A - III - Quantum Algorithms.

In this part we will start exploring the power and potential applications of QC.

There are 3 approaches that we can pursue to study the differences btw the capabilities of classical and quantum computers.

1) Exponential speedup for "apparently" hard pbs :

efficient quantum algo that solve pbs we don't know how to solve efficiently with a classical computer -

("efficient" means that the pb can be solved in poly time).

ex: Shor's algorithm for solving the factoring pb which is strongly suspected (although not proved) to not be in BPP.

2) Polynomial speed up

There are quantum algo that can solve pbs faster than any classical algo, but not exponentially faster.

=> these shed no light on the classification of complexity, but they are nonetheless useful for practical applications.

ex: Grover's search algorithm.

3) "Relativized" exponential speed up
↳ Quantum black box model

Model in which we have a "black box" (also called "oracle") that performs an a priori unknown operation and the task is to find out what this black box does.

It is possible to show that there are quantum black boxes that exist with the following property:

By feeding quantum superpositions to the box, we can learn what is inside the box with an exponential speed up compared to the classical case (if we were only allowed classical input states).

↳ " $BPP^{O} \neq BQP^{O}$ "

↳ BQP "relative to the oracle".

Example: Simon's pb

Even though such black box pbs do not necessarily have practical applications, they are good " demonstration " models.

→ We will start by discussing such black box pbs.

* Complexity of "black box" models

(= "oracle" model = "query" model)

The box computes an a priori unknown function $f(x)$.

$$f: \{0,1\}^n \longrightarrow \{0,1\}$$
$$x \longmapsto f(x)$$

f may not be invertible but we saw previously that this can be computed by a quantum black box that applies a unitary transfo U_f :

$$U_f : |x\rangle \otimes |y\rangle \longmapsto |x\rangle \otimes |y \oplus f(x)\rangle$$

($\oplus$: addition "modulo 2")

we may be given some "promise" about f
(i.e. that it is has a certain structure...)
and our task is to find f or some prop. of f

The way we define complexity in such model is by evaluating the number of times we have to evaluate the fct in order to solve pb.

Query complexity = minimal number of evaluations of $f(x)$ ("queries") needed to solve the pb.

→ How does it scale with n ?

here we do not worry about the complexity of the task inside the box.

→ still useful because the # of queries will be a lower bound on the gate complexity (at least one gate in the box)

We will distinguish between

"classical queries" → inputs are n-bit strings (can be done by querying the quantum box with computational basis states).

"quantum queries" → inputs are superpositions of computational-basis states.

⇒ can the ability of querying with superpositions states give us additional computational power ?

* Deutsch's Problem (1985) $(n = m = 1)$

↳ demonstration of how quantum parallelism and interference can achieve quantum speed up

$$f : \{0,1\} \longrightarrow \{0,1\}$$
$$x \longmapsto f(x)$$

⇒ The quantum black box does:

$$U_f : |x\rangle \otimes |y\rangle \longmapsto |x\rangle \otimes |y \oplus f(x)\rangle$$

($|x\rangle$: 1 qubit, $|y\rangle$: 1 qubit)

Two classical queries completely characterize the fct.
But suppose we only want to know whether the fct is constant or balanced.

↳ $f(0) = f(1)$ ↳ $f(0) \neq f(1)$

Classically, we still need two queries.
(the knowledge of $f(0)$ or $f(1)$ alone is not enough to tell).

But if we can query in <u>superposition</u>, one query is enough:

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

$$U_f : |x\rangle \otimes |-\rangle \longmapsto |x\rangle \otimes |- \oplus f(x)\rangle$$

$$= |x\rangle \otimes \frac{1}{\sqrt{2}}\left(|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle\right)$$

$$= |x\rangle \otimes \frac{1}{\sqrt{2}}\left(\underbrace{|f(x)\rangle - |1 \oplus f(x)\rangle}\right)$$

$$\begin{cases} |0\rangle - |1\rangle & \text{if } f(x)=0 \\ |1\rangle - |0\rangle & \text{if } f(x)=1 \end{cases}$$

$$= |x\rangle \otimes (-)^{f(x)} \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$$

$$= (-)^{f(x)} \; |x\rangle \otimes |-\rangle$$

$\Rightarrow$ The function $f(x)$ has been isolated in a phase and the right register $(|-\rangle)$ is actually unmodified.

