

Lecture 6:

Finding the period of a function (Simon's algorithm and its generalisation)

Luís Soares Barbosa
www.di.uminho.pt/~lsb/



Universidade do Minho



INESC TEC



INTERNATIONAL IBERIAN
NANOTECHNOLOGY
LABORATORY



**UNU
EGOV**

Quantum Computation

Mestrado em Engenharia Física

Universidade do Minho, 2026-27

Recall: Query algorithms

Input accessed through an oracle

Input provided as a function $f : 2^n \longrightarrow 2^m$ that can be queried by the algorithm, which has, in this way, random access to segments of the input.

Example: the parity problem

Function f can be thought as a sequence of 2^n bits which can be accessed randomly through its evaluation. For example,

000	$\mapsto$	1
001	$\mapsto$	1
010	$\mapsto$	0
...		...

Recall: Phase kick-back

Typically, the oracle keeps the input (in the top qubit) unchanged, e.g.

$$|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle = |x\rangle X^{f(x)}|y\rangle$$

Phase kick-back is forced by supplying to the oracle's second qubit an **eigenvector** of X , thus

$$U_f|x\rangle|-\rangle = |x\rangle X^{f(x)}|-\rangle = (-1)^{f(x)}|x\rangle|-\rangle$$

What's for today?

Until now we have discussed examples with **moderate** gains in performance, typically counting the number of queries as a simple measure of efficiency.

A step ahead

- Another **query** algorithm,
- not making use of **phase kick-back**,
- which exhibits an effective **quantum advantage**, drawing a **exponential separation** wrt classical computation.

Simon's problem

The problem

Let $f : 2^n \longrightarrow 2^n$ be such that for some $s \in 2^n$,

$$f(x) = f(y) \text{ iff } x = y \text{ or } x = y \oplus s$$

Find s .

Exercise

What characterises f if $s = 0$? And if $s \neq 0$?

Simon's problem

Exercise

- f is **bijjective** if $s = 0$, because $y \oplus 0 = y$.
- f is **two-to-one** otherwise ,because, for a given s there is only a pair of values x, y such that $x \oplus y = s$.

Let us assume $s \neq 0$, and thus f to be **two-to-one**, and rewrite the problem as follows:

Equivalent formulation as a period-finding problem

Determine the period s of a function f **periodic** under $\oplus$:

$$f(x \oplus s) = f(x)$$

Simon's problem

Example

Let $f : 2^3 \rightarrow 2^3$ be defined as

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

Clearly $s = 110$. Indeed, every output of f occurs twice, and the bitwise XOR of the corresponding inputs gives s .

Simon's problem, classically

The best one can do is to evaluate the function on random inputs and hope to find two distinct values with the same image, i.e., Compute f for sequence of values until finding a value x_j such that $f(x_j) = f(x_i)$ for a previous x_i , i.e. a **collision**. Then

$$x_j \oplus x_i = x_i \oplus (x_i \oplus s) = s$$

- Since f is **two-to-one**, after collecting 2^{n-1} evaluations with no collisions, the next evaluation must cause a collision.
- So in the **worst case** $2^{n-1} + 1$ evaluations are needed.

Simon's problem, classically

Suppose we made q queries to the oracle, resulting in a sequence of q -tuples $(x, f(x))$. The sequence contains

$$\frac{q(q-1)}{2}$$

possible pairs $((x_0, f(x_0)), (x_1, f(x_1)))$ and the probability that a randomly chosen pair has the same output is

$$\frac{1}{2^{n-1}}$$

and the probability of at least one such pair in the list is

$$\frac{q(q-1)}{2^n} \equiv \frac{q^2}{2^n}$$

which means that ideally the oracle should be queried around $q = \sqrt{2^n}$ times.

Simon's problem, classically

Or, more generically, how many evaluations do we need to have a collision **with probability p** ?

To have a collision with probability $p = \frac{1}{k} \leq \frac{1}{2}$ we need

$$\approx \sqrt{(2 \cdot 2^n) \cdot p} = \sqrt{\frac{2}{k} \cdot 2^n} = \sqrt{\frac{2}{k}} \cdot \sqrt{2^n} \text{ evaluations}$$



See the Birthday's problem

The problem **query complexity** is exponential on the input ...
Simon's algorithm, however, solves the problem in **polynomial time** with probability $\approx \frac{1}{4}$.

