Quantum Computation: a Tutorial

Benoît Valiron

University of Pennsylvania,
Department of Computer and Information Science,
3330 Walnut Street, Philadelphia, Pennsylvania, 19104-6389, USA

benoit.valiron@monoidal.net

Received 15 April 2012

Abstract This tutorial is the first part of a series of two articles on quantum computation. In this first paper, we present the field of quantum computation from a broad perspective. We review the mathematical background and informally discuss physical implementations of quantum computers. Finally, we present the main constructions specific to quantum computation yielding algorithms.

Keywords Quantum Computation, Quantum Algorithms, Quantum Computers.

§1 Introduction

Whether the notion of data is thought of concretely or abstractly, it is usually supposed to behave classically: a piece of data is supposed to be clonable, erasable, readable as many times as needed, and is not supposed to change when left untouched.

Quantum computation is a paradigm where data can be encoded with a medium governed by the law of quantum physics. Although only rudimentary

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20168. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

quantum computers have been built so far, the laws of quantum physics are mathematically well described. It is therefore possible to try to understand the capabilities of quantum computation, and quantum information turns out to behave substantially differently from usual (classical) information.

In Sections 2, 3 and 4 of this tutorial, we describe the mathematics needed for quantum computation together with an overview of the theory of quantum computation. In Section 5, we briefly present the range of physical implementations of quantum devices. We then discuss in Section 6 three subtle algorithmic constructions specific to quantum computation that can be used in order to design algorithms able, in some case, to outperform classical algorithms on particular problems.

This paper is the first part of diptych on quantum computation. The second part[20] will be concerned with programmatic perspective on quantum computation.

§2 One quantum bit

In classical computation, the smallest unit of data is the *bit*, element of the two-element set $\{0, 1\}$. In quantum computation, the smallest unit of data is a *quantum bit*, or *qubit*, defined as a ray in a 2-dimensional Hilbert space.

2.1 Mathematical formalism

A Hilbert space is a complex vector space equipped with a notion of *length* and a notion of *orthogonality*, both defined by a scalar product. In this section, we develop the required notions for the 2-dimensional context.

Complex numbers. A complex number is of the form $a + b \cdot i$, where a and b are usual real numbers, and where i is a special symbol. Complex numbers can be added and multiplied as follows: $(a + b \cdot i) + (c + d \cdot i) = (a + c) + (b + d) \cdot i$ and $(a + b \cdot i)(c + d \cdot i) = (ac - bd) + (ad + bc) \cdot i$. The symbol i has therefore the property $i^2 = -1$. Given a complex number $\alpha = a + b \cdot i$, the *conjugate* of α is the complex number $\bar{\alpha} = a - b \cdot i$. The *norm* of α is $|\alpha| = a^2 + b^2$, also equal to $\bar{\alpha}\alpha$.

A complex number $a + b \cdot i$ can be seen as a point in the *complex plane* with coordinates (a, b). One can therefore propose an alternative representation for complex numbers using polar coordinates: the complex $\rho e^{\phi i}$ corresponds to $\rho \cos(\phi) + \rho \sin(\phi) \cdot i$. If the polar representation of complex numbers does not play

well with addition, it supports multiplication: $(\rho_1 e^{\phi_1 i})(\rho_2 e^{\phi_2 i}) = (\rho_1\rho_2)e^{(\phi_1+\phi_2)i}$ and conjugation: $\overline{\rho e^{\phi i}} = \rho e^{-\phi i}$. The angle ϕ is called the *phase* and ρ the *amplitude* of the complex number.