Effectively the box does $|x\rangle \longmapsto (-)^{f(x)} |x\rangle$

If we query with $|x\rangle = |0\rangle$ or $|1\rangle$, we get the classical case and just a global phase. But in the quantum case we can query with $|x\rangle$ being a superposition state.

2) Take $|x\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

This state is transformed as:

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \longmapsto \frac{1}{\sqrt{2}}\underbrace{\left((-)^{f(0)}|0\rangle + (-)^{f(1)}|1\rangle\right)}$$

$\propto |+\rangle$ if $f(0) = f(1)$

$\propto |-\rangle$ if $f(0) \neq f(1)$

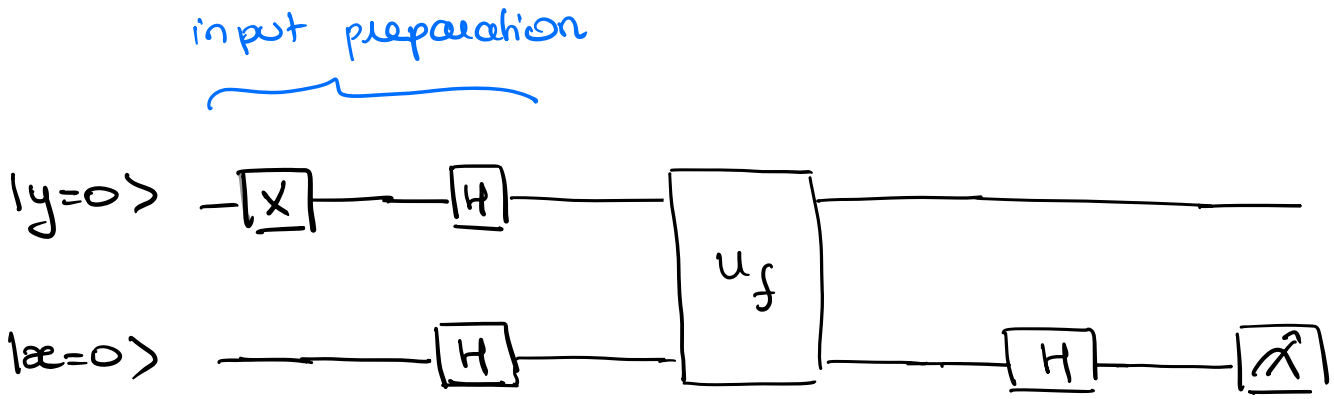
$\Rightarrow$ if we now measure in the basis $|\pm\rangle$ the outcome will be $\begin{cases} |+\rangle & \text{if } f \text{ is constant} \\ |-\rangle & \text{if } f \text{ is balanced} \end{cases}$

In practice the measurement in the $|\pm\rangle$ basis is done by first applying the Hadamard gate and then measuring in the computational $\{|0\rangle, |1\rangle\}$ basis:

$$\frac{1}{\sqrt{2}}\left((-)^{f(0)}|0\rangle + (-)^{f(1)}|1\rangle\right) \xrightarrow{H} \begin{cases} \propto |0\rangle & \text{if } f(0)=f(1) \\ \propto |1\rangle & \text{if } f(0)\neq f(1) \end{cases}$$

$\Rightarrow$ measurement in the $\{|0\rangle, |1\rangle\}$ basis yields outcome $|0\rangle$ if f is constant, $|1\rangle$ if f is balanced.

