Quantum Computation
Introduction to quantum algorithms

Luís Soares Barbosa & Renato Neves

MSc Physics Engineering

Universidade do Minho, 2024-25

Physics of information

Information

is encoded in the state of a physical system

Computation

is carried out on an actual physically realizable device

- the study of information and computation cannot ignore the underlying physical processes.
- ... although progress in Computer Science has been made by abstracting from the physical reality
- more precisely: by building more and more abstract models of a sort of reality, i.e. a way of understanding it
- ... until now ...

Physics of information

How physics constrains our ability to use and manipulate information?

- Landauer's principle (1961): information deleting is necessarily a dissipative process.
- Charles Bennett (1973): any computation can be performed in a reversible way, and so with no dissipation.

NAND $\Longrightarrow$ Toffoli

$(x, y) \mapsto \neg(x \wedge y)$ $\quad$ $(x, y, z) \mapsto (x, y, z \oplus (x \wedge y))$

with $z = 1$

Physics of information

Information is physical, and the physical reality is quantum mechanical:

How does quantum theory shed light on the nature of information?

- Quantum dynamics is truly random
- Acquiring information about a physical system disturbs its state (which is related to quantum randomness)
- Noncommuting observables cannot simultaneously have precisely defined values: the uncertainty principle
- Quantum information cannot be copied with perfect fidelity: the no-cloning theorem (Wootters, Zurek, Dieks, 1982)
- Quantum information is encoded in nonlocal correlations between the different parts of a physical system, i.e. the predictions of quantum mechanics cannot be reproduced by any local hidden variable theory (John Bell, 1967)

Quantum computation

The meaning of computable remains the same

A classical computer can simulate a quantum computer to arbitrarily good accuracy.

... but the order of complexity may change

However, simulation is computationally hard, i.e. extremely inefficient as the number of qubits increases:

- For 100 qubits the state space would require to store $2^{100} \approx 10^{30}$ complex numbers!
- And what about rotating a vector in a vector space of dimension 10^{30}?

Quantum computation

In a sense this might not be a decisive argument:

Simulating the evolution of a vector in an exponentially large space can be done locally through a probabilistic classical algorithm in which each qubit has a value at each time step, and each quantum gate can act on the qubits in various possible ways, one of which is selected as determined by a (pseudo)-random number generator.

... After all, the computation provides a means of assigning probabilities to all the possible outcomes of the final measurement...

Quantum computation

However, Bell's result precludes such a simulation: there is no local probabilistic algorithm that can reproduce the conclusions of quantum mechanics.

In the presence of entanglement, one can access only an exponentially small amount of information by looking at each subsystem separately.

Quantum computing as using quantum reality as a computational resource

Richard Feynman, *Simulating Physics with Computers* (1982)

Non deterministic computation

... can be represented by oriented graph (often call a transition system), each node standing for a computational state and edges representing transitions from a state to another.
Globally, the computational dynamics is encoded in the graph's adjacency matrix, a Boolean matrix M where $M_{i,j} = 1$ stands for a transition from j to i. The next state is computed by matrix multiplication.

$$MS = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$(MS)_i = \sum_{k=0}^{5} M_{i,k} S_k$$

Non deterministic computation

Exercise

Discuss how this model captures non deterministic and weighted transitions. If weights correspond to tokens or costs, then one step computation $(MS)_i$ computes the number of tokens (resp., the global cost) that will reach node i in the next time click as the sum of all tokens (resp., costs) that are (resp., label) currently in the nodes connected to i.

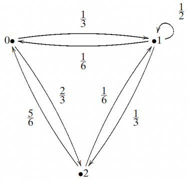
Exercise

How does multi-step computation proceed?

