LECTURE 4
INTRODUCTION TO QUANTUM COMPUTING, THE CIRCUIT MODEL

Thomas Debris-Alazard

Inria, École Polytechnique

Computer science: art of computing...

What do we mean by quantum computing?

$\longrightarrow$ The quantum circuit model!

What is the cost to compute 2^n?

What is the cost to compute 2^n?

- Trivial approach: compute $2 \times 2 \times 2 \times \ldots\ n$ times...

  $\longrightarrow$ It costs n operations!

- Clever approach: recursive algorithm, given n if $n > 1$ compute res $\leftarrow 2^{n/2}$ and compute res^2 otherwise output 2

  $\longrightarrow$ It costs $\approx \log_2(n)$ operations $\Big($exponential improvement$\Big)$!

Two lessons to take-away:

1. You have to be smart when computing something $\Big($algorithmic science$\Big)$
2. A first model of cost: enumerate the number of basic operations $\Big($additions and multiplications$\Big)$

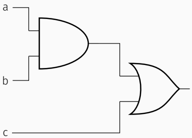
   $\longrightarrow$ It is an high level point of view, often convenient but rather "limited"

Boolean Circuits:

In what follows: focus on a "low" level to estimate the computational cost

$\longrightarrow$ boolean circuits & number of gates

Boolean circuit: finite directed acyclic $\Big($no loop$\Big)$ graph with **AND**, **OR** and **NOT** classical gates which has input and output nodes



A circuit computes $f : \{0,1\}^n \longrightarrow \{0,1\}^m$ if given n input bits **x**, it outputs m bits given by f(**x**)

Two questions:

- What are the classical gates that enable to compute any function $f : \{0,1\}^n \longrightarrow \{0,1\}^m$?
- How to define the efficiency of a circuit?

Universality:

Logic gates **AND**, **OR** and **NOT** are enough to compute any function $f : \{0,1\}^n \longrightarrow \{0,1\}^m$

$\left(\text{yes these gates enable to compute } n \mapsto 2^{n}\right)$

Is it doable quantumly?

Problem: *any quantum operation is invertible* $\left(\textit{even unitary}\right)$ but **AND** is not invertible. . .

Universality:

Logic gates **AND**, **OR** and **NOT** are enough to compute any function $f : \{0,1\}^n \longrightarrow \{0,1\}^m$

$\Big($yes these gates enable to compute $n \mapsto 2^n$ $\Big)$

Is it doable quantumly?

Problem: *any quantum operation is invertible* $\Big($*even unitary*$\Big)$ but **AND** is not invertible. . .

Toffoli $\Big($also CCNOT$\Big)$ gate:

The Toffoli gate takes 3 input bits and it outputs 3 bits as follows:

$$\textbf{Toffoli}(x, y, z) = \Big(x, y, z \textbf{ XOR } (x \textbf{ AND } y)\Big)$$

Inversability and universality:

- The Toffoli gate is invertible
- Any classical circuit computing a function f consisting of N gates in the set $\{\textbf{AND}, \textbf{OR}, \textbf{NOT}\}$ can be computed using $O(N)$ Toffoli gates

$\longrightarrow$ In particular: the number of Toffoli gates is roughly the same

Many different circuits can compute a function $f : \{0,1\}^n \longrightarrow \{0,1\}^m$

How can we distinguish them?

Many different circuits can compute a function $f : \{0,1\}^n \longrightarrow \{0,1\}^m$

How can we distinguish them?

$\longrightarrow$ Some circuits are more efficient than others!

Running time:

We define the running time of a circuit computing f as the number of used gates **AND**, **OR** and **NOT**

Ideal situation: an efficient circuit

Given n input nodes: the circuit uses $O(n^k)$ gates for some constant k

$\longrightarrow$ We say that it has a cost $\text{poly}(n)$

In this course: we only care of being $\text{poly}(n)$ $\Big($even if the constant k is large...$\Big)$

Exercise:

Is it equivalent to define our running-time model as the number of **Toffoli** gates to compute a function f? Why?

But is the classical circuit model meaningful?

P: class of languages $L \subseteq \{0,1\}^\star$ “for which it exists an efficient algorithm” to decide $x \in L$ or not

Complexity theory: uniformly polynomial circuits

Family of circuits $C \stackrel{\text{def}}{=} \{C_n\}_n$ with n input bits and one output bit *such that* there is polylog(n)-space Turing machine that outputs C_n given n

$$L_C \stackrel{\text{def}}{=} \bigcup_n \left\{ \mathbf{x} \in \{0,1\}^n \ : \ C_n(\mathbf{x}) = 1 \right\}$$

$L \in \mathrm{P}$ if and only if there exits a uniform family of circuits C such that $L = L_C$

$\longrightarrow$ Given a uniform family of circuits $C = \{C_n\}$: **C_n has at most poly(n)-gates!**

What about quantum computation?

Is the circuit model reasonable? If yes, what is doable quantumly and at which cost?