Corresponding circuit:



$$|xy\rangle = |00\rangle \xrightarrow{X} |01\rangle \xrightarrow{H\otimes H} |+-\rangle \xrightarrow{U_f} |x_{out}\rangle|-\rangle \xrightarrow{H} \begin{cases} |0\rangle|-\rangle & \text{if } f \text{ cst} \\ |1\rangle|-\rangle & \text{if } f \text{ balanced} \end{cases}$$

$$|x_{out}\rangle \equiv \frac{1}{\sqrt{2}}\left((-)^{f(0)}|0\rangle + (-)^{f(1)}|1\rangle\right)$$

Because we can act on a superposition of $|0\rangle$ and $|1\rangle$, the quantum computer allows us to determine a global property of $f(x)$ (i.e. property that depends on both $f(0)$ and $f(1)$) using only one query.

→ This is quantum parallelism.

* Deutsch-Josza Algorithm

↳ generalization of the Deutsch algo for arbitrary input size n.
(will allow us to learn about complexity)

$$f: \{0,1\}^n \longrightarrow \{0,1\}$$
$$x \longmapsto f(x)$$

($x \equiv x_{n-1} 2^{n-1} + \dots + x_1 2 + x_0$ is now an n-bit string)

and we are promised that f is either

constant → $f(x) \overset{=c}{}$ is the same for all 2^n inputs.

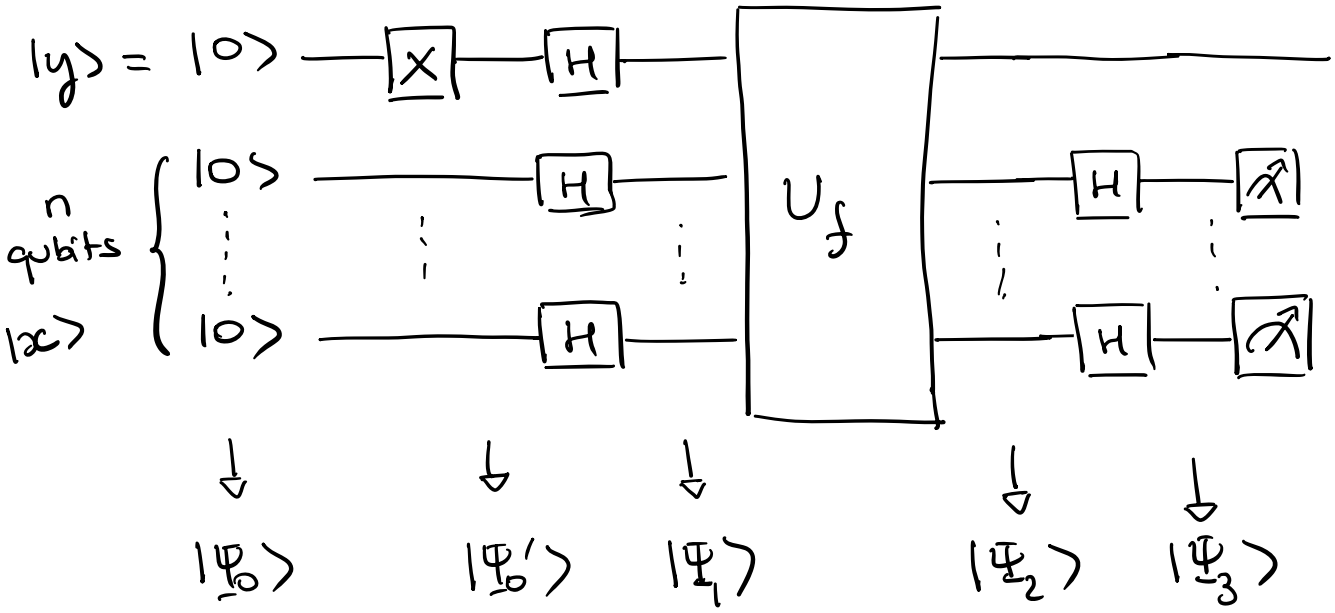
or balanced → $\begin{cases} f(x) = 0 & \text{for half } (2^{n-1}) \text{ of the inputs } x. \\ f(x) = 1 & \text{for the other half.} \end{cases}$

The pb is again to determine whether f is constant or balanced, and in fact this can also be done with only one query of our quantum algo.

$$U_f : |x\rangle \otimes |y\rangle \longmapsto |x\rangle \otimes |y \oplus f(x)\rangle$$

