f to $|\psi\rangle$, we have

$$U_f|\psi\rangle = \frac{1}{\sqrt{2^n}}\left(\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x\rangle\right)|-\rangle$$

Notice that when f is constant, the state reduces to

$$U_f \left|\psi\right\rangle = \frac{1}{\sqrt{2^n}} \left(\sum_{x\in\{0,1\}^n} (-1)^{f(0)} \left|x\right\rangle \right) \left|-\right\rangle = \pm \left|+\right\rangle \left|+\right\rangle \dots \left|+\right\rangle \left|-\right\rangle$$

The state $U_f \left|\psi\right\rangle$ has a more complicated form when f is not constant. Still, we know that, if f_1 is any constant function and f_2 is any balanced function, the states $\left|\psi_1\right\rangle \doteq U_{f_1} \left|\psi\right\rangle$ and $\left|\psi_2\right\rangle \doteq U_{f_2} \left|\psi\right\rangle$ are orthogonal. Indeed, since f_2 is balanced, $\sum_x (-1)^{f_2(x)} = 0$, and since the state of the ancilla in both $\left|\psi - 1\right\rangle$ and $\left|\psi_2\right\rangle$ is the same, the inner product of $\left|\psi\right\rangle_1$ and $\left|\psi\right\rangle f_2$ equals $\frac{1}{2^n} \sum_x (-1)^{f_1(0)} \cdot (-1)^{f_2(x)} = 0$. The inner product being zero means that $\left|\psi_1\right\rangle$ and $\left|\psi_2\right\rangle$ are orthogona and, in fact, the states of the first register are orthogonal. Hence, there exists a transformation that maps the only state when f is constant, $\left|+\right\rangle^{\otimes n}$, to the known basis $\left|0^n\right\rangle$ and the states when f is balanced to states with no component along $\left|0^n\right\rangle$. Measuring the first register then yields 0^n for sure in case f is balanced, and a string other than 0^n in case f is balanced.

Such transformation is, again, a Hadamard gate applied to each qubit of the first register. (Recall that $H \left|+\right\rangle = \left|0\right\rangle$.) See Figure 4 for the resulting circuit. It allows us to solve the problem with no error and only one query.

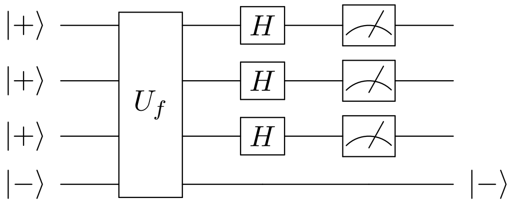

Figure 4: Deutsch-Jozsa algorithm

Note that we can achieve $\left|+\right\rangle$ by applying Hadamard gate to $\left|0\right\rangle$. Therefore, our circuit may be implemented as in Figure 5 where $H^{\otimes n}$ denotes a Hadamard tensor gate which is applying Hadamard gate to each qubit.

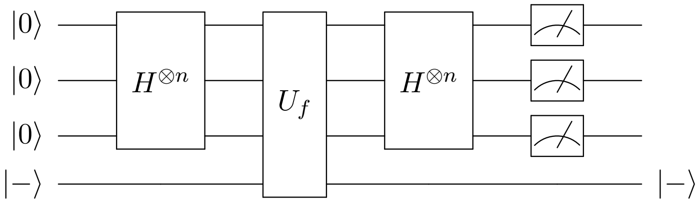

Figure 5: Deutsch-Jozsa algorithm

Query complexity. As we showed, in the quantum setting one query suffices to solve the problem with 100% certainty. What about classical query complexity?

Deterministically, we need $N/2 + 1$ queries where $N = 2^n$. Intuitively, this is because iwhenwe make only $N/2$ queries, they could be all 0 in both the constant 0 and balanced cases, in which case we don't know whether the underlying function f is identically zero or balanced; the next query will determine it for sure. The lower bound can be formalized as an adversary argument, where given a candidate deterministic algorithm, we construct two instances f_1 and f_2 on which the algorithm performs the same but where the answers are different. In this case, we run the purported algorithm and answer the first $N/2$ queries all with 0. (If the algorithm queries fewer points, query some arbitrary additional ones.) Now, let f_1 be the function that is identically zero, and f_2 the function that is zero on all the $N/2$ points queried thus far and one everywhere else. Then f_1 is constant and f_2 balanced, but the algorithm outputs the same answer for both.