2-dimensional Hilbert space. The set of column vectors $\binom{\alpha}{\beta}$ where α and β are complex numbers can be equipped with a structure of vector space. If $u = \binom{\alpha}{\beta}$ and $v = \binom{\gamma}{\delta}$, then $u + v = \binom{\alpha+\gamma}{\beta+\delta}$ and $\gamma u = \binom{\gamma\alpha}{\gamma\beta}$. The *scalar product* of the vectors u and v is the operation $\langle u|v\rangle = \bar{\alpha}\gamma + \bar{\beta}\delta$. It can also be seen as the multiplication of the row-vector $u^* = (\bar{\alpha}\ \bar{\beta})$ with the column vector v (u^* is called the *conjugate transpose* of u): see Figure 1(d). If $\langle u|v\rangle = 0$, we say that u and v are *orthogonal*. For example, $\binom{1}{0}$ and $\binom{0}{1}$ are orthogonal; so are $\binom{1}{1}$ and $\binom{1}{-1}$. The scalar product induces a *norm*: $||u||^2 = \langle u|u\rangle$. A *normalized vector* is a vector of norm 1. A *basis* is a pair of vectors u and v that can generates the whole space when linearly combined. The basis is *orthonormal* if u and v are normalized and orthogonal. For example, $\binom{1}{0}$ and $\binom{0}{1}$ form an orthonormal basis. So do $\frac{1}{\sqrt{2}}\binom{1}{1}$ and $\frac{1}{\sqrt{2}}\binom{1}{-1}$.

Ray. A ray is an equivalence class of vectors stable by (non-zero) scalar multiplication. So $\binom{\alpha}{\beta}$ is in the same ray as $\binom{-\alpha}{-\beta}$ and $\binom{2\alpha}{2\beta}$. A corollary is that for every non-zero vector $\binom{\alpha}{\beta}$ it is possible to find a normalized vector whose first coordinate is a non-negative real number: if $\alpha = \rho_1 e^{\phi_1 i}$ and $\beta = \rho_2 e^{\phi_2 i}$, choose $\frac{1}{\rho_1^2+\rho_2^2}\binom{\rho_1}{\rho_2 e^{(\phi_2-\phi_1)i}}$.

Unitary map. Regarding quantum computation, a particularly interesting operation on Hilbert space is the *unitary map*. It is simply a rotation, i.e. a change of orthonormal basis, and it can be efficiently represented by a 2×2 matrix: if $\binom{1}{0}$ is sent to $\binom{\alpha_1}{\beta_1}$ and $\binom{0}{1}$ is sent to $\binom{\alpha_2}{\beta_2}$, the unitary map is $U = \binom{\alpha_1\ \alpha_2}{\beta_1\ \beta_2}$. The transformation U on a particular vector is the application of the matrix onto that vector, and the composition of two unitaries is matrix multiplication (see Figure 1).

2.2 The quantum bit as vector of information

A quantum bit is merely a vector $u = \binom{\alpha}{\beta}$ in the 2-dimensional Hilbert space. In order to make computational sense out of it, we choose the orthonormal basis $\binom{1}{0}, \binom{0}{1}$ that we write $|0\rangle, |1\rangle$. The vector u can then be seen as the quantum superposition of the two classical boolean values true and false: $\alpha|0\rangle + \beta|1\rangle$. The two values $|0\rangle$ and $|1\rangle$ are orthogonal to each other.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x & y \\ z & t \end{pmatrix} = \begin{pmatrix} ax+bz & ay+bt \\ cx+dz & cy+dt \end{pmatrix}$$
(a) matrix-matrix

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ z \end{pmatrix} = \begin{pmatrix} ax+bz \\ cx+dz \end{pmatrix}$$
(b) matrix-column vector

$$\begin{pmatrix} a \\ c \end{pmatrix} \begin{pmatrix} x & y \end{pmatrix} = \begin{pmatrix} ax & ay \\ cx & cy \end{pmatrix}$$
(c) column vector-row vector

$$\begin{pmatrix} a & b \end{pmatrix} \begin{pmatrix} x \\ z \end{pmatrix} = \begin{pmatrix} ax+bz \end{pmatrix}$$
(d) row vector-column vector

Fig. 1 Matrix multiplication

We will use the convention that $|x\rangle, |y\rangle, |z\rangle \dots$ refer to base states while greek letters: $|\phi\rangle, |\psi\rangle, \dots$ refer to general quantum bits.