$|x\rangle$: n qubits; $|y\rangle$: 1 qubit

Now the circuit is:



- $|\Phi_0\rangle = \underbrace{|00\ldots0\rangle}_{|x\rangle} \otimes \underbrace{|0\rangle}_{|y\rangle} \equiv |0\rangle^{\otimes n} \otimes |0\rangle$

- $|\Psi_0'\rangle = |0\rangle^{\otimes n} \otimes |1\rangle$

- what is the effect of applying n Hadamard gates in parallel to n qubits in comp. basis states?

$$\underbrace{(H \otimes H \otimes \cdots \otimes H)}_{H^{\otimes n}} \underbrace{|x\rangle}_{\equiv |x_{n-1} x_{n-2} \cdots x_1 x_0\rangle \quad (x_i = 0,1)} = ?$$

→ $\underline{n=1}$: $H|x_0\rangle = \dfrac{|0\rangle + (-)^{x_0}|1\rangle}{\sqrt{2}}$

→ $\underline{n=2}$: $|x\rangle \equiv |x_1 x_0\rangle$

$$(H \otimes H)|x_1 x_0\rangle = H|x_1\rangle \otimes H|x_0\rangle$$

$$= \frac{1}{2}\left(|0\rangle + (-)^{x_0}|1\rangle\right)\left(|0\rangle + (-)^{x_1}|1\rangle\right)$$

$$= \frac{1}{2}\left(|00\rangle + (-)^{x_0}|10\rangle + (-)^{x_1}|01\rangle + (-)^{x_0+x_1}|11\rangle\right)$$

$$= \frac{1}{2} \sum_{\substack{z_0=0,1 \\ z_1=0,1}} (-)^{x_0 z_0 + x_1 z_1} |z_0 z_1\rangle$$

→ in general :

$$H^{\otimes n} |x_{n-1} \cdots x_1 x_0\rangle = \frac{1}{2^{n/2}} \sum_{\substack{z_{n-1} \\ \vdots \\ z_0}} (-)^{x_0 z_0 + \cdots x_{n-1} z_{n-1}} |z_{n-1} \cdots z_1 z_0\rangle$$

$$\equiv \frac{1}{2^{n/2}} \sum_{z=0}^{2^n-1} (-)^{x \cdot z} |z\rangle \qquad (*)$$

The bottom register is in state $|0\rangle^{\otimes n} = |00\ldots0\rangle$ before $H^{\otimes n}$ and becomes $\frac{1}{2^{n/2}} \sum_{z=0}^{2^n-1} |z\rangle$ after.

⇒ The state $|\Psi_1\rangle$ after action of all the Hadamard gates $H^{\otimes n} \otimes H$ is :

$$|\Psi_1\rangle = \frac{1}{2^{n/2}} \sum_{z=0}^{2^n-1} |z\rangle \otimes |-\rangle .$$

- Now we act with U_f :

$$|\Psi_2\rangle = U_f |\Psi_1\rangle$$

$$= \frac{1}{2^{n/2}} \sum_{z=0}^{2^n-1} |z\rangle \otimes |- \oplus f(z)\rangle$$

$$= \frac{1}{2^{n/2}} \sum_{z=0}^{2^n-1} |z\rangle \otimes \frac{1}{\sqrt{2}} \left(|0 \oplus f(z)\rangle - |1 \oplus f(z)\rangle \right)$$

$$= \frac{1}{2^{n/2}} \sum_{z=0}^{2^n-1} (-)^{f(z)} |z\rangle \otimes |-\rangle .$$

- Now we act again with $H^{\otimes n}$ on the bottom register:

$$|\Psi_z\rangle = \frac{1}{2^{n/2}} \sum_{z=0}^{2^n-1} (-)^{f(z)} H^{\otimes n} |z\rangle \otimes |-\rangle$$