What about quantum computation?

Is the circuit model reasonable? If yes, what is doable quantumly and at which cost?

Two intuitions:

- "Quantum circuit" can simulate classical circuits because Toffoli gates are universal and invertible. . .

  $\longrightarrow$ Therefore: quantum circuits define a "reasonable" model of computation

- Complexity of computation will be taken into account from the number of "quantum gates"

  $\longrightarrow$ Therefore: we expect quantum circuits to measure the complexity in a similar vein than in the classical case

NOTATION AND BASIC CIRCUITS

During this course we consider the state space $\mathbb{C}^{2^n} = \underbrace{\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2}_{n \text{ times}}$ of n-qubits register

State space, computational basis and measurement:

We will always write n-qubits registers as

$$\sum_{\mathbf{x} \in \{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle \quad \text{where } |\mathbf{x}\rangle = |x_1, \ldots, x_n\rangle \ \Big(= |x_1\rangle \otimes \cdots \otimes |x_n\rangle \Big) \text{ and } \sum_{\mathbf{x} \in \{0,1\}^n} |\alpha_{\mathbf{x}}|^2 = 1$$

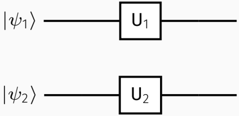
The family $(|\mathbf{x}\rangle)_{\mathbf{x} \in \{0,1\}^n}$ is known as the computational basis

$\longrightarrow$ All the considered measurements $\Big($in this course$\Big)$ will be in the computational basis

Given two quantum states $|\psi_1\rangle, |\psi_2\rangle$ and two unitaries $\mathsf{U}_1, \mathsf{U}_2$, the circuit representation of

$$(\mathsf{U}_1 \otimes \mathsf{U}_2)\,(|\psi_1\rangle \otimes |\psi_2\rangle)$$

is given by



Exercise:

1. What becomes $\frac{|00\rangle+|01\rangle}{\sqrt{2}}$ when feeding to the above circuit?
2. Describe a quantum circuit that transforms $|00\rangle$ into $\frac{|10\rangle-|11\rangle}{\sqrt{2}}$

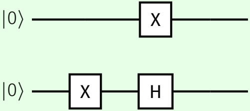
QUANTUM CIRCUIT AND TENSORIAL PRODUCT

Solution:

1. What becomes $\frac{|00\rangle+|01\rangle}{\sqrt{2}}$ when feeding to the above circuit?

   It becomes: $\mathbf{U}_1\,|0\rangle \otimes \mathbf{U}_2\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right) = \frac{1}{\sqrt{2}}\mathbf{U}_1\,|0\rangle \otimes \mathbf{U}_2\,|0\rangle + \frac{1}{\sqrt{2}}\mathbf{U}_1\,|0\rangle \otimes \mathbf{U}_2\,|1\rangle$

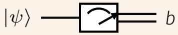
2. Describe a quantum circuit that transforms $|00\rangle$ into $\frac{|10\rangle-|11\rangle}{\sqrt{2}}$



A measurement *in the computational basis* converts $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ into a probabilistic classical bit $b \in \{0, 1\}$ where

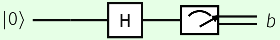
$$\mathbb{P}(b = 0) = |\alpha|^2 \quad \text{and} \quad \mathbb{P}(b = 1) = |\beta|^2$$

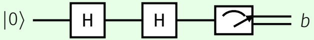
The circuit representation of a measurement is:



Exercise:

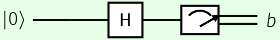
Give the distribution of the following probabilistic bits b:

1. 

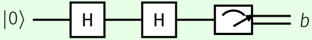
2. 

Solution:

Give the distribution of the following probabilistic bits b:

1. 

   The output bit b is uniform, namely: $\mathbb{P}(b = 0) = \mathbb{P}(b = 1) = \frac{1}{2}$

2. 

   As $\mathbf{H}^2 = \mathbf{I}_2$, the output bit b is always zero

THE QUANTUM CNOT GATE:

Let us introduce the **Controlled-NOT** gate $\Big($unitary$\Big)$ over 2-qubits:

$$\mathsf{CNOT} : |a, b\rangle \mapsto |a, a \oplus b\rangle$$

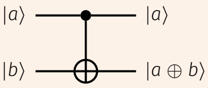
It is a unitary $\Big($it maps the computational basis to the computational basis$\Big)$

Quantum CNOT-gate $|a, b\rangle \mapsto |a, a \oplus b\rangle$

- **Matrix representation:**

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

- **Circuit representation:**



$$|a, b\rangle \mapsto |a, a \oplus b\rangle$$

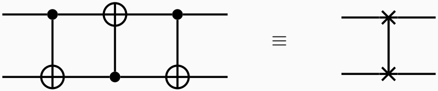
is the quantum generalization of the **XOR** operation!