Probabilistic computation

States: Given a set of possible configurations, states are vectors of probabilities in $\mathbb{R}^n$ which express indeterminacy about the exact physical configuration, e.g. $\begin{bmatrix} p_0 \cdots p_n \end{bmatrix}^T$ st $\sum_i p_1 = 1$
Dymanics: double stochastic matrix (*must come (go) from (to) somewhere*), where $M_{i,j}$ specifies the probability of evolution from configuration j to i

Example:



$$\begin{bmatrix} 0 & \frac{1}{6} & \frac{5}{6} \\ \frac{1}{3} & \frac{1}{2} & \frac{1}{6} \\ \frac{2}{3} & \frac{1}{3} & 0 \end{bmatrix}$$

Probabilistic computation

Evolution: computed through matrix multiplication with a vector $|u\rangle$ of current probabilities

- $M|u\rangle$ (next state)

$$MS = \begin{bmatrix} 0 & \frac{1}{6} & \frac{5}{6} \\ \frac{1}{3} & \frac{1}{2} & \frac{1}{6} \\ \frac{2}{3} & \frac{1}{3} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{6} \\ \frac{1}{6} \\ \frac{2}{3} \end{bmatrix} = \begin{bmatrix} \frac{21}{36} \\ \frac{9}{36} \\ \frac{6}{36} \end{bmatrix}$$

  i.e. it node 1 is the current node with $\frac{1}{6}$ probability, it will remain so after a computational step with $\frac{9}{36}$ probability

- Matrix M^T reverses computation, carrying us to the previous state

Measurement: the system is always in some configuration — if found in i, the new state will be a vector $|t\rangle$ st $t_j = \delta_{j,i}$

Probabilistic computation

Composition:

$$p \otimes q = \begin{bmatrix} p_1 \\ 1-p_1 \end{bmatrix} \otimes \begin{bmatrix} q_1 \\ 1-q_1 \end{bmatrix} = \begin{bmatrix} p_1 q_1 \\ p_1(1-q_1) \\ (1-p_1)q_1 \\ (1-p_1)(1-q_1) \end{bmatrix}$$

- correlated states: cannot be expressed as $p \otimes q$, e.g.

$$\begin{bmatrix} 0.5 \\ 0 \\ 0 \\ 0.5 \end{bmatrix}$$

- Different dynamics (operators) are also composed by $\otimes$ (Kronecker):

$$M \otimes N = \begin{bmatrix} M_{1,1}N & \cdots & M_{1,n}N \\ \vdots & & \vdots \\ M_{m,1}N & \cdots & M_{m,n}N \end{bmatrix}$$

Quantum computation

States

State of n-qubits encoded as a unit vector

$$v \in \underbrace{\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2}_{n \text{ times}} \cong \mathbb{C}^{2^n}$$

A vector cell is no more a real value in $[0, 1]$, but a complex c tah $|c|^2 \in [0, 1]$. This model expresses a fundamental physical concept in quantum mechanics: interference — complex numbers may *cancel* each other out when added.

Exercise

Recall this fact considering numbers $5 + 3i$ and $-3 - 2i$.

Quantum computation

Dynamics

n-qubit operation encoded as a unitary transformation

$$\mathbb{C}^{2^n} \longrightarrow \mathbb{C}^{2^n}$$

i.e. a linear map that preserves inner products, thus norms.

Recall that the norm squared of a unitary matrix forms a double stochastic one.

Quantum computation

Evolution: computed through matrix multiplication with a vector $|u\rangle$ of current amplitudes (wave function)

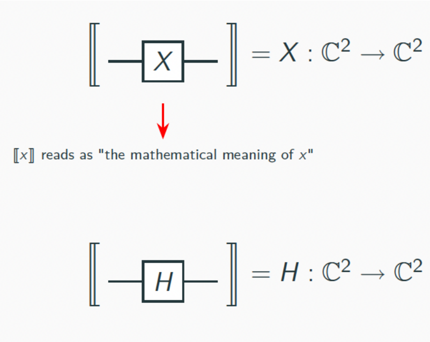
- $M|u\rangle$ (next state)

Measurement: configuration i is observed with probability $|\alpha_i|^2$ if found in i, the new state will be a vector $|t\rangle$ st $t_j = \delta_{j,i}$

Composition: also by a tensor on the complex vector space; may exist entangled states.

Basic operations

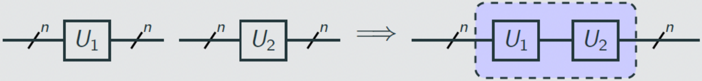
We start with a set of quantum operations, e.g.