(*)

$$= \frac{1}{2^n} \sum_{z=0}^{2^n-1} (-)^{f(z)} \sum_{z'=0}^{2^n-1} (-)^{z \cdot z'} |z'\rangle \otimes |-\rangle$$

$$= \sum_{z'=0}^{2^n-1} \left[\frac{1}{2^n} \sum_{z=0}^{2^n-1} (-)^{f(z)} (-)^{z \cdot z'} \right] |z'\rangle \otimes |-\rangle$$

$$\equiv S(z')$$

* if f is constant $\Rightarrow f(z) = c \quad \forall z$

$$\Rightarrow \quad S = \frac{(-)^c}{2^n} \sum_{z=0}^{2^n-1} (-)^{z \cdot z'} = (-)^c \, \delta_{z'_{n-1},0} \cdots \delta_{z'_1,0} \delta_{z'_0,0}$$

$z \cdot z' = z_{n-1} z'_{n-1} + z_{n-2} z'_{n-2} + \cdots z_1 z'_1 + z_0 z'_0$

↳ if $z' = 0$ ie $z'_{n-1} = z'_{n-2} = \cdots = z'_1 = z'_0 = 0$

$\Rightarrow z \cdot z' = 0 \quad \Rightarrow \sum_z (-)^{z z'} = 2^n$

↳ otherwise the terms will cancel

ex: $z'_0 = 1 \quad z'_{n-1} = \cdots = z'_1 = 0$

$$\Rightarrow \sum_z (-)^{z \cdot z'} = \sum_{z_{n-1} \cdots z_1} \underbrace{\sum_{z_0 = 0,1} (-)^{z_0}}_{1-1=0}$$

$$\Rightarrow |\Psi_z\rangle = \sum_{z=0}^{2^n-1} (-)^c \, \delta_{z',0} \, |z'\rangle \otimes |-\rangle$$

$$= (-)^c \, |0\rangle^{\otimes n} \otimes |-\rangle$$

$\Rightarrow$ When we measure the bottom register, the outcome will <u>always</u> be $|z'\rangle = |0\rangle^{\otimes n}$.

* if f is balanced $\Rightarrow$ $\begin{cases} f(z) = 0 \text{ for } 2^{n-1} \text{ of the inputs} \\ f(z) = 1 \text{ for the other } 2^{n-1} \text{ inputs} \end{cases}$

$\Rightarrow$ for $|z'\rangle = |0\rangle^{\otimes n}$, the sum is

$$S' = \frac{1}{2^n} \sum_{z=0}^{2^n-1} (-)^{f(z)} (-)^{z \cdot z'} = \frac{1}{2^n} \sum_{z=0}^{2^n-1} (-)^{f(z)} = 0.$$

$\Rightarrow$ The bottom register is never in state $|z'\rangle = |0\rangle^{\otimes n}$ $\Rightarrow$ the proba of obtaining this state as outcome of the measurement is zero.

Note : another way to see this is to look directly at the proba of outcome $|z'\rangle = |00\ldots0\rangle$ which is:

$$|S(z'=00\ldots0)|^2 = \left| \frac{1}{2^n} \underbrace{\sum_{z=0}^{2^n-1} (-)^{f(z)}}_{\begin{cases} 2^n & \text{if } f(z)=0 \ \forall z \\ -2^n & \text{if } f(z)=1 \ \forall z \\ 0 & \text{if } f \text{ balanced} \end{cases}} \right|^2$$

$$\Rightarrow |S(z'=000)|^2 = \begin{cases} 1 & \text{if } f \text{ cst} \\ 0 & \text{if } f \text{ balanced} \end{cases}$$

In conclusion, one query of the quantum oracle is sufficient to tell whether f is constant or balanced :

outcome $|z'\rangle = |0\rangle^{\otimes n}$ means f is constant

any other outcome means f is balanced.

→ the pb is in the class BQP^O ↳ "relative to the oracle"

What about classically ?

To tell whether f is balanced or constant we would have to compute $f(x)$ for many inputs x (keeping the previous one in memory).

If at some point we have $f(x) \neq f(x_{previous})$ then we know f is balanced.

But if f is constant, we won't know for sure until we have tried more than half of the input, this means at least :

$$\frac{2^n}{2} + 1 = 2^{n-1} + 1 \quad \text{queries.}$$

↳ which is exponential in n.

