Introduction to Quantum Information Processing

Lecture 8

Richard Cleve

Overview of Lecture 8

- BV problem: **1** vs. ***n*** separation robust against probabilistic algorithms
- Preview of where black-box results are headed: period-finding
- Simulating black boxes
- Simon's problem: **1** vs. $\mathbf{2}^{n/2}$ separation robust against probabilistic algorithms

Quantum vs. classical separations

black-box problem	quantum	classical	
constant vs. balanced	**1** (query)	**2** (queries)	
1-out-of-4 search	**1**	**3**	
constant vs. balanced	**1**	**$\frac{1}{2} 2^n + 1$**	**(only for exact)**
BV problem	**1**	***n***	**(probabilistic)**

BV problem

BV problem

[Bernstein & Vazirani, 1993]

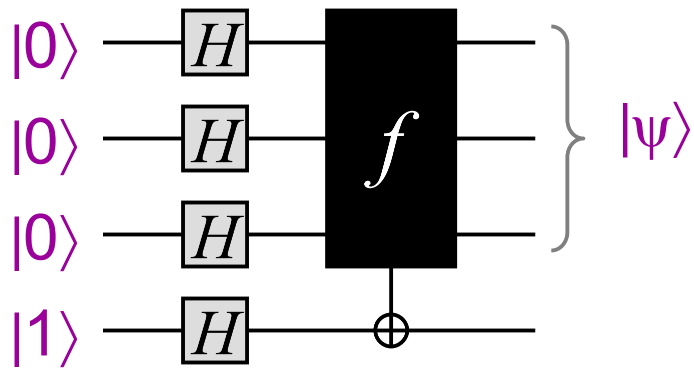
Let $f : \{0,1\}^n \rightarrow \{0,1\}$ be of the form $f(x) = a_1 x_1 \oplus \ldots \oplus a_n x_n$, where $(a_1, \ldots , a_n) \in \{0,1\}^n$ is unknown

Goal: determine $(a_1, \ldots , a_n)$

Classically: n queries needed, even to succeed with probability > ½ (why?)

Quantumly: 1 query suffices

Quantum algorithm for BV



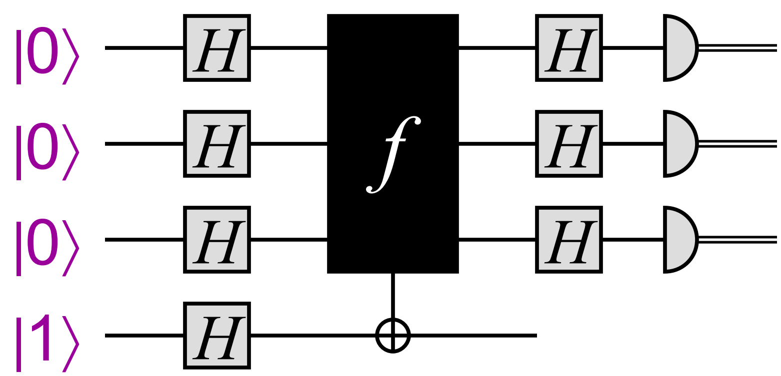
where $|\psi\rangle = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} (-1)^{a \bullet x} |x\rangle$

Question: what is $|\psi\rangle$?

Answer: $|\psi\rangle = H^{\otimes n} |a_1, \ldots, a_n\rangle$

Therefore, $H^{\otimes n} |\psi\rangle = |a_1, \ldots, a_n\rangle$

Final algorithm:



Quantum vs. classical separations

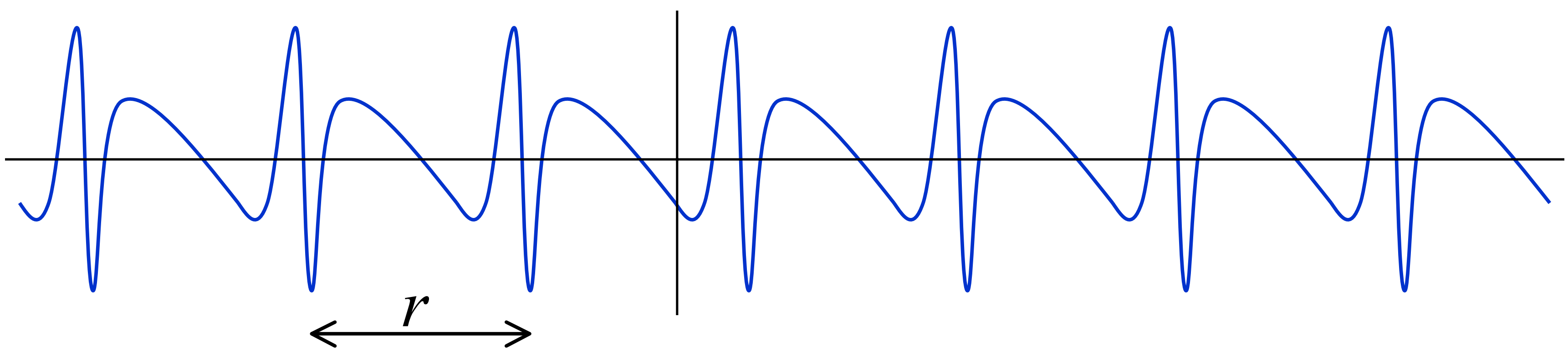
black-box problem	quantum	classical	
constant vs. balanced	**1** (query)	**2** (queries)	
1-out-of-4 search	**1**	**3**	
constant vs. balanced	**1**	**$\frac{1}{2} 2^n + 1$**	**(only for exact)**
BV problem	**1**	***n***	**(probabilistic)**
Simon's problem	**$O(n)$**	**$\Omega(2^{n/2})$**	**(probabilistic)**

Before getting into Simon's problem: where are all these black-box results headed?

preview of applications of black-box results

Period-finding

Given: $f : \mathbf{Z} \rightarrow \mathbf{Z}$ such that f is (strictly) r-periodic, in the sense that $f(x) = f(y)$ iff $x - y$ is a multiple of r (unknown)



Goal: find r

Classically, the number of queries required can be ***"huge"*** (essentially as hard as finding a collision)

There is a quantum algorithm that makes only a ***constant*** number of queries (which will be explained later on)

Application of period-finding algorithm

Order-finding problem: given a and m (positive integers such that $\gcd(a,m) = 1$), find the minimum positive r such that $a^r \bmod m = 1$

Note that this is ***not*** a black-box problem!

No classical polynomial-time algorithm is known for this problem (in fact, the factoring problem reduces to it)

The problem reduces to finding the period of $f(x) = a^x \bmod m$, and the aforementioned period-finding algorithm in the black-box model can be used to solve it in polynomial-time

The function f is substituted into the black-box ...

on simulating black boxes

How *not* to simulate a black box

Given an explicit function, such as $f(x) = a^x \bmod m$, and a finite domain $\{0, 1, 2, ..., 2^n - 1\}$, simulate f-queries over that domain

Easy to compute mapping $|x\rangle|y\rangle|00...0\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle|g(x)\rangle$, where the third register is "work space" with accumulated "garbage" (e.g., two such bits arise when a Toffoli gate is used to simulate an AND gate)

This works fine as long as f is not queried in superposition

If f is queried in superposition then the resulting state can be $\sum_x \alpha_x |x\rangle|y \oplus f(x)\rangle|g(x)\rangle$ (can we just discard the third register?)

No ... there could be entanglement ...