Be careful:

The **XOR** operation $(a, b) \mapsto a \oplus b$ cannot be a quantum operation because is not invertible

SWAPPING

Given two wires, is it possible to swap two qubits?



$$\begin{aligned} |a, b\rangle &\longrightarrow |a, a \oplus b\rangle \\ &\longrightarrow |a \oplus (a \oplus b)\,, a \oplus b\rangle \\ &\longrightarrow |b, (a \oplus b) \oplus b\rangle \\ &= |b, a\rangle \end{aligned}$$

Given a qubit $|\psi\rangle$, is it possible to build a quantum circuit that copies it?

$\longrightarrow$ **No!** Because the no-cloning theorem

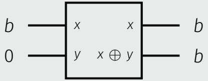
But it is doable for classical bit $(b, 0) \mapsto (b, 0 \oplus b) = (b, b) \ldots$

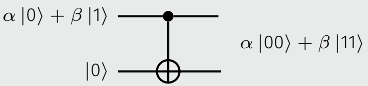
Given a qubit $|\psi\rangle$, is it possible to build a quantum circuit that copies it?

$\longrightarrow$ No! Because the no-cloning theorem

But it is doable for classical bit $(b, 0) \mapsto (b, 0 \oplus b) = (b, b) \ldots$

Take a look at the quantum case:



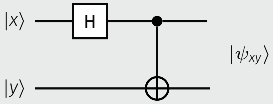


We have built an entangled state!

Bell states:

$$|\psi_{xy}\rangle \stackrel{\text{def}}{=} \frac{|0, y\rangle + (-1)^x \, |1, (1 \oplus y)\rangle}{\sqrt{2}}$$

The quantum circuit building Bell states:



$$|x, y\rangle \xrightarrow{\text{H}\otimes\text{I}_2} \frac{|0\rangle + (-1)^x \, |1\rangle}{\sqrt{2}} \otimes |y\rangle = \frac{|0, y\rangle + (-1)^x \, |1, y\rangle}{\sqrt{2}} \xrightarrow{\text{CNOT}} \frac{|0, y\rangle + (-1)^x \, |1, (1 \oplus y)\rangle}{\sqrt{2}}$$

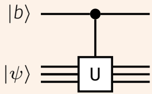
CONTROLLED UNITARY

Controlled U-gate:

Let $\mathbf{U}$ be any unitary over n-qubits. The controlled $\mathbf{U}$-gate has one control qubit $|b\rangle$ and n target qubits $|\psi\rangle$. It is defined as

- If $b = 0$, it outputs $|b\rangle \otimes |\psi\rangle$
- If $b = 1$, it outputs $|b\rangle \otimes \mathbf{U}\,|\psi\rangle$

Circuit representation:



$\longrightarrow$ Controlled-$\mathbf{U}$ $\equiv$ If condition then instruction $\mathbf{U}$ otherwise do nothing

Exercise:

Show that the CNOT gate is the controlled $\mathbf{X}$-gate

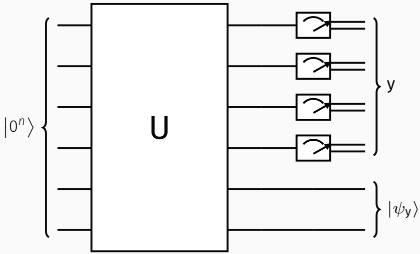
Quantum circuits: starting from n qubits initialized at $|0^n\rangle$ and then successively apply the two admissible operations $\big($unitary and measurements$\big)$

Applying $\mathbf{U}_1$ and then $\mathbf{U}_2$ is equivalent to applying $\mathbf{U}_2\mathbf{U}_1$

$\longrightarrow$ We can assume the algorithm performs a unitary, then a measurement, then a unitary, then measurement and so on...

We will consider only algorithms where we first perform all the unitary operations and then perform measurements in the computational basis

$\longrightarrow$ As powerful as general algorithms $\big($admitted$\big)$



AUXILIARY QUBITS

$$\mathsf{U} : |\psi\rangle \longrightarrow \mathsf{U}\,|\psi\rangle$$

$\longrightarrow$ It is often easier to build $\mathsf{U}' : |\psi\rangle\,|0\rangle_{\text{aux}} \longrightarrow \mathsf{U}\Big(\,|\psi\rangle\,\Big)\,|0\rangle_{\text{aux}}$

Extra qubits are called auxiliary qubits, ancilliary qubits or workspace

$\longrightarrow$ it is important that they start at $|0\rangle$ and end at $|0\rangle$ $\Big($see Exercise Session$\Big)$

SOLOVAY-KITAEV THEOREM AND GATE MODEL

Any classical function can be computed with gates $\{\textbf{AND}, \textbf{OR}, \textbf{NOT}\}$ (universal gates)

What are the universal quantum gates?

