Introduction to Quantum Information Processing

CS 467 / CS 667
Phys 467 / Phys 767
C&O 481 / C&O 681

Lecture 5 (2005)

Richard Cleve
DC 653
cleve@cs.uwaterloo.ca

Course web site at:
http://www.cs.uwaterloo.ca/~cleve

Contents

- Continuation of Simon's problem
- Preview of applications of black-box results
- On simulating black boxes

- Continuation of Simon’s problem
- Preview of applications of black-box results
- On simulating black boxes

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**	**½ 2^n + 1**	**(only for exact)**
Simon's problem	***$O(n)$***	**$\Omega(2^{n/2})$**	**(probabilistic)**

Simon's problem

Let $f : \{\mathbf{0},\mathbf{1}\}^n \rightarrow \{\mathbf{0},\mathbf{1}\}^n$ have the property that there exists an $r \in \{\mathbf{0},\mathbf{1}\}^n$ 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**

What is r in this case?

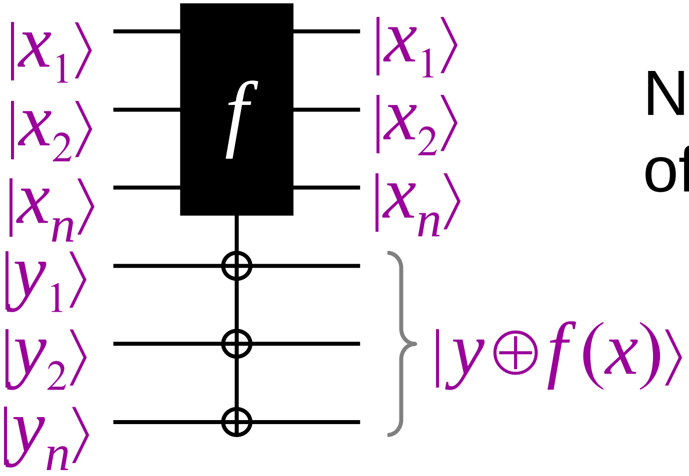
Answer: r = **101**

Classical lower bound

Theorem: ***any*** classical algorithm solving Simon's problem must make $\Omega(2^{n/2})$ queries, to succeed with probability $\geq$ ¾

A *quantum* algorithm for Simon I

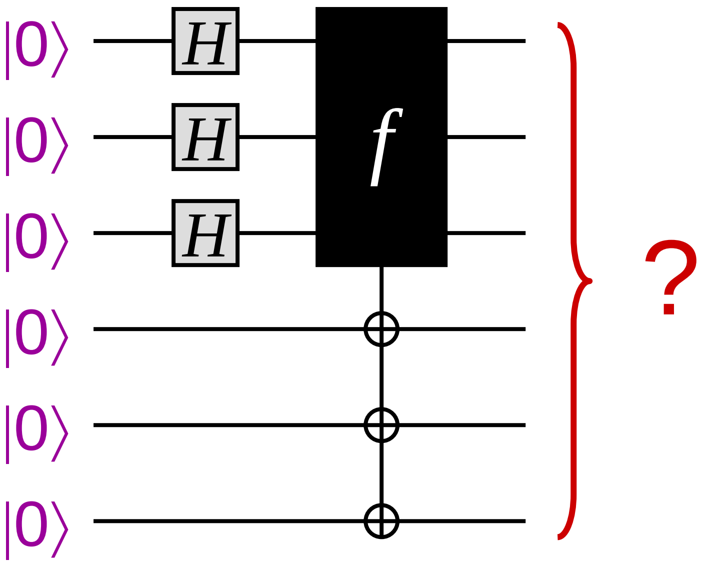
Queries:



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

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

What is the output state of this circuit?



A quantum algorithm for Simon II

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

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

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

Then the output state can be written as:

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

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

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

A quantum algorithm for Simon III

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?

