

Lecture 4:

Algorithms: Phase kick-back

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



Universidade do Minho



INESC TEC



INTERNATIONAL IBERIAN
NANOTECHNOLOGY
LABORATORY



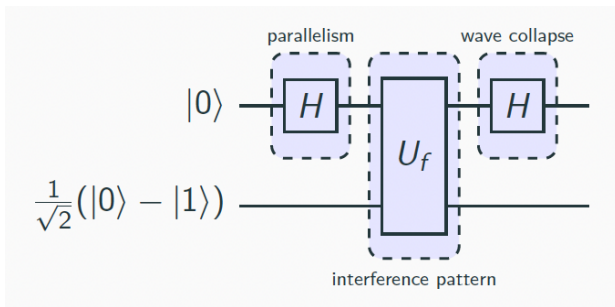
**UNU
EGOV**

Quantum Computation

Mestrado em Engenharia Física

Universidade do Minho, 2026-27

Revisiting the Deutsch algorithm



Basic ingredients

- Input in **superposition**
- An **oracle** for f taking the form of a **controlled gate** on the input
- A specific **preparation of the first qubit**

Revisiting the Deutsch algorithm

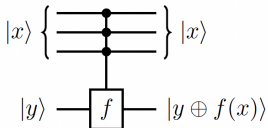
The **oracle** for the Deutsch algorithm

$$|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$$

takes the form of a **generalised cX** gate:

$$\sum_{x \in \{0,1\}^n} |x\rangle\langle x| \otimes X^{f(x)}$$

where $X^{f(x)}$ is the identity I (when $f(x) = 0$) or X (when $f(x) = 1$).



Going even simpler: cX as an oracle



$$\overbrace{\begin{bmatrix} I & 0 \\ 0 & X \end{bmatrix}}^{cX}$$

corresponds to the oracle: $|xy\rangle \mapsto |x, x \oplus y\rangle$

$$cX|0\rangle|\varphi\rangle = |0\rangle I|\varphi\rangle$$

$$cX|1\rangle|\varphi\rangle = |1\rangle X|\varphi\rangle$$

Seen as an oracle, note that **input** is presented at the **control** qubit and **output** is produced on the **target** qubit.

Going even simpler: cX as an oracle

Consider now a special case: prepare the target qubit with $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$ which is an **eigenvector** of both

- X (with $\lambda = -1$) and of I (with $\lambda = 1$)
- and, thus, $X \frac{|0\rangle - |1\rangle}{\sqrt{2}} = -1 \frac{|0\rangle - |1\rangle}{\sqrt{2}}$ and $I \frac{|0\rangle - |1\rangle}{\sqrt{2}} = 1 \frac{|0\rangle - |1\rangle}{\sqrt{2}}$

Therefore,

$$\begin{aligned} cX |1\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) &= |1\rangle \left(X \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \right) \\ &= |1\rangle \left((-1) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \right) \\ &= -|1\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \end{aligned}$$

$$\text{while } cX |0\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) = |0\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

Going even simpler: cX as an oracle

A phase (1 or -1 , i.e., a eigenvalue)

jumps, or is kicked back

from the **second** (target) to the **first** (control) qubit where the input is presented.

This effect is suitably recorded in the following formulation of cX :

$$cX|b\rangle|-\rangle = (-1)^b|b\rangle|-\rangle \quad \text{with } b \in \mathbf{2}$$

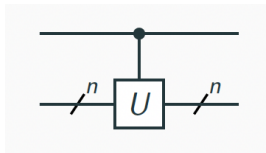
Observe now that, through the kick-back effect, **ouput** arises in the **control** qubit, whereas the **target** qubit remains unchanged.