The following gate is crucial:

The $\pi/8$-gate:

It maps $|0\rangle \mapsto |0\rangle$ and $|1\rangle \mapsto e^{i\pi/4}|1\rangle$:

$$\mathbf{T} \stackrel{\text{def}}{=} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

Origin of the terminology:

Up to an unimportant global phase $\mathbf{T}$ is equal to $\mathbf{T} = e^{i\pi/8}\begin{pmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{pmatrix}$

$\left\{\mathsf{CNOT}, \mathsf{H}, \mathsf{T}\right\}$ *are universal quantum gates*

Solovay-Kitaev theorem (admitted):

Let $\mathcal{G} = \{\mathsf{CNOT}, \mathsf{H}, \mathsf{T}\}$. Any unitary $\mathbf{U}$ over n-qubits can be approximated by applying

$$O\left(2^{2n} \log^4\left(\frac{1}{\varepsilon}\right)\right)$$

gates from $\mathcal{G}$ with accuracy ε

In other words, from the description of $\mathbf{U}$, one can construct a sequence $\mathbf{G}_1, \ldots, \mathbf{G}_N \in \mathcal{G}$ with $N = O(2^{2n} \log^4(\frac{1}{\varepsilon}))$ and

$$\|\mathbf{G}_N \ldots \mathbf{G}_1 - \mathbf{U}\| \leq \varepsilon,$$

where $\|\mathbf{G}_N \cdots \mathbf{G}_1 - \mathbf{U}\| \stackrel{\text{def}}{=} \max_{|\psi\rangle} \|\mathbf{G}_N \cdots \mathbf{G}_1 |\psi\rangle - \mathbf{U} |\psi\rangle\|$ is the *operator norm*

$\longrightarrow$ The log term is important: exponential accuracy with a polynomial number of gates

Other universal quantum gates?

Yes! The **CNOT** and qubits gates are also universal

How many resources are needed to compute a fixed unitary **U over n qubits**?

- **First definition:** it requires one resource, the unitary **U**

$\longrightarrow$ *Stupid definition*: same thing that saying, to compute classically any function f asks one resource, the function f

We want a the smallest and simplest set of operations to define the needed resources

How many resources are needed to compute a fixed unitary **U over n qubits**?

- **First definition:** it requires one resource, the unitary **U**

$\longrightarrow$ *Stupid definition*: same thing that saying, to compute classically **any** function f asks one resource, the function f

We want a **the smallest and simplest** set of operations to define the needed resources

- **Second definition:** the number of quantum gates $\{\mathbf{CNOT}, \mathbf{H}, \mathbf{T}\}$ to approximate well-enough **U**

$\longrightarrow$ *Problem*: is this definition meaningful?

How many resources are needed to compute a fixed unitary U over n qubits?

- **First definition:** it requires one resource, the unitary **U**

$\longrightarrow$ *Stupid definition*: same thing that saying, to compute classically any function f asks one resource, the function f

We want a the smallest and simplest set of operations to define the needed resources

- **Second definition:** the number of quantum gates $\{\mathbf{CNOT}, \mathbf{H}, \mathbf{T}\}$ to approximate well-enough **U**

$\longrightarrow$ *Problem*: is this definition meaningful? Yes, by Solovay-Kitaev,

possible with $O\left(2^{2n} \log^4\left(\frac{1}{\varepsilon}\right)\right)$ gates $\{\mathbf{CNOT}, \mathbf{H}, \mathbf{T}\}$ to approximate any unitary

How many resources are needed to compute a fixed unitary **U over n qubits**?

- **First definition:** it requires one resource, the unitary **U**

$\longrightarrow$ *Stupid definition*: same thing that saying, to compute classically any function f asks one resource, the function f

We want a the smallest and simplest set of operations to define the needed resources

- **Second definition:** the number of quantum gates $\{\mathsf{CNOT}, \mathsf{H}, \mathsf{T}\}$ to approximate well-enough **U**

$\longrightarrow$ *Problem*: is this definition meaningful? Yes, by Solovay-Kitaev,

possible with $O\left(2^{2n}\log^4\left(\frac{1}{\varepsilon}\right)\right)$ gates $\{\mathsf{CNOT}, \mathsf{H}, \mathsf{T}\}$ to approximate any unitary

Be careful:

Solovay-Kitaev tells it is possible to approximate any unitary by using $\{\mathsf{CNOT}, \mathsf{H}, \mathsf{T}\}$
but a priori it asks for 2^{2n} resources. . .

Does any unitary need an exponential number of $\{\mathsf{CNOT}, \mathsf{H}, \mathsf{T}\}$ to be built?

How many resources are needed to compute a fixed unitary U over n qubits?

- **First definition:** it requires one resource, the unitary **U**

$\longrightarrow$ *Stupid definition*: same thing that saying, to compute classically any function f asks one resource, the function f

We want a the smallest and simplest set of operations to define the needed resources

- **Second definition:** the number of quantum gates $\{\mathsf{CNOT}, \mathsf{H}, \mathsf{T}\}$ to approximate well-enough **U**

$\longrightarrow$ *Problem*: is this definition meaningful? Yes, by Solovay-Kitaev,

possible with $O\left(2^{2n}\log^4\left(\frac{1}{\varepsilon}\right)\right)$ gates $\{\mathsf{CNOT}, \mathsf{H}, \mathsf{T}\}$ to approximate any unitary

Be careful:

Solovay-Kitaev tells it is possible to approximate any unitary by using $\{\mathsf{CNOT}, \mathsf{H}, \mathsf{T}\}$
but a priori it asks for 2^{2n} resources. . .