⇒ If we have a deterministic classical comp, the quantum computer presents exponential "speed-up" over the classical one.

→ the pb is not in P°.

⇒ BQP° is strictly larger than P°.

But if we use a randomized classical computer and allow for an error $\varepsilon < 1/2$:

* 1 query $f(x) \rightarrow$ can only guess $\rightarrow \varepsilon = 1/2$.

* 2 queries $f(x)$, $f(x')$
  - if $f(x) \neq f(x') \rightarrow$ know it is balanced
  - if $f(x) = f(x') \rightarrow$ guess it is constant — what's the error

proba that get twice same outcome when f is balanced = ($\#$ of x' with $f(x') = f(x)$, $x' \neq x$)

$$\frac{1}{2} \cdot \frac{2^{n-1} - 1}{2^n - 1} < 1/2$$

$\frac{1}{2}$: proba for 1st outcome.

$\frac{2^{n-1}-1}{2^n-1}$: proba that 2nd outcome is the same ($\rightarrow$ total $\#$ of x' (with $x' \neq x$.))

* in fact the error gets exponentially small as we increases the $\#$ of queries :

for k queries, if $k \ll 2^n$

$$\varepsilon \sim \left(\frac{1}{2}\right)^k = 2^{-k}$$

- for given ε → need $k \sim \log(1/\varepsilon)$ queries.

even if we want expo small error $\varepsilon \sim 2^{-n}$

$\Rightarrow$ $O(n)$ queries

$\Rightarrow$ in fact the pb is in BPP°, i.e. can be solved efficiently with a randomized CC.

$\Rightarrow$ doesn't show that BQP° is strictly larger than BPP°.

* Simon's Algorithm

The Deutsch-Josza algo showed an exponent. quantum improvement over the best deterministic classical algorithm, but not over the best randomized classical one.

The Simon's pb is a computational pb for which quantum computer can be exponentially more efficient than randomized CC.

(meaning BQP^O is strictly larger than BPP^O where "O" refers to the oracle of the Simon's pb)

This pb actually inspired Shor's factoring algorithm (see later).

-> Simon's pb:

Now the quantum black box computes a fct:

$$f : \{0,1\}^n \longrightarrow \{0,1\}^n$$
$$x \longmapsto f(x)$$

=> corresponding unitary:

(both n-qubit states)

$$U_f : |x\rangle \otimes |y\rangle \longmapsto |x\rangle \otimes |y \oplus f(x)\rangle$$

and we are assured that f is a 2-to-1 function with "period" a (which is an n-bit string), meaning:

$$f(x) = f(x') \iff x' = x \oplus a$$

$\oplus$ = addition modulo 2 (XOR gate)

Since x, y, a are n-bit strings:

$$\begin{cases} x \equiv x_{n-1}\, 2^{n-1} + \dots + x_1\, 2 + x_0 \\ x' \equiv x'_{n-1}\, 2^{n-1} + \dots + x'_1 2 + x_0' \\ \\ a \equiv a_{n-1}\, 2^{n-1} + \dots + a_1\, 2 + a_0 \end{cases}$$

$x' = x \oplus a$ means:

$$\begin{cases} x'_{n-1} = x_{n-1} \oplus a_{n-1} \\ \vdots \\ x'_1 = x_1 \oplus a_1 \\ x'_0 = x_0 \oplus a_0 . \end{cases}$$

This is all we know about f -
The task is to find a.

→ How hard is this pb classically ?

* Simon's pb is in NP^O : if someone gives us the period a, we can verify with 2 classical queries that $f(0) = f(a)$

* but it is not in BPP^O : we need a number of randomized queries that is exponential in n to find the solution with success proba $> \frac{1}{2}$.

How do we see that ?
The best we can do classically is to randomly query a number of times (k times) which random values of the input each time.

We will only solve the pb if we are fortunate enough to choose two inputs x and x' that get mapped to the same $f(x)$ (ie if $x' = x \oplus a$) -

What is the proba that this happens ?

• with k queries we can form $\binom{k}{2} = \frac{k(k-1)}{2}$ different pairs (x, x').