Example:

$$cX \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) = \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

The phase kick-back pattern

This can be generalised to every **controlled** quantum operation:



Let v be an eigenvector of U (i.e. $Uv = e^{i\theta} v$). Thus,

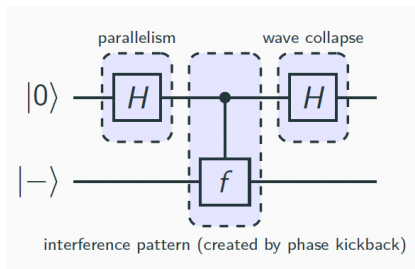
$$\begin{aligned} & cU((\alpha|0\rangle + \beta|1\rangle) \otimes v) \\ &= cU(\alpha|0\rangle \otimes v + \beta|1\rangle \otimes v) \\ &= \alpha|0\rangle \otimes v + \beta|1\rangle \otimes Uv \\ &= \alpha|0\rangle \otimes v + \beta|1\rangle \otimes e^{i\theta} v \\ &= \alpha|0\rangle \otimes v + e^{i\theta} \beta|1\rangle \otimes v \\ &= (\alpha|0\rangle + e^{i\theta} \beta|1\rangle) \otimes v \end{aligned}$$

Our two examples

Phase kick-back can be represented as

in the cX gate: $cX|b\rangle|-\rangle = (-1)^b|b\rangle|-\rangle$

in Deutsch algorithm: $U_f|x\rangle|-\rangle = (-1)^{f(x)}|x\rangle|-\rangle$



A parenthesis on global/local phase

$$\left(\begin{array}{ccc} & & \\ & & \\ & & \end{array} \right)$$

Global phase factor

Definition

Let $|v\rangle, |u\rangle \in \mathbb{C}^{2^n}$. If $|v\rangle = e^{i\theta}|u\rangle$ we say they are equal up to **global phase factor** $e^{i\theta}$

Theorem

$e^{i\theta}|v\rangle$ and $|v\rangle$ are *indistinguishable* in the world of quantum mechanics

Proof sketch

Consider a phase factor $e^{i\theta}$ affecting two states $|u\rangle$ and $|v\rangle$. Clearly, the internal product of $e^{i\theta}|u\rangle$ and $e^{i\theta}|v\rangle$ boils down to

$$\langle u|e^{-i\theta}e^{i\theta}|v\rangle = \langle u|v\rangle$$

Note that $|0\rangle + |1\rangle$ and $|0\rangle - |1\rangle$ differ by a **relative** phase factor. They are **distinguishable**!

End of parenthesis

...)

The Bernstein-Vazirani algorithm

Let $2^n = \{0, 1\}^n = \{0, 1, 2, \dots, 2^n - 1\}$ be the set of non-negative integers (represented as bit strings up to n bits). Then, consider the following problem:

The problem

Let s be an unknown non-negative integer less than 2^n , encoded as a bit string, and consider a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ which hides secret s as follows: $f(x) = x \cdot s$, where $\cdot$ is the bitwise product of x and s modulo 2. i.e.

$$x \cdot s = x_1 s_1 \oplus x_2 s_2 \oplus \dots \oplus x_n s_n$$

Find s .

Note that juxtaposition abbreviates conjunction, i.e. $x_1 s_1 = x_1 \wedge s_1$

Setting the stage

Lemma

(1) For $a, b \in 2$

$$(-1)^a (-1)^b = (-1)^{a \oplus b}$$

Proof sketch

Build a truth table for each case and compare the corresponding contents.

Lemma

(2) For any three binary strings $x, a, b \in 2^n$

$$(x \cdot a) \oplus (x \cdot b) = x \cdot (a \oplus b)$$

Proof sketch

Follows from the fact that for any three bits $a, b, c \in 2$ the equation $(a \wedge b) \oplus (a \wedge c) = a \wedge (b \oplus c)$ holds.

Setting the stage

Lemma

(3) For any element $|b\rangle$ in the computational basis of $\mathbb{C}^2$,

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

Proof sketch

Build a truth table and compare the corresponding contents.

Theorem

(1) For any element $|b\rangle$ in the computational basis of $\mathbb{C}^{2^n}$,

$$H^{\otimes n}|b\rangle = \frac{1}{\sqrt{2^n}} \sum_{z \in 2^n} (-1)^{b \cdot z} |z\rangle$$

Proof sketch

Follows by induction on the size of n .

The Bernstein-Vazirani algorithm

How many times f has to be called to determine s ?