Does any unitary need an exponential number of $\{\mathsf{CNOT}, \mathsf{H}, \mathsf{T}\}$ to be built?

No! As for classical computations there are algorithms/unitaries easy to compute, other not. . .

A reasonable model to define the cost of a quantum computation, i.e. computing a unitary

. The number of $\left\{\text{CNOT}, \text{H}, \text{T}\right\}$ to approximate well-enough the unitary

But would you be happy to implement X *or* Y *with this set of quantum gates?*

$\longrightarrow$ *A priori* ***no***! The set of operations $\left\{\text{CNOT}, \text{H}, \text{T}\right\}$ is not very flexible...

Unitary over 1 **and** 2-**qubits** are the "simplest" operations

Wouldn't be more reasonable to use as model of cost: the number of unitaries over 1 and 2-qubits?

A reasonable model to define the cost of a quantum computation, i.e. computing a unitary

. The number of $\left\{\mathsf{CNOT}, \mathsf{H}, \mathsf{T}\right\}$ to approximate well-enough the unitary

But would you be happy to implement X *or* Y *with this set of quantum gates?*

$\longrightarrow$ *A priori* ***no****!* The set of operations $\left\{\mathsf{CNOT}, \mathsf{H}, \mathsf{T}\right\}$ is not very flexible...

Unitary over 1 **and** 2**-qubits** are the "simplest" operations

Wouldn't be more reasonable to use as model of cost: the number of unitaries over 1 and 2-qubits?

Yes and by Solovay-Kitaev both models are "poly(λ)-equivalent"

We can approximate any unitary over 1 and 2 qubits with accuracy $2^{-\lambda}$ and

$O\left(\lambda^4\right)$ quantum gates $\left\{\mathsf{CNOT}, \mathsf{H}, \mathsf{T}\right\}$

The quantum gate model:

The quantum running time of a unitary $\mathbf{U}$ is the amount of 1 and 2-qubit gates needed to apply $\mathbf{U}$

The running time of a single-qubit measurement is 1

Exercise:

Give a simple argument to explain why quantum gates over 1-qubit are not universal, *i.e.* are not enough to describe any quantum computation

One may say that estimating the running time as the number of 1-2 qubits unitaries is an overkill

$\longrightarrow$ It can be hard to build some 1 or 2 qubits unitary...

A more reasonable model:

Running time: number **H**, **T** and **CNOT** gates that are used

$\longrightarrow$ The "difficulty" to implement quantum circuits reduces to build this small set of gates!

By the Solovay-Kitaev theorem:

The running time of the above model is the same than in the quantum gate model, but up to polynomial factor $\left(\text{in the input length } n\right)$ if one targets an exponentially close accuracy...

In conclusion: lot of debates to define the running time of quantum circuits...

For us: no debates, we don't care of polynomial factors (even if it is a hard problem to handle in "practice"...) and we will use the quantum gate model

To take away: you said algorithm?

- Algorithm: series of simple and determined in advance instructions $\Big($*addition, multiplication, if condition then instruction, while condition do instruction*$\Big)$

  $\longrightarrow$ Efficient algorithm: small number of instructions!

- Quantum algorithm: series of 1, 2-qubits unitaries and then measurements

  $\longrightarrow$ Efficient quantum algorithm: small amount of 1, 2-qubits unitaries and measurements!

Efficient quantum algorithm: poly(n)-repetitions of a circuit starting from $|0^n\rangle$ with poly(n) unitaries and measurements over 1, 2-qubits

Efficient computing: a difficult task

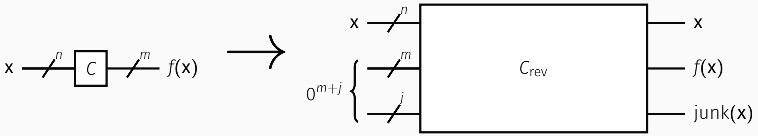
For many problems, it is $\Big($very$\Big)$ hard to find a small number of instructions solving it

Shor's quantum algorithm has been a breakthrough: it solves with "few" quantum-instructions a problem $\Big($factoring$\Big)$ such that all known classical algorithms ask a huge number of instructions...

CLASSICAL CIRCUITS WITH QUANTUM CIRCUITS

CLASSICAL CASE

Computing classically a function f with T gates can be transformed into a reversible circuit C_{rev} that only consists of $O(T)$ Toffoli gates, possibly with some junk state $\mathrm{junk}(\mathbf{x})$.