... thus, we are approaching an interesting point ...

Note: The birthday problem

Seeks to determine the probability that, in a set of n randomly chosen people, at least two will share a birthday.

$n = 23$ leads to $p(n) \approx 0.5$

Let the universe be $U = 365$ (days) and $n = 23$.

U^n is the space of birthdays and $V = \frac{U!}{(U-n)!}$ (n permutations of U) the number of birthdays with no repetitions.

Then,

$$p(n) = 1 - \frac{V}{U^n} \approx 1 - 0.493 \approx 0.507$$

Heuristic for cases leading with $p(n) \leq 0.5$

$$p(n) \approx \frac{n^2}{U} \Rightarrow n \approx \sqrt{2U * p(n)}$$

which yields for $p(n) = 0.5$, $n \approx 19$.

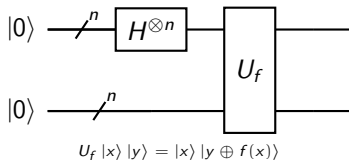
Simon's algorithm: The key steps

1. Prepare a superposition $\frac{1}{\sqrt{2}}(|x\rangle + |x \oplus s\rangle)$ for some string x
2. Use **interference** to find s (indeed, to extract a string y s.t. $y \cdot s = 0$)
3. Repeat previous steps **a sufficient number of** times to obtain system of equations in the form $y \cdot s = 0$
4. Solve the system for s using Gaussian elimination



Complexity n^3

Simon's algorithm: Preparing the superposition



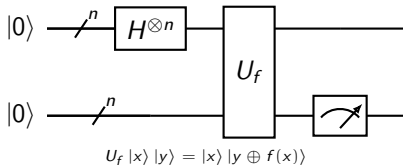
$$U_f(H^{\otimes n} \otimes I) |0\rangle |0\rangle = U_f\left(\frac{1}{\sqrt{2^n}} \sum_{x \in 2^n} |x\rangle |0\rangle\right) = \frac{1}{\sqrt{2^n}} \sum_{x \in 2^n} |x\rangle |f(x)\rangle$$

The state after the oracle can be rewritten as

$$\frac{1}{\sqrt{2^{n-1}}} \sum_{x \in P} \frac{1}{\sqrt{2}} (|x\rangle + |x \oplus s\rangle) |f(x)\rangle \quad (1)$$

Set P is composed of one representative of each of the 2^{n-1} sets of strings $\{x, x \oplus s\}$, into which 2^n can be partitioned.

Simon's Algorithm: Preparing the superposition



If the result of measuring the bottom qubits is $f(x)$, then the top ones will contain superposition

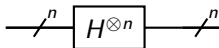
$$\frac{1}{\sqrt{2}}(|x\rangle + |x \oplus s\rangle)$$

as they are the unique values yielding $f(x)$.

i.e. a measurement of the bottom qubits chooses randomly one of the 2^{n-1} possible outcomes of f ...

as f gives the same output for x and $x \oplus s$, to 2^n possible inputs correspond 2^{n-1} possible outcomes.

Simon's Algorithm: Interference to find s



Recall

$$H|x\rangle = \frac{1}{\sqrt{2}} \sum_{z \in 2} (-1)^{xz} |z\rangle$$

which extends to a n -qubit as follows

$$\begin{aligned} H^{\otimes n}|x\rangle &= H|x_1\rangle H|x_2\rangle \cdots H|x_n\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{z \in 2^n} (-1)^{x \cdot z} |z\rangle \end{aligned}$$

where $x \cdot z$ denotes the bitwise product of x and z , modulo 2.