- Classically, we run f n -times by computing

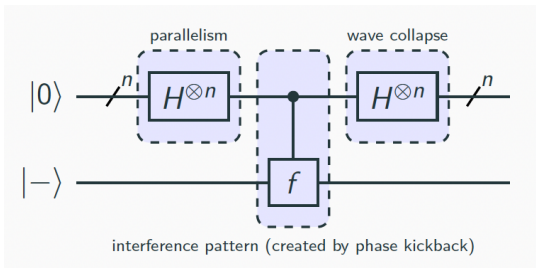
$$f(1 \dots 0) = (s_1 \wedge 1) \oplus \dots \oplus (s_n \wedge 0) = s_1$$

$$\vdots$$

$$f(0 \dots 1) = (s_1 \wedge 0) \oplus \dots \oplus (s_n \wedge 1) = s_n$$

- With a quantum algorithm, we may discover s by running f only once

The circuit



Why?

First observe that

$$\frac{1}{2^n} \sum_{z \in 2^n} (-1)^{z \cdot w} = \begin{cases} 1 & \text{if } w = 0 \\ 0 & \text{if } w \neq 0 \end{cases}$$

The reason is easy to guess:

- for $w = 0$, $\frac{1}{2^n} \sum_{z=0}^{2^n-1} (-1)^{z \cdot w} = \frac{1}{2^n} \sum_{z=0}^{2^n-1} 1 = 1$.
- for $w \neq 0$, as z spans all numbers from 0 to $2^n - 1$, half of the 2^n factors in the sum will be -1 and the other half 1 , thus summing up to 0.

Why?

$$\dots = \frac{1}{2^n} \sum_{z \in 2^n} \sum_{z' \in 2^n} (-1)^{z \cdot (s \oplus z')} |z'\rangle |-\rangle = \dots$$

In our case

- $w = s \oplus z'$, for each z'
- and $(s \oplus z') = 0$ iff $z' = s$

Thus, the only non-zero amplitude is the one associated with s .

Why?

Alternatively, consider the probability of measuring s at the end of the computation:

$$\begin{aligned} & \left| \frac{1}{2^n} \sum_{z \in 2^n} (-1)^{z \cdot (s \oplus s)} \right|^2 \\ &= \left| \frac{1}{2^n} \sum_{z \in 2^n} (-1)^{z \cdot 0} \right|^2 \\ &= \left| \frac{1}{2^n} \sum_{z \in 2^n} 1 \right|^2 \\ &= \left| \frac{2^n}{2^n} \right|^2 \\ &= 1 \end{aligned}$$

This means that somehow all values yielding wrong answers were completely **cancelled**.

Deutsch-Josza

The Problem

Take a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, which is known to be either constant or balanced.

Find out which case holds.

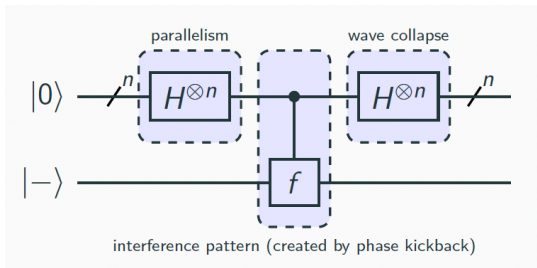
Classically, we evaluate half of the inputs ($\frac{2^n}{2} = 2^{n-1}$), evaluate one more and run the decision procedure,

- output always the same $\implies$ constant
- otherwise $\implies$ balanced

which requires running f $2^{n-1} + 1$ times.

A quantum algorithm replies by running f only once.

The circuit



The computation

$$\begin{aligned}
 & H^{\otimes n} |0\rangle |-\rangle \\
 &= \frac{1}{\sqrt{2^n}} \sum_{z \in 2^n} |z\rangle |-\rangle && \{\text{Theorem 1}\} \\
 &\xrightarrow{U_f} \frac{1}{\sqrt{2^n}} \sum_{z \in 2^n} (-1)^{f(z)} |z\rangle |-\rangle && \{\text{Definition}\} \\
 &\xrightarrow{H^{\otimes n} \otimes I} \underbrace{\frac{1}{2^n} \sum_{z \in 2^n} (-1)^{f(z)} \left(\sum_{z' \in 2^n} (-1)^{z \cdot z'} |z'\rangle \right)}_{\square \text{ upper qubits}} |-\rangle && \{\text{Theorem 1}\}
 \end{aligned}$$