Informally, the junk part keeps a place to perform intermediary computations

Simulating classical circuits with quantum circuits:

Classical Toffoli gates can be interpreted as a quantum unitary acting on three qubits:

$$\textbf{Toffoli}\,|x, y, z\rangle \overset{\text{def}}{=} |x, y, z \oplus xy\rangle$$

Therefore: C_{rev} can be interpreted as a unitary $\mathbf{U}$:

$$\mathbf{U}\left|\mathbf{x}, 0^{m+j}\right\rangle \overset{\text{def}}{=} |\mathbf{x}\rangle\,|f(\mathbf{x})\rangle\,|\mathrm{junk}(\mathbf{x})\rangle$$

$\longrightarrow$ Quantum computers are at least as powerful as classical computers!

The unitary U_f:

For any function $f : \{0,1\}^n \to \{0,1\}^m$ that can be computed classically with a circuit running in time T, there exists a quantum circuit on $n+m$ qubits that runs in time $O(T)$ that can perform the unitary

$$U_f : |x\rangle\,|y\rangle \to |x\rangle\,|y \oplus f(x)\rangle$$

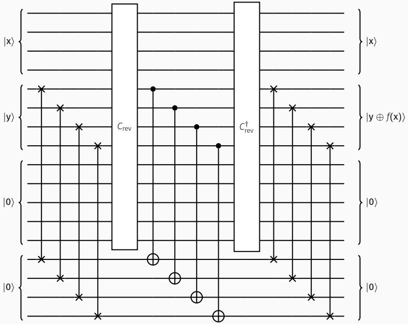
Be careful:

$|x\rangle \mapsto |f(x)\rangle$ may not be a quantum operation $\Big($for instance f be the zero function$\Big)$

$\longrightarrow$ The auxiliary qubit $|y\rangle$ ensures that **U_f is a unitary!**

Proof:

1. On input $|x\rangle\,|y\rangle\,|0\rangle\,|0\rangle$, first swap the second and fourth registers to get $|x\rangle\,|0\rangle\,|0\rangle\,|y\rangle$.
2. Apply C_{rev} on the 3 first registers to get the state $|x\rangle\,|f(x)\rangle\,|\mathrm{junk}(x)\rangle\,|y\rangle$.
3. For i from 1 to m, apply a **CNOT** gate between the i^{th} wire of the second register and the i^{th} wire of the forth register. We then have the state $|x\rangle\,|f(x)\rangle\,|\mathrm{junk}(x)\rangle\,|y \oplus f(x)\rangle$.
4. Apply $C_{\mathrm{rev}}^{\dagger}$ on the three first registers to get the state $|x\rangle\,|0\rangle\,|0\rangle\,|y \oplus f(x)\rangle$.
5. Swap the second and forth register to get the state $|x\rangle\,|y \oplus f(x)\rangle\,|0\rangle\,|0\rangle$.

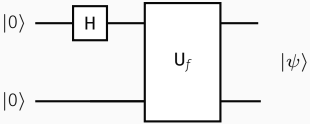


QUANTUM PARALLELISM AND INTERFERENCE

$$\text{Let } f : \{0,1\} \to \{0,1\}$$

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

Consider the following quantum circuit:



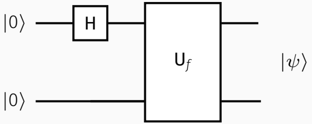
What quantum state is $|\psi\rangle$?

QUANTUM PARALLELISM: ONE BIT FUNCTIONS

$$\text{Let } f : \{0,1\} \to \{0,1\}$$

$$\mathbf{U}_f : |x\rangle\, |y\rangle \to |x\rangle\, |y \oplus f(x)\rangle$$

Consider the following quantum circuit:



What quantum state is $|\psi\rangle$?

1. After the first gate we have: $\frac{|0\rangle+|1\rangle}{\sqrt{2}} \otimes |0\rangle = \frac{|00\rangle+|10\rangle}{\sqrt{2}}$,
2. Applying $\mathbf{U}_f$ leads to $\Big($use the linearity$\Big)$:

$$|\psi\rangle = \frac{|0,f(0)\rangle+|1,f(1)\rangle}{\sqrt{2}}$$

$\longrightarrow$ We have a superposition of the values of f

Tensorization of the Hadamard gate:

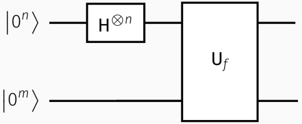
Consider,

$$\mathsf{H}^{\otimes n} \stackrel{\text{def}}{=} \underbrace{\mathsf{H} \otimes \cdots \otimes \mathsf{H}}_{n \text{ times}}$$

Then,

$$\mathsf{H}^{\otimes n} \left|0^n\right\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$$

The following circuit performs the quantum parallelism $\left(\text{here } f : \{0,1\}^n \to \{0,1\}^m\right)$



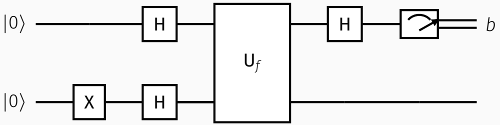
$$\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x, f(x)\rangle$$

Measurement of $\frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}} |\mathbf{x}, f(\mathbf{x})\rangle$ gives $f(\mathbf{x})$ for only one value of $\mathbf{x}$ . . .