• for each pair the proba that $x' = x \oplus a$ is $\frac{1}{2^n - 1}$

$\Rightarrow$ total proba of successfuly finding a with k queries =

$$p(\text{success}) = \frac{k(k-1)}{2(2^n-1)} < \frac{k^2}{2^n}$$

$\Rightarrow$ even with exponentially number of queries k, e.g.

$$k = 2^{n(1-\varepsilon)/2}$$

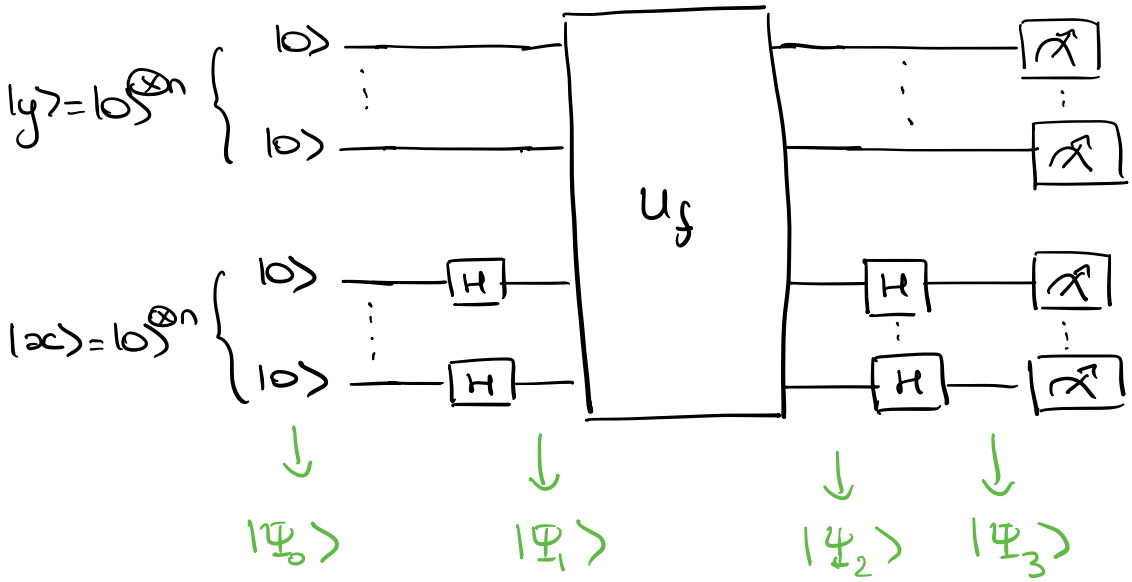
$$p(\text{success}) < \frac{2^{n(1-\varepsilon)}}{2^n} = 2^{-n\varepsilon}$$

$\Rightarrow$ still exponentially small.

$\Rightarrow$ Simon's pb is not in BPP^O

But we will now show that it is in BQP^O.
(i.e. that we can solve the pb efficiently if quantum queries are allowed)

The quantum algo. is represented by a circuit similar to the Deutsch-Josza circuit but now <u>both</u> query and answer registers are expanded to n qubits:



- $|\Psi_0\rangle = |x\rangle \otimes |y\rangle = |0\rangle^{\otimes n} \otimes |0\rangle^{n}$

- $|\Psi_1\rangle = H^{\otimes n} |x\rangle \otimes |y\rangle$

$$= H^{\otimes n} |0\rangle^{\otimes n} \otimes |0\rangle^{\otimes n}$$

$$= \frac{1}{2^{n/2}} \sum_{z=0}^{2^n-1} |z\rangle \otimes |0\rangle^{\otimes n}$$

- $|\Psi_2\rangle = \frac{1}{2^{n/2}} \sum_{z=0}^{2^n-1} |z\rangle \otimes |f(z)\rangle$

⇒ At this point the state of the total system is a massively entangled state.

- Now we measure the top register.