Simon's Algorithm: Interference to find s

Exercise 2 - Q 3.5

$$\begin{aligned} H^{\otimes n}|x\rangle &= H|x_1\rangle H|x_2\rangle \cdots H|x_n\rangle \\ &= \frac{1}{\sqrt{2}} \sum_{z_1 \in 2} (-1)^{x_1 z_1} |z_1\rangle + \frac{1}{\sqrt{2}} \sum_{z_2 \in 2} (-1)^{x_2 z_2} |z_2\rangle \cdots \frac{1}{\sqrt{2}} \sum_{z_n \in 2} (-1)^{x_n z_n} |z_n\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{z_1, z_2, \dots, z_n \in 2} (-1)^{x_1 z_1 + x_2 z_2 + \cdots + x_n z_n} |z_1 z_2 \cdots z_n\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{z \in 2^n} (-1)^{x \cdot z} |z\rangle \end{aligned}$$

Justify the last step.

Simon's Algorithm: Interference to find s

$$\begin{aligned}
& H^{\otimes n} \otimes I \left(\frac{1}{\sqrt{2}} (|x\rangle + |x \oplus s\rangle) |f(x)\rangle \right) \\
&= \frac{1}{\sqrt{2^n}} \sum_{z \in 2^n} \frac{1}{\sqrt{2}} ((-1)^{x \cdot z} + (-1)^{(x \oplus s) \cdot z}) |z\rangle |f(x)\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{z \in 2^n} \frac{1}{\sqrt{2}} ((-1)^{x \cdot z} + (-1)^{(x \cdot z) \oplus (s \cdot z)}) |z\rangle |f(x)\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{z \in 2^n} \frac{1}{\sqrt{2}} ((-1)^{x \cdot z} + (-1)^{(x \cdot z)} (-1)^{(s \cdot z)}) |z\rangle |f(x)\rangle \\
&= \frac{1}{\sqrt{2^{n+1}}} \sum_{z \in 2^n} \underbrace{(-1)^{x \cdot z} (1 + (-1)^{s \cdot z})}_{(*)} |z\rangle |f(x)\rangle
\end{aligned}$$

Simon's Algorithm: Interference to find s

$$\frac{1}{\sqrt{2^{n+1}}} \sum_{z \in 2^n} \underbrace{(-1)^{x \cdot z} (1 + (-1)^{s \cdot z})}_{(*)} |z\rangle |f(x)\rangle$$

- $s \cdot z = 1 \Rightarrow (*) = 0$ and the corresponding basis state $|z\rangle$ **vanishes**
- $s \cdot z = 0 \Rightarrow (*) \neq 0$: and the corresponding basis state $|z\rangle$ **is kept**.

In this case the probability of getting z upon measurement is $\frac{1}{2^{n-1}}$
(why?)

Simon's Algorithm: Interference to find s

Indeed, this state can be rewritten as follows:

$$\begin{aligned} & \frac{1}{\sqrt{2^{n+1}}} \sum_{z \in 2^n} (-1)^{x \cdot z} (1 + (-1)^{s \cdot z}) |z\rangle |f(x)\rangle \\ &= \frac{1}{\sqrt{2^{n+1}}} \sum_{z \in S^\perp} 2(-1)^{x \cdot z} |z\rangle |f(x)\rangle \\ &= \frac{1}{\sqrt{2^{n-1}}} \sum_{z \in S^\perp} (-1)^{x \cdot z} |z\rangle |f(x)\rangle \end{aligned}$$

where $S^\perp$, for $S = \{0, s\}$ is the **orthogonal complement** of subspace S ,
with $\dim(S^\perp) = n - 1$
(because $\dim(S) = 1$, as S is the subspace generated by s)

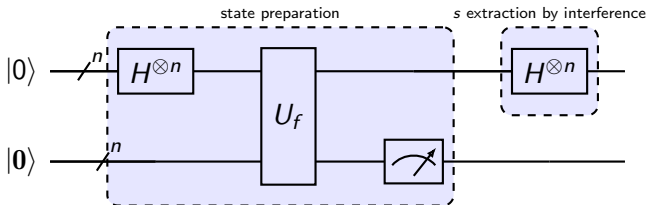
S and $S^\perp$

Both are subspaces of the vector space 2^n (often also referred as Z_n^2) whose vectors are **strings** of length n over $2 = \{0, 1\}$.