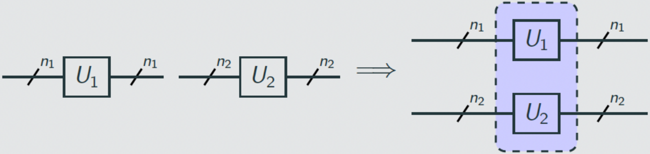
Each operation U_i manipulates the state of n_i-qubits received from its left-hand side . . . and returns the result on its right-hand side

Composition

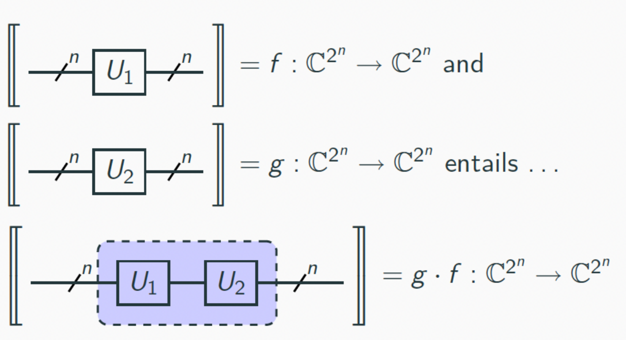
Sequential Composition



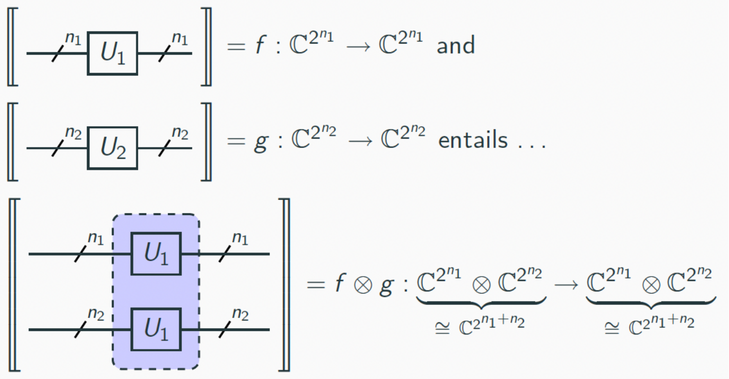
Parallel Composition



What does sequential composition mean?



What does parallel composition mean?



My first quantum algorithm

The Deutsch problem

Is $f : \mathbf{2} \longrightarrow \mathbf{2}$ constant, with a unique evaluation?

- Classically, to determine which case $f(1) = f(0)$ or $f(1) \neq f(0)$ holds requires running f twice
- Resorting to quantum computation, however, it suffices to run f once . . . due to two quantum effects superposition and interference

Turning f into a quantum operation

$f : \mathbf{2} \longrightarrow \mathbf{2}$ extends to a linear map $\mathbb{C}^2 \to \mathbb{C}^2$

... but not necessarily to a unitary transformation.

proof

The extended f does not preserve norms: Actually, when f is constant on 0 we obtain $f|0\rangle = |0\rangle$ and $f|1\rangle = |0\rangle$.
Thus,

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

However,

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

Turning f into a quantum operation

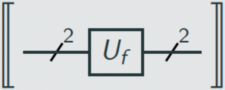
Intuition

f potentially loses information whereas pure quantum operations are reversible [Charles Bennett, 1973]

Actually, a unitary transformation is always injective so if a map loses information it cannot be unitary.

Turning f into a quantum operation

Proposed Solution



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

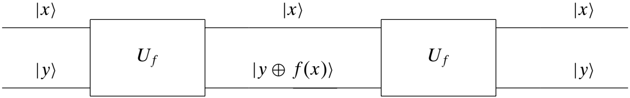
Addition modulo 2

- The oracle takes input $|x\rangle|y\rangle$ to $|x\rangle|y \oplus f(x)\rangle$
- Fixing $y = 0$ it encodes f:

$$U_f(|x\rangle \otimes |0\rangle) = |x\rangle \otimes |0 \oplus f(x)\rangle = |x\rangle \otimes |f(x)\rangle$$