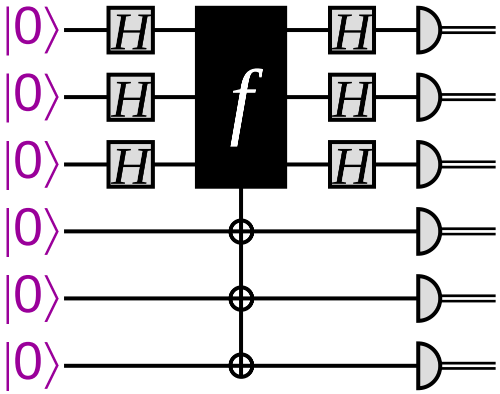
Try applying $H^{\otimes n}$ to the state, yielding:

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

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

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 IV



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

How does this help?

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)

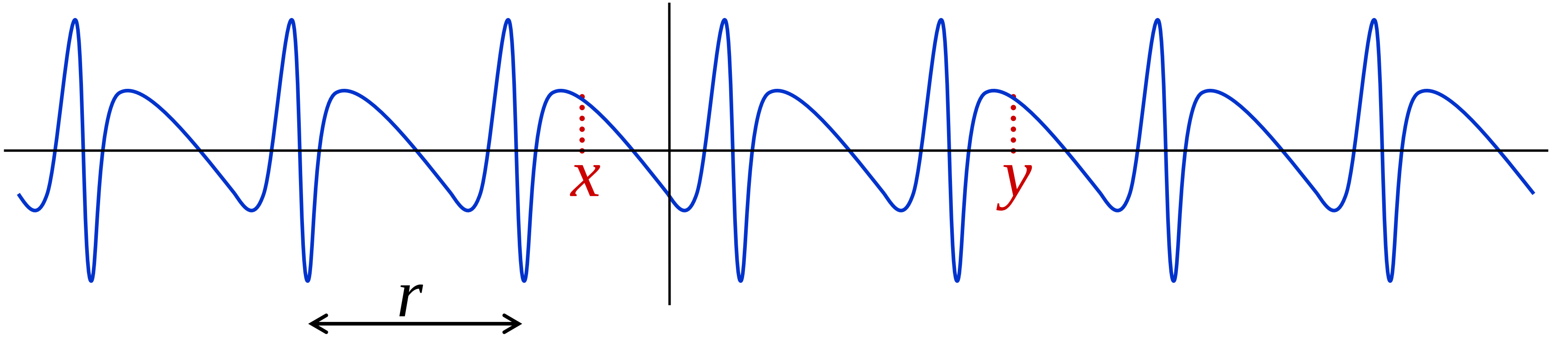
Conclusion of Simon's algorithm

- Any classical algorithm has to query the black box $\Omega(2^{n/2})$ times, even to succeed with probability ¾
- There is a quantum algorithm that queries the black box only $O(n)$ times, performs only $O(n^3)$ auxiliary operations (for the Hadamards, measurements, and linear algebra), and succeeds with probability ¾

- Continuation of Simon’s problem
- Preview of applications of black-box results
- On simulating black boxes

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)

Simon's problem vs. period-finding

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

This problem meaningfully generalizes to domain $\mathbf{Z}^n$, where the periodicity is multidimensional

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

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 $a^r \bmod m = 1$

Example: let $a = 4$ and $m = 35$
(note that $\gcd(4,35) = 1$)

In this case, $r = ?$

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

$4^1 \bmod 35 = 4$
$4^2 \bmod 35 = 16$
$4^3 \bmod 35 = 29$
$4^4 \bmod 35 = 11$
$4^5 \bmod 35 = 9$
$4^6 \bmod 35 = 1$
$4^7 \bmod 35 = 4$
$4^8 \bmod 35 = 16$
:

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 $a^r \bmod m = 1$

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

A circuit computing the function f is substituted into the black-box ...

- Continuation of Simon's problem
- Preview of applications of black-box results
- 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 $|x\rangle|y\rangle|00...0\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle|00...0\rangle$, (i.e., clean up the "garbage")

To do this, use an additional register and:

1. compute $|x\rangle|y\rangle|00...0\rangle|00...0\rangle \rightarrow |x\rangle|y\rangle|f(x)\rangle|g(x)\rangle$ (ignoring the 2nd 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 2nd and 3rd registers)
3. compute $|x\rangle|y \oplus f(x)\rangle|f(x)\rangle|g(x)\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle|00...0\rangle|00...0\rangle$ (by reversing the procedure in step 1)

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

THE END