- The dimension of 2^n is n ; a basis is provided by strings with exactly one 1 in the k th position (for $k = 1, 2, \dots, n$).
- Two vectors v, u in 2^n are orthogonal iff $v \cdot u = 0$. Thus, a set of strings is **linearly independent** if no string in it can be expressed as the bitwise sum of other elements in the set.
- Thus, for any subspace F of 2^n , $F^\perp = \{u \in 2^n \mid \forall v \in F. u \cdot v = 0\}$

Warning: to not confuse with the Hilbert space in which the algorithm is executed and whose basis vectors are labeled by elements of 2^n .

Simon's algorithm: The circuit



Simon's Algorithm: Computing s

Running this circuit and measuring the control register results in some z in 2^n satisfying

$$s \cdot z = 0,$$

the distribution being uniform over all the strings that satisfy this constraint.

Question

Are we done?

Of course not:

This procedure needs to be repeated until $n - 1$ **linearly independent** such strings $\{z_1, z_2, \dots, z_{n-1}\}$ are found

Simon's Algorithm: Computing s

Then, it is enough to solve the following set of $n - 1$ equations in n unknowns:

$$z_1 \cdot s = 0$$

$$z_2 \cdot s = 0$$

$$\vdots$$

$$z_{n-1} \cdot s = 0$$

to determine s . Actually,

$\text{span}\{z_1, z_2, \dots, z_{n-1}\} = S^\perp$ and $\{z_1, z_2, \dots, z_{n-1}\}$ forms a **base** for $S^\perp$

Thus, s is the unique non-zero solution of

$$Zs = 0$$

where Z is the matrix whose line i corresponds to vector z_i .

Simon's Algorithm: Computing s

Question

What is the probability of obtaining such a system of equations by running the circuit $n-1$ times (i.e., not having to discard and run again)?

Simon's Algorithm: Probability of success

- Let $Y = \{y_1, \dots, y_k\}$ be a set of binary strings z linearly independents.
- Y spans a sub-space with 2^k elements with the general form

$$\bigoplus_{i=1..k} b_i y_i \quad \text{for each } b_i \in 2$$

- A new y obtained will be **independent** of the ones in Y iff it lives **out** of the subspace generated by Y which occurs with probability

$$1 - \frac{2^k}{2^n}$$

i.e. the probability of failure is $\frac{2^k}{2^n}$

Simon's algorithm: Probability of success

#	Probability of failure
1	$\frac{2^0}{2^{n-1}}$
2	$\frac{2^1}{2^{n-1}}$
3	$\frac{2^2}{2^{n-1}}$
...	...
$n-1$	$\frac{2^{n-2}}{2^{n-1}}$

This table yields the sequence of probabilities of failure,

$$\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots, \frac{1}{2^{n-1}} \quad (\text{from bottom to top})$$

Probability of failing in the first $n-2$ steps is thus

$$\frac{1}{4} + \frac{1}{8} + \dots = \frac{1}{4} \left(1 + \frac{1}{2} + \dots \right) \leq \frac{1}{4} \cdot \left(\sum_{i \in \mathbb{N}} \frac{1}{2^i} \right) = \frac{1}{2}$$



Geometric series whose sum is equal to two

Simon's algorithm: Probability of success

- Probability of succeeding in the first $n - 2$ steps at least $\frac{1}{2}$
- Probability of succeeding in the $(n - 1)$ -th step is $\frac{1}{2}$
- Thus probability of succeeding in all $n - 1$ steps at least $\frac{1}{4}$
- More advanced maths tells us that the probability is slightly higher (around 0.28878 ...)

Exponential separation

The period s of f can be computed with some constant probability of error after repeating Simon's algorithm $\mathcal{O}(n)$ times, which witnesses an **exponential separation** between classical and quantum computation.

The algorithm

1. Prepare the **initial state** $\frac{1}{\sqrt{2^n}} \sum_{x \in 2^n} |x\rangle |0\rangle$ and make $i := 1$
2. Apply the oracle U_f to obtain the state

$$\frac{1}{\sqrt{2^n}} \sum_{x \in 2^n} |x\rangle |f(x)\rangle$$

which can be re-written as

$$\frac{1}{\sqrt{2^{n-1}}} \sum_{x \in P} \frac{1}{\sqrt{2}} (|x\rangle + |x \oplus s\rangle) |f(x)\rangle$$

and **measure** the bottom qubits not strictly necessary but makes the analysis simpler.