Developing □ by case distinction

f is constant

$$\begin{aligned} & \frac{1}{2^n} \sum_{z \in 2^n} (-1)^{f(z)} \left(\sum_{z' \in 2^n} (-1)^{z \cdot z'} |z'\rangle \right) \\ &= \frac{1}{2^n} (\pm 1) \sum_{z \in 2^n} \left(\sum_{z' \in 2^n} (-1)^{z \cdot z'} |z'\rangle \right) \end{aligned}$$

Therefore, at state $|0\rangle$ is

$$\boxed{f \text{ is constant at } 1} \rightsquigarrow \frac{-(2^n)}{2^n} |0\rangle = -|0\rangle$$

$$\boxed{f \text{ is constant at } 0} \rightsquigarrow \frac{(2^n)}{2^n} |0\rangle = |0\rangle$$

Developing □ by case distinction

Actually the probability of measuring $|0\rangle$ at the end given by

$$\begin{aligned} & \left| \frac{1}{2^n} (\pm 1) \sum_{z \in 2^n} (-1)^{z \cdot 0} \right|^2 \\ &= \left| \frac{1}{2^n} (\pm 1) \sum_{z \in 2^n} 1 \right|^2 \\ &= \left| \frac{2^n}{2^n} \right|^2 \\ &= 1 \end{aligned}$$

So if f is constant we measure $|0\rangle$ with probability 1.

Developing $\square$ by case distinction f is balanced

$$\begin{aligned} & \frac{1}{2^n} \sum_{z \in 2^n} (-1)^{f(z)} \left(\sum_{z' \in 2^n} (-1)^{z \cdot z'} |z'\rangle \right) \\ &= \frac{1}{2^n} \left(\sum_{z \in 2^n, f(z)=0} (-1)^{f(z)} \left(\sum_{z' \in 2^n} (-1)^{z \cdot z'} |z'\rangle \right) \right. \\ & \quad \left. + \sum_{z \in 2^n, f(z)=1} (-1)^{f(z)} \left(\sum_{z' \in 2^n} (-1)^{z \cdot z'} |z'\rangle \right) \right) \\ &= \frac{1}{2^n} \left(\sum_{z \in 2^n, f(z)=0} \left(\sum_{z' \in 2^n} (-1)^{z \cdot z'} |z'\rangle \right) \right. \\ & \quad \left. + \sum_{z \in 2^n, f(z)=1} (-1) \left(\sum_{z' \in 2^n} (-1)^{z \cdot z'} |z'\rangle \right) \right) \end{aligned}$$

Developing □ by case distinction

Probability of measuring $|0\rangle$ at the end given by

$$\begin{aligned} & \left| \frac{1}{2^n} \left(\sum_{z \in 2^n, f(z)=0} (-1)^{z \cdot 0} + \sum_{z \in 2^n, f(z)=1} (-1)(-1)^{z \cdot 0} \right) \right|^2 \\ &= \left| \frac{1}{2^n} \left(\sum_{z \in 2^n, f(z)=0} 1 + \sum_{z \in 2^n, f(z)=1} (-1) \right) \right|^2 \\ &= \left| \frac{1}{2^n} \left(\sum_{z \in 2^n, f(z)=0} 1 - \sum_{z \in 2^n, f(z)=1} 1 \right) \right|^2 \\ &= 0 \end{aligned}$$

So if f is balanced we measure $|0\rangle$ with probability 0

Concluding

Deutsch problem

Classically, need to run f **twice**. With a quantum algorithm **once** is enough.

Berstein-Varziani problem

Classically, need to run f **n** times. With a quantum algorithm **once** is enough.

Deutsch-Josza problem

Classically, need to evaluate half of the inputs ($\frac{2^n}{2} = 2^{n-1}$), evaluate one more and run the decision procedure,

- output always the same $\implies$ constant
- otherwise $\implies$ balanced

Resorting to a quantum algorithm, **once** is enough.