Turning f into a quantum operation

- U_f is a unitary, i.e. a reversible gate

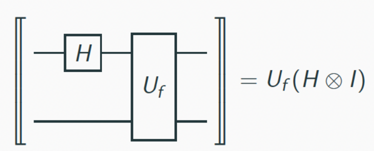


$$|x\rangle|(y \oplus f(x)) \oplus f(x)\rangle = |x\rangle|y \oplus (f(x) \oplus f(x))\rangle = |x\rangle|y \oplus 0\rangle = |x\rangle|y\rangle$$

Exploiting quantum parallelism

Can f be evaluated for $|0\rangle$ and $|1\rangle$ in one step?

Consider the following circuit



$$
\begin{aligned}
& U_f(H \otimes I)(|0\rangle \otimes |0\rangle) \\
&= U_f\left(\tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle\right) && \{\text{Defn. of } H \text{ and } I\} \\
&= U_f\left(\tfrac{1}{\sqrt{2}}(|00\rangle + |10\rangle)\right) && \{\otimes \text{ distributes over } +\} \\
&= \tfrac{1}{\sqrt{2}}(|0\rangle|0 \oplus f(0)\rangle + |1\rangle|0 \oplus f(1)\rangle) && \{\text{Defn. of } U_f\} \\
&= \underbrace{\tfrac{1}{\sqrt{2}}(|0\rangle|f(0)\rangle + |1\rangle|f(1)\rangle)}_{f(0) \text{ and } f(1) \text{ in a single run}} && \{0 \oplus x = x\}
\end{aligned}
$$

Are we done?

$$U_f(H \otimes I)(|0\rangle \otimes |0\rangle) = \underbrace{\frac{1}{\sqrt{2}}(|0\rangle|f(0)\rangle + |1\rangle|f(1)\rangle)}_{f(0) \text{ and } f(1) \text{ in a single run}}$$

NO

Although both values have been computed simultaneously, only one of them is retrieved upon measurement in the computational basis: Actually, 0 or 1 will be retrieved with identical probability (why?).

YES

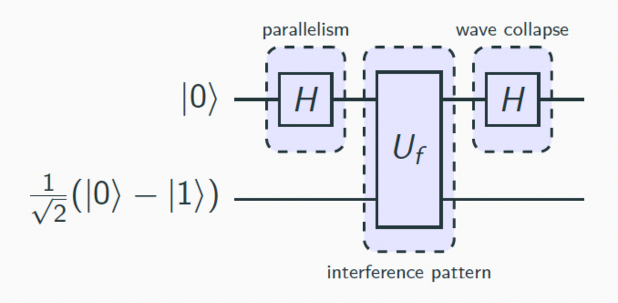
The Deutsch problem is not interested on the concrete values f may take, but on a global property of f: whether it is constant or not, technically on the value of

$$f(0) \oplus f(1)$$

Exploiting quantum parallelism and interference

Actually, the Deutsch algorithm explores another quantum resource — interference — to obtain that global information on f

Let us create an interference pattern dependent on this property, and resort to wave collapse to prepare for the expected result:



Exploiting quantum parallelism and interference

Let us start with a simple, auxiliary computation:

$$
\begin{array}{ll}
U_f\,(|x\rangle \otimes (|0\rangle - |1\rangle)) & \\
= U_f\,(|x\rangle|0\rangle - |x\rangle|1\rangle) & \{\otimes \text{ distributes over } +\} \\
= |x\rangle|0 \oplus f(x)\rangle - |x\rangle|1 \oplus f(x)\rangle & \{\text{Defn. of } f\} \\
= |x\rangle|f(x)\rangle - |x\rangle|\neg f(x)\rangle & \{0 \oplus x = x, 1 \oplus x = \neg x\} \\
= |x\rangle \otimes (|f(x)\rangle - |\neg f(x)\rangle) & \{\otimes \text{ distributes over } +\} \\
= \begin{cases} |x\rangle \otimes (|0\rangle - |1\rangle) & \text{if } f(x) = 0 \\ |x\rangle \otimes (|1\rangle - |0\rangle) & \text{if } f(x) = 1 \end{cases} & \{\text{case distinction}\}
\end{array}
$$

leading to

$$U_f\,(|x\rangle \otimes (|0\rangle - |1\rangle)) \;=\; (-1)^{f(x)}|x\rangle \otimes (|0\rangle - |1\rangle)$$