$\longrightarrow$ Interference is a nice example of how using quantum parallelism!

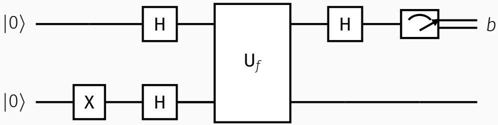
The "−1" of the Hadamard gate gives you a huge power. . .

Consider the following circuit $\left(\text{here } f : \{0,1\} \to \{0,1\}\right)$



What is the value of b?

Consider the following circuit $\left(\text{here } f : \{0,1\} \to \{0,1\}\right)$



What is the value of b?

1. After applying the **X** and **H** gates: $\frac{|0\rangle+|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle-|1\rangle}{\sqrt{2}} = \frac{|00\rangle - |01\rangle + |10\rangle - |11\rangle}{2}$,

2. Applying $\mathbf{U}_f$ leads to (use the linearity):

$$\frac{|0, f(0)\rangle - |0, 1\oplus f(0)\rangle + |1, f(1)\rangle - |1, 1\oplus f(1)\rangle}{2} = \begin{cases} \pm \frac{|0\rangle+|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle-|1\rangle}{\sqrt{2}} & \text{if } f(0) = f(1) \\ \pm \frac{|0\rangle-|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle-|1\rangle}{\sqrt{2}} & \text{if } f(0) \neq f(1) \end{cases}$$

3. Applying the last Hadamard gate leads to (use that $\mathbf{H}^2 = \mathrm{I}_2$):

$$\begin{cases} \pm |0\rangle \otimes \frac{|0\rangle-|1\rangle}{\sqrt{2}} & \text{if } f(0) = f(1) \\ \pm |1\rangle \otimes \frac{|0\rangle-|1\rangle}{\sqrt{2}} & \text{if } f(0) \neq f(1) \end{cases}$$

4. Measuring the first qubit always leads to $f(0) \oplus f(1)$!

$\longrightarrow$ We obtained a global property of f $\left(\textit{i.e.}, f(0) \oplus f(1)\right)$ with only one evaluation of $f(x)$!

SIMON'S ALGORITHM

The problem:

- **Input:** A function $f : \{0,1\}^n \longrightarrow \{0,1\}^n$
- **Promise:** $\exists \mathsf{s} \in \{0,1\}^n$: $\Big(f(\mathsf{x}) = f(\mathsf{y}) \iff (\mathsf{x} = \mathsf{y})$ or $(\mathsf{x} = \mathsf{y} \oplus \mathsf{s})\Big)$
- **Goal:** Find s

1. Start from the $2n$ qubit state, with 2 registers of n qubits

$$|\psi_0\rangle = |0^n\rangle |0^n\rangle$$

2. Apply $\mathsf{H}^{\otimes n}$ on the first n qubits to get

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle |0^n\rangle$$

3. Apply U_f on the state to get

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle = \frac{1}{\sqrt{\sharp \mathrm{Im}(f)}} \sum_{y \in \mathrm{Im}(f)} \frac{1}{\sqrt{2}} \left(|x_y\rangle + |x_y \oplus s\rangle\right) |y\rangle$$

4. Measure the second register and obtain some value $y \in \mathrm{Im}(f)$. The resulting state on the first register is

$$|\psi_4(y)\rangle = \frac{1}{\sqrt{2}} \left(|x_y\rangle + |x_y \oplus s\rangle\right)$$

5. Apply $\mathsf{H}^{\otimes n}$ on the first register to get

$$|\psi_5(y)\rangle = \frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} \left(\frac{1}{\sqrt{2}} (-1)^{x_y \cdot z} + \frac{1}{\sqrt{2}} (-1)^{(x_y \oplus s) \cdot z} \right) |z\rangle$$

5. Apply $\mathsf{H}^{\otimes n}$ on the first register to get

$$|\psi_5(\mathsf{y})\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathsf{z}\in\{0,1\}^n} \left(\frac{1}{\sqrt{2}}(-1)^{\mathsf{x_y}\cdot\mathsf{z}} + \frac{1}{\sqrt{2}}(-1)^{(\mathsf{x_y}\oplus\mathsf{s})\cdot\mathsf{z}} \right) |\mathsf{z}\rangle \,.$$