How *to* simulate a black box

Simulate the mapping $|x\rangle|y\rangle|00\ldots0\rangle \rightarrow |x\rangle|y\oplus f(x)\rangle|00\ldots0\rangle$,
(i.e., clean up the "garbage")

To do this, use an additional register and:

1. compute $|x\rangle|y\rangle|00\ldots0\rangle|00\ldots0\rangle \rightarrow |x\rangle|y\rangle|f(x)\rangle|g(x)\rangle$
   (ignoring the 2^{nd} register in this step)
2. compute $|x\rangle|y\rangle|f(x)\rangle|g(x)\rangle \rightarrow |x\rangle|y\oplus f(x)\rangle|f(x)\rangle|g(x)\rangle$
   (using CNOT gates between the 2^{nd} and 3^{rd} registers)
3. compute $|x\rangle|y\oplus f(x)\rangle|f(x)\rangle|g(x)\rangle \rightarrow |x\rangle|y\oplus f(x)\rangle|00\ldots0\rangle|00\ldots0\rangle$
   (by reversing the procedure in step 1)

Total cost: around twice the cost of computing f, plus n auxiliary gates

Simon’s problem

Simon's problem

Let $f : \{0,1\}^n \rightarrow \{0,1\}^n$ have the property that there exists an $r \in \{0,1\}^n$ such that $f(x) = f(y)$ iff $x \oplus y = r$ or $x = y$

Example:

x	$f(x)$
000	011
001	101
010	000
011	010
100	101
101	011
110	010
111	000

Question: what is r is this case?

Answer: r = 101

Simon's problem vs. period-finding

Period-finding problem: domain is $\mathbf{Z}$ and property is $f(x) = f(y)$ iff $x - y$ is a multiple of r

This problem meaningfully generalizes to domain $\mathbf{Z}^n$

Deutsch's problem: domain is $\mathbf{Z}_2$ and property is $f(x) = f(y)$ iff $x \oplus y$ is a multiple of r
($r = 0$ means $f(0) = f(1)$ and $r = 1$ means $f(0) \neq f(1)$)

Simon's problem: domain is $(\mathbf{Z}_2)^n$ and property is $f(x) = f(y)$ iff $x \oplus y$ is a multiple of r

A classical algorithm for Simon

Search for a ***collision***, an $x \neq y$ such that $f(x) = f(y)$

1. Choose $x_1, x_2, \ldots, x_k \in \{0,1\}^n$ randomly (independently)
2. For all $i \neq j$, if $f(x_i) = f(x_j)$ then output $x_i \oplus x_j$ and halt

A hard case is where r is chosen randomly from $\{0,1\}^n - \{0^n\}$ and then the "table" for f is filled out randomly subject to the structure implied by r

Question: how big does k have to be for the probability of a collision to be a constant, such as ¾?

Answer: order $2^{n/2}$ (each (x_i, x_j) collides with prob. $O(2^{-n})$)

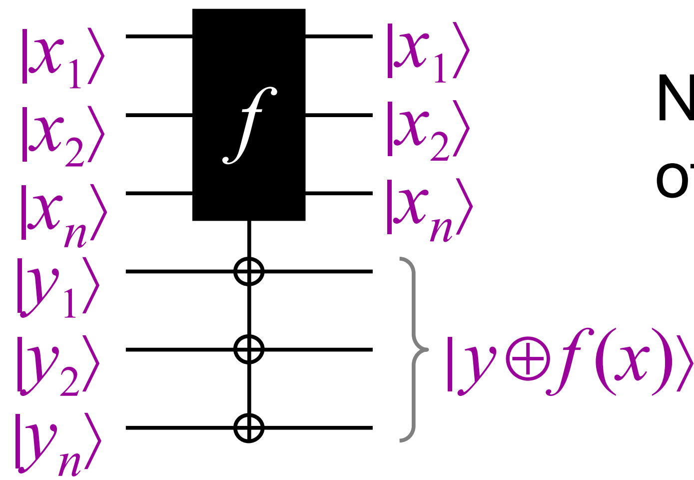
Classical lower bound

Theroem: ***any*** classical algorithm solving Simon's problem must make $\Omega(2^{n/2})$ queries

Proof is omitted here—note that the performance analysis of the previous algorithm does ***not*** imply the theorem