The measurement outcome is selected at of the 2^{n-1} possible values of $f(z)$ with equal probability → e.g. $f(z_0)$

($2^{n-1} = \frac{2^n}{2}$ values because the fct is 2 to 1)

⇒ due to entanglement the bottom register collapses to a superposition:

$$\frac{|z_0\rangle + |z_0 \oplus a\rangle}{\sqrt{2}}$$

(because both z_0 and $z_0 \oplus a$ (and only these values) are mapped to $f(z_0)$).

Now we want to extract information about the period a.

It would be no good to measure the bottom register in the computational basis now as we would get either $|z_0\rangle$ or $|z_0 \oplus a\rangle$ with proba $1/2$, but would not learn anything about a.

. Instead we first apply $H^{\otimes n}$.

We showed before that

$$H^{\otimes n} |z_0\rangle = \frac{1}{2^{n/2}} \sum_{z'=1}^{2^n-1} (-1)^{z_0 \cdot z'} |z'\rangle$$

$$\Rightarrow H^{\otimes n} |z_0 \oplus a\rangle = \frac{1}{2^{n/2}} \sum_{z'=1}^{2^n-1} (-1)^{(z_0 \oplus a) \cdot z'} |z'\rangle$$

when we put this together, interferences will occur:

$$H^{\otimes n} \left(\frac{|z_0\rangle + |z_0 \oplus a\rangle}{\sqrt{2}} \right) =$$

$$= \frac{1}{2^{n/2}} \frac{1}{\sqrt{2}} \sum_{z'=1}^{2^n-1} (-1)^{z_0 \cdot z'} \left[1 + (-1)^{a \cdot z'} \right] |z'\rangle$$

The term

$$\left[1 + (-1)^{a \cdot z'}\right] \begin{cases} = 0 & \text{if } a \cdot z' = 1 \pmod 2 \\ & \rightarrow \text{destructive interference} \\ = 2 & \text{if } a \cdot z' = 0 \pmod 2 \\ & \rightarrow \text{constructive interference} \end{cases}$$

$$\Rightarrow H^{\otimes n} \left(\frac{|z_0\rangle + |z_0 \oplus a\rangle}{\sqrt{2}} \right)$$

$$= \frac{1}{2^{(n-1)/2}} \sum_{z'=1}^{2^n - 1} (-1)^{z_0 \cdot z'} |z'\rangle$$

$$\left\{ \begin{array}{l} \text{such that} \\ a \cdot z' = 0 \end{array} \right\}$$

- We now measure the bottom register - The measurement outcome $|z'\rangle$ is then selected at random from all possible values of z' orthogonal to a (i.e. with $a \cdot z' = 0$), each occuring with proba $\left(\frac{1}{2^{(n-1)/2}} \right)^2 = \frac{1}{2^{(n-1)}}$

Note: The phase $(-1)^{3_0 \cdot 3'}$ does not affect the proba distribution of the outcomes. (and thus 3_0)

⇒ with each measurement we learn something about a but one run is not enough. We need to run the algorithm multiple times, each time we get a value of $3'$ orthogonal to a.

Once we have found n different values $3'_{(1)}, \dots 3'_{(n)}$, we can solve the system of equations:

$$\begin{cases} 3'_{(1)} \cdot a = 0 \\ 3'_{(2)} \cdot a = 0 \\ \vdots \\ 3'_{(n)} \cdot a = 0 \end{cases}$$

(matrix inversion problem)

→ this uniquely determines the period a.

⇒ The period a is found with

$$\underline{O(n) \text{ queries}}$$

⇒ exponential speed-up wrt classical algo!

$\Rightarrow$ The Simon's pb is therefore an oracle pb that can be solved in polynomial time with quantum queries (superposition input states) while exponential time is required if one is limited to (randomized) classical queries (computational basis states as input) –

$$\Rightarrow \quad BPP^{O} \neq BQP^{O}$$

where "O" refers to Simon's oracle.

This is a remarkable result –
However this pb does not have any known practical application.

(would need an instantiation of Simon's oracle such that the pb is still hard classically even when the fct is known).

Because of that Simon's 1994 original paper was in fact rejected.

However this pb and solution inspired another pb, for which separation btw classical and quantum query complexity can be established, and which has practical applications → finding the period of a fct (see soon)