Now, if $\mathsf{s}\cdot\mathsf{z} = 0 \bmod 2$, we have $\left(\frac{1}{\sqrt{2}}(-1)^{\mathsf{x_y}\cdot\mathsf{z}} + \frac{1}{\sqrt{2}}(-1)^{(\mathsf{x_y}\oplus\mathsf{s})\cdot\mathsf{z}} \right) = \sqrt{2}(-1)^{\mathsf{x_y}\cdot\mathsf{z}}$ and if $\mathsf{s}\cdot\mathsf{z} = 1 \bmod 2$, we have $\left(\frac{1}{\sqrt{2}}(-1)^{\mathsf{x_y}\cdot\mathsf{z}} + \frac{1}{\sqrt{2}}(-1)^{(\mathsf{x_y}\oplus\mathsf{s})\cdot\mathsf{z}} \right) = 0$. Therefore, we can write

$$|\psi_5(\mathsf{y})\rangle = \sqrt{\frac{2}{2^n}} \sum_{\substack{\mathsf{z}\in\{0,1\}^n \\ \mathsf{s}\cdot\mathsf{z}=0 \bmod 2}} (-1)^{\mathsf{x_y}\cdot\mathsf{z}} |\mathsf{z}\rangle \,.$$

6. Measure this state in the computational basis. You get a random z satisfying $\mathsf{z}\cdot\mathsf{s} = 0$

WHY DOES IT WORK?

$\mathbb{F}_2$ denotes $\{0,1\}$ modulo 2. It is a field

$\mathbb{F}_2^n$ is a n-dimensional $\mathbb{F}_2$ vector space

$\left\{\mathbf{z} \in \mathbb{F}_2^n : \ \mathbf{z} \cdot \mathbf{s} = \sum_{i=1}^n z_i s_i = 0 \in \mathbb{F}_2\right\}$ is a subspace of $\mathbb{F}_2^n$ with dimension $n-1$

The above algorithm gives $(z_1, \ldots, z_n)$ s.t $\sum_{i=1}^n z_i s_i = 0 \textbf{ mod } 2$

We repeat the algorithm m times to get m random values $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)} \in \mathbb{F}_2^n$ satisfying $\mathbf{z}^{(k)} \cdot \mathbf{s} = 0$

We obtain the following system $\Big(\mathbf{s}$ is the unknown$\Big)$: $\mathbf{Z}\mathbf{s} = \mathbf{0}$ where $\mathbf{Z} \stackrel{\text{def}}{=} \left(z_j^{(i)}\right)_{\substack{1 \le i \le n \\ 1 \le j \le m}}$

$\longrightarrow$ If $\mathbf{Z} \in \mathbb{F}_2^{m \times n}$ has rank $n-1$, we perform a Gaussian elimination to recover $\mathbf{s}$!

It will be verified with high probability if m large enough, $m = Cn$ for some constant $C > 0$

CONCLUSION: RUNNING TIME OF SIMON'S ALGORITHM

T be the classical running-time of f

Running time in the quantum gate model of one iteration:

- In Step 3 we apply $\mathbf{U}_f$: it can be done by using $O(T)$ quantum gates over qubits
- In Steps 2 and 5 we apply $2n$ times $\mathbf{H}$
- In Step 4 we perform a measurement on n-registers qubits: n measurements over qubits $\Big($in the computational basis$\Big)$

One iteration:

It costs quantumly $4n + O(T)$

- We repeat $O(n)$ times an iteration: it costs $O(n^2 + nT)$
- We solve a system by a classical Gaussian elimination: it costs $O(n^3)$

Overall cost in the quantum gate model:

Simon's algorithm costs $O(n^2 + n^3 + nT) = O(n^3 + nT)$

A LAST CONCLUSION

- We have solved Simon's problem in polynomial time with high probability with **only $O(n)$ queries to f** $\left(\textit{i.e.}, O(n) \text{ calls to } \mathbf{U}_f, \text{ step } 3\right)$

Is it doable classically?

- Simon has proved that any classical randomized algorithm that finds $\mathbf{s}$ with high probability needs **to make $\geq C\sqrt{2^n}$ queries to f** where C constant

$\longrightarrow$ **Quantum computing provides an exponential advantage!**

There are many results about the query complexity of quantum algorithms

- *Ronald de Wolf's lecture notes, Chapters* 11-12.

`https://arxiv.org/pdf/1907.09415.pdf`

A LAST CONCLUSION

- We have solved Simon's problem in polynomial time with high probability with **only $O(n)$ queries to f** $\left(\textit{i.e.}, O(n) \text{ calls to } \mathbf{U}_f, \text{ step } 3\right)$

Is it doable classically?

- Simon has proved that any classical randomized algorithm that finds **s** with high probability needs **to make $\geq C\sqrt{2^n}$ queries to f** where C constant

$\longrightarrow$ **Quantum computing provides an exponential advantage!**

There are many results about the query complexity of quantum algorithms

- *Ronald de Wolf's lecture notes, Chapters* 11-12.

https://arxiv.org/pdf/1907.09415.pdf

But one may say that solving Simon's problem is useless. . .

Simon's algorithm has been "the starting point" of Shor's algorithm that quantumly breaks all current deployed public-key cryptography

$\longrightarrow$ Come at Lecture 6!

EXERCISE SESSION