A quantum algorithm for Simon

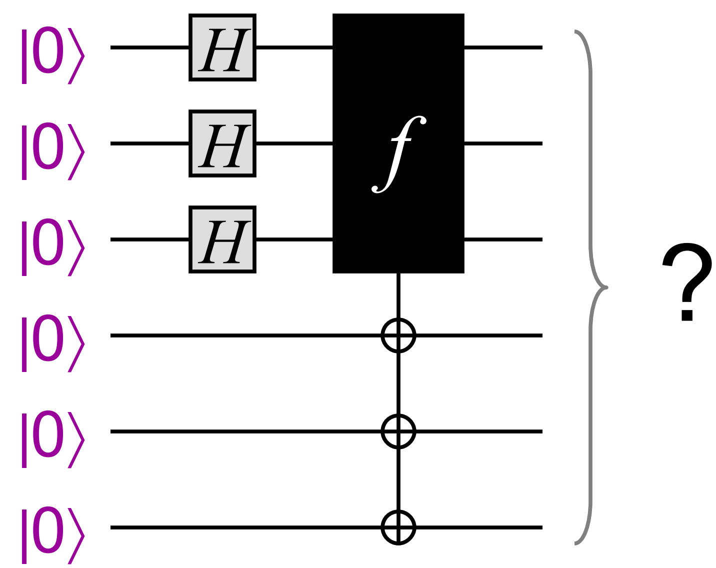
Queries:



Not clear what ***eigenvector*** of target registers is ...

Proposed start of quantum algorithm: query all values of f in superposition

Question: what is the output state of this circuit?



A quantum algorithm for Simon

Answer: the output state is $\sum_{x\in\{0,1\}^n} |x\rangle|f(x)\rangle$

Let $T \subseteq \{0,1\}^n$ be such that ***one*** element from each matched pair is in T (assume $r \neq$ 00...0)

Example: could take $T = \{$000, 001, 111, 011$\}$

Then the output state can be written as:

$$\sum_{x\in T} |x\rangle|f(x)\rangle + |x\oplus r\rangle|f(x\oplus r)\rangle$$

$$= \sum_{x\in T} \left(|x\rangle + |x\oplus r\rangle\right)|f(x)\rangle$$

x	$f(x)$
000	011
001	101
010	000
011	010
100	101
101	011
110	010
111	000

A quantum algorithm for Simon

Measuring the second register yields $|x\rangle + |x \oplus r\rangle$ in the first register, for a random $x \in T$

How can we use this to obtain ***some*** information about r?

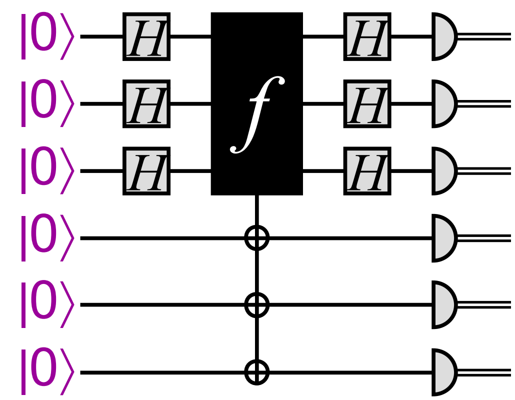
Try applying $H^{\otimes n}$ to the state, yielding:

$$\sum_{y \in \{0,1\}^n} (-1)^{x \bullet y} |y\rangle + \sum_{y \in \{0,1\}^n} (-1)^{(x \oplus r) \bullet y} |y\rangle$$

$$= \sum_{y \in \{0,1\}^n} (-1)^{x \bullet y} \left(1 + (-1)^{r \bullet y}\right) |y\rangle$$

Measuring this state yields y with prob. $\begin{cases} (1/2)^{n-1} & \text{if } r \cdot y = 0 \\ 0 & \text{if } r \cdot y \neq 0 \end{cases}$

A quantum algorithm for Simon



Executing this algorithm $k = O(n)$ times yields random $y_1, y_2, \ldots, y_k \in \{0,1\}^n$ such that $r \cdot y_1 = r \cdot y_2 = \ldots = r \cdot y_n = 0$

This is a system of k linear equations:

$$\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k1} & y_{k2} & \cdots & y_{kn} \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

With high probability, there is a unique non-zero solution that is r (which can be efficiently found by linear algebra)