Exploiting quantum parallelism and interference

Now computing the semantics of the whole circuit leads to

$$
\begin{aligned}
& (H \otimes I)\,U_f\,(H \otimes I)\,(|0\rangle \otimes |-\rangle) \\
&= (H \otimes I)\,U_f\,(|+\rangle \otimes |-\rangle) && \{\ldots\} \\
&= \tfrac{1}{\sqrt{2}}(H \otimes I)\,U_f\,((|0\rangle + |1\rangle) \otimes |-\rangle) && \{\ldots\} \\
&= \tfrac{1}{\sqrt{2}}(H \otimes I)\,(U_f|0\rangle \otimes |-\rangle + U_f|1\rangle \otimes |-\rangle) && \{\ldots\} \\
&= \tfrac{1}{\sqrt{2}}(H \otimes I)\,\big((-1)^{f(0)}|0\rangle \otimes |-\rangle + (-1)^{f(1)}|1\rangle \otimes |-\rangle\big) && \{\text{Previous slide}\} \\
&= \begin{cases} (H \otimes I)(\pm 1)|+\rangle \otimes |-\rangle & \text{if } f(0) = f(1) \\ (H \otimes I)(\pm 1)|-\rangle \otimes |-\rangle & \text{if } f(0) \neq f(1) \end{cases} && \{\text{Case distinction}\} \\
&= \begin{cases} (\pm 1)|0\rangle \otimes |-\rangle & \text{if } f(0) = f(1) \\ (\pm 1)|1\rangle \otimes |-\rangle & \text{if } f(0) \neq f(1) \end{cases} && \{\ldots\}
\end{aligned}
$$

Lessons learnt

- A typical structure fro a quantum algorithm includes three phases:
    1. State preparation
       (fix initial setting)
    2. Transformation
       (combination of unitary transformations)
    3. Measurement
       (projection onto a basis vector associated with a measurement tool)
- This 'toy' algorithm is an illustrative simplification of the first

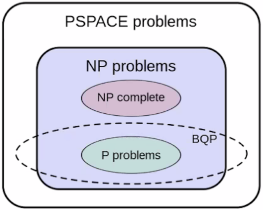
  algorithm with quantum advantage

  presented in literature [Deutsch, 1985]
- All other quantum algorithms crucially rely on similar ideas of quantum interference

Algorithms for quantum advantage

Quantum computers are conjectured to provide exponential advantage for specific computational problems.

- New complexity classes can be defined relevant to quantum computation (theory).
- Algorithmic patterns exclusive to quantum computation make the difference (practice).



(Nielsen & Chuang, 2010)

The quest for efficient quantum algorithms

The quest

- Non exponential speedup. Not relevant for the complexity debate, but shed light on what a quantum computer can do.
  Example: Grover's search of an unsorted data base.
- Exponential speedup relative to an oracle. By feeding quantum superpositions to an oracle, one can learn what is inside it with an exponential speedup.
  Example: Simon's algorithm for finding the period of a unction.
- Exponential speedup for apparently hard problems
  Example: Shor's factoring algorithm.

The quest for efficient quantum algorithms

Factoring in polynomial time - $\mathcal{O}((ln\, n)^3)$

Peter Shor, *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer* (1994)

- Classically believed to be superpolynomial in $\log n$, i.e. as n increases the worst case time grows faster than any power of $\log n$.
- The best classical algorithm requires approximately

$$e^{1.9(\sqrt[3]{\ln n}\sqrt[3]{(\ln \ln n)^2})}$$

- From the best current estimation (the 65 digit factors of a 130 digit number can be found in around one month in a massively parallel computer network) one can extrapolate that to factor a 400 digit number will take about the age of the universe (10^{10} years)

What's next?

1. Study a number of algorithmic techniques
2. and their application to the development of quantum algorithms
3. ... in between, revisit basic notions of computability and complexity