We can also consider a probabilistic algorithm that uses k queries to find the correct answer with probability at least $1-1/2^{k-1}$. Pick the queries uniformly at random. If the values are not the same, then the function is balanced for sure. Otherwise, we can't conclude the property for sure but we guess that it is constant. The probability that we guess wrong is the probability that all k random queries yield the same answer in case f is balanced. This probability is $(1/2)^{k-1}$ since, when f is balanced, for each query after the first the probability that it returns the same value as the first is $1/2$, and these events are independent. This simple probabilistic algorithm is pretty much the best one can do in this, as the following exercise shows.

Exercise (intended for theory students only). Show that every probabilistic algorithm with k queries has error $\Omega(1/2^k)$. *Hint:* Use Yao's Principle.

5 Hadamard tensor

The Hadamard tensor $H^{\otimes n}$ is the operation of n Hadamard gates on n qubits state. We already saw that we can create a uniform superposition on n qubits by applying $H^{\otimes n}$ to $|0^n\rangle$.

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

What if we apply $H^{\otimes n}$ to a generic basis state $|x\rangle$ for $x = x_1x_2\dots x_n \in \{0,1\}^n$?

$$\begin{aligned}
H^{\otimes n}|x\rangle &= H|x_1\rangle H|x_2\rangle \dots H|x_n\rangle \\
&= \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_1}|1\rangle \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_2}|1\rangle)\dots\frac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_n}|1\rangle) \\
&= \frac{1}{\sqrt{2}}\left(\sum_{y_1=0}^{1}(-1)^{x_1y_1}|y_1\rangle\right)\frac{1}{\sqrt{2}}\left(\sum_{y_2=0}^{1}(-1)^{x_2y_2}|y_2\rangle\right)\dots\frac{1}{\sqrt{2}}\left(\sum_{y_n=0}^{1}(-1)^{x_ny_n}|y_n\rangle\right) \\
&= \frac{1}{\sqrt{N}}\sum_{y\in\{0,1\}^n}(-1)^{x\cdot y}|y\rangle\,, \qquad (1)
\end{aligned}$$

where $x \cdot y \doteq \sum_{i=1}^{n} x_i y_i$.

6 Learning linear functions

Consider the problem where we are promised that the blackbox function $f : \{0,1\}^n \to \{0,1\}$ is of the form $f(x) = a \cdot x \bmod 2$ for some $a \in \{0,1\}^n$. Our goal is to find a using blackbox access to f.

We claim that our quantum circuit for the Deutsch-Josza problem solves our new problem exactly when we return the outcome of the final measurement as our candidate for a.

this problem is exactly the same for the one for distinguishing constant from balanced, but the final observation is going to be our a. Since $f(x) = a \cdot x \bmod 2$, we can rewrite $(-1)^{f(x)} = (-1)^{a \cdot x \bmod 2} = (-1)^{a \cdot x}$. Thus, the state after applying U_f is

$$\frac{1}{\sqrt{2^n}} \sum_x (-1)^{f(x)} |x\rangle |-\rangle = \frac{1}{\sqrt{2^n}} \sum_x (-1)^{a \cdot x} |x\rangle |-\rangle . \tag{2}$$

Comparing the right-hand side of (2) with (1), we can rewrite (2) as $H^{\otimes n} |a\rangle |-\rangle$. As the Hadamard gate is its own inverse, apply $H^{\otimes n}$ to the first n qubits yields the state $|a\rangle |-\rangle$. Measuring the first n qubits of this state yields a with certainty. Thus, we can find a exactly with a single quantum query.

The resulting quantum algorithm is commonly used as a benchmark for quantum computers. To do so, the blackbox needs to be realized, for a given a, in terms of elementary gates, as well as the other operations. The following exercise shows how it can be done efficiently and yields another explanation for why the algorithm works.

Exercise #7. Fix $a \in \{0,1\}^n$.

(a) Implement U_f for $f(x) = a \cdot x \bmod 2$ using CNOTs only.

(b) Show that $H^{\otimes 2} \circ \text{CNOT} \circ H^{\otimes 2}$ is equivalent to a CNOT with the control swapped.

(c) Use (b) to reduce the number of elementary gates in the resulting quantum circuit.

Query complexity. We can deterministically find a using n queries by finding $f(100\ldots0)$, $f(010\ldots0)$, ..., $f(0\ldots01)$, which tells us a_1, a_2, ..., a_n. Note that n queries is optimal since we can retrieve at most one bit of information from each deterministic query, and there are 2^n possibilities for a. Also, every probabilistic algorithm with error less than 1/2 needs to make at least n queries. We leave the proof as an exercise. Our quantum algorithm improves the query complexity to just one query with no error.

Exercise (intended for theory students only). Show that every probabilistic algorithm with $n-1$ queries has error at least 1/2. Hint: use Yao's Principle.