3. Apply $H^{\otimes n}$ to the top qubits yielding a uniform superposition of elements of $S^\perp$.

The algorithm

4. Measure the first register and record the value observed z_i , which is a randomly selected element of $S^\perp$.
5. If the dimension of the span of $\{z_1, z_2, \dots, z_i\}$ is less than $n - 1$, increment i and to go step 2; else proceed.
6. Compute s as the unique non-zero solution of

$$Zs = 0$$

The crucial observation is that the set of observed values must form a basis to $S^\perp$.

The problem

The problem

Let $f : 2^n \longrightarrow X$, for some X finite, be such that,

$$f(x) = f(y) \text{ iff } x - y \in S$$

for some subspace S of Z_2^n with dimension m .

Find a basis $\{s_1, s_2, \dots, s_m\}$ for S .

In Simon's problem

- $x = y \oplus s$, i.e. $x - y = s$.
- s is a basis for the space S generated by $\{s\}$.

Note

The tuple $(2^n, \oplus, 0)$ forms a group with bitwise negation

Groups

A group (G, θ, u) is a set G with a binary operation θ which is associative, and equipped with an identity element u and an inverse:

$$a^{-1}\theta a = u = a\theta a^{-1}$$

Each set $\{x, x \oplus s\}$ in (1) is a coset of subgroup $S = (\{0, s\}, \oplus, 0)$

Coset

The coset of a subgroup S of a group (G, θ, u) wrt $g \in G$ is

$$gS = \{g\theta s \mid s \in S\}$$

In this case

$$xS = \{x \oplus 0, x \oplus s\} = \{x, x \oplus s\}$$

Generalised Simon's algorithm

If $S = \{0, y_1, \dots, y_{2^m-1}\}$ is a subspace of dimension m of 2^n , it can be decomposed into 2^{n-m} cosets of the form

$$\{x, x \oplus y_1, x \oplus y_2, \dots, x \oplus y_{2^m-1}\}$$

Then Step 2 yields

$$\begin{aligned} & \sum_{x \in 2^n} |x\rangle |f(x)\rangle \\ &= \frac{1}{\sqrt{2^{n-m}}} \sum_{x \in P} \frac{1}{\sqrt{2^m}} (|x\rangle + |x \oplus y_1\rangle + |x \oplus y_2\rangle + \dots + |x \oplus y_{2^m-1}\rangle) |f(x)\rangle \\ &= \frac{1}{\sqrt{2^{n-m}}} \sum_{x \in P} |x + S\rangle |f(x)\rangle \end{aligned}$$

where P be a subset of 2^n consisting of one representative of each 2^{n-m} disjoint cosets, and

$$|x + S\rangle = \sum_{s \in S} \frac{1}{\sqrt{2^m}} |s\rangle$$

Generalised Simon's algorithm

- In step 4 the first register is left in a state of the form $|x + S\rangle$ for a random x .
- After applying the Hadamard transformation, the first register contains a uniform superposition of elements of $S^\perp$ and its measurement yields a value sampled uniformly at random from $S^\perp$.

This leads to the revised algorithm:

5. If the dimension of the span of $\{z_1, z_2, \dots, z_i\}$ is less than $n - m$, increment i and to go step 2; else proceed.
6. Compute the system of linear equations

$$Zs = 0$$

and let $s_1, s_2, \dots, s_m$ be the generators of the solution space. They form the envisaged basis.

The hidden subgroup problem

The group S is often called the **hidden subgroup**.

The (generalised) Simon's algorithm is an instance of a much general scheme, leading to exponential advantage, known as

The hidden subgroup problem

Let (G, θ, u) be a group and $f : G \longrightarrow X$ for some finite set X with the following property:

f is constant on cosets of S and distinct on different cosets

i.e.

there is a subgroup S of G such that for any $x, y \in G$,

$$f(x) = f(y) \text{ iff } x\theta S = y\theta S$$

Characterise S .