2.3 Operations

Since a quantum system comes equipped with a preferred basis $|0\rangle, |1\rangle$, the operations are described in this basis.

The first class of operations we can perform is the *creation* of a quantum bit out of nothing. Both $|0\rangle$ and $|1\rangle$ can be created at wish. The second class of allowed operations consists in unitaries, i.e. *reversible* operations. Beside the identity I, some of the usual gates are the not-gate, the Hadamard, the phase shift and the phase-flip:

$$N = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad V_\theta = \begin{pmatrix} 1 & 0 \\ 0 & e^{\theta i} \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

The gate N sends $|0\rangle$ to $|1\rangle$ and $|1\rangle$ to $|0\rangle$: it effectively acts as a negation. The Hadamard gate creates quantum superposition: it sends $|0\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. The gate V_θ does not change the vector $|0\rangle$ but sends $|1\rangle$ to $e^{\theta i}|1\rangle$. Z is just V_π.

Unitaries are only rotating the state of the quantum system. In order to get some classical information out, the only available operation is the *measurement*. It is a *probabilistic* operation defined as follows: if $u = \binom{\alpha}{\beta}$ is a normalized vector representing a quantum bit, the measure of u returns true with probability $|\alpha|^2$ and false with probability $|\beta|^2$. Moreover, the state of the quantum bit was modified by the measure: the quantum bit is in state $\binom{1}{0}$ if the result of the measure was true and $\binom{0}{1}$ is the result was false. We say that the measurement was performed *against the basis* $\binom{1}{0}, \binom{0}{1}$. In general, given a basis u_0, u_1, a *measurement of* v *against* u_0, u_1 returns true with probability $|\langle u_1|v\rangle|^2$ and false with probability $|\langle u_2|v\rangle|^2$. It turned v into u_1 if the measurement returned true and into u_2 if it returned false.

Note that if we are physically restricted to measurements against the base $\binom{1}{0}, \binom{0}{1}$, it is still possible to simulate a measurement against an arbitrary basis u_0, u_1 by first applying the unitary sending u_0 to $\binom{1}{0}$ and u_1 to $\binom{0}{1}$ on the vector under consideration, then measuring, then applying the unitary sending $\binom{1}{0}$ and $\binom{0}{1}$ respectively back to u_0 and u_1.

Together with one quantum bit, one can already build a simple experiment: the coin-toss. Create a quantum bit in state $|0\rangle$, apply the Hadamard gate to change the state of the qubit to $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, the measure in the canonical basis: we get true and false with equal probability.

Destructive measurements. A measurement can equivalently be thought of as *destroying* the quantum bit we were considering. In the physical realization with photons, this is what happen for example. It is not incompatible with the first approach since one can always re-create a quantum bit in the canonical basis to simulate a non-destructive measure.

2.4 The Bloch sphere

The rays of $\mathbb{C}^2$ are in one-to-one correspondence with the points on the unit sphere of $\mathbb{R}^3$. In spherical coordinates, a point (θ, φ) with $\theta \in [0, \pi], \psi \in [-\pi, \pi]$ is in correspondence with the ray of representative

$$r(\theta, \varphi) = \begin{pmatrix} \cos\frac{\theta}{2} \\ e^{i\varphi}\sin\frac{\theta}{2} \end{pmatrix}. \tag{1}$$

The correspondence is pictured in Figure 2: index s indicates spherical coordinates; index c indicates 3-dimensional coordinate; index $\mathcal{H}$ indicates coordinates in $\mathbb{C}^2$. This unit sphere is called the *Bloch sphere*.

Using the Bloch sphere, one can define three canonical "orthogonal" bases for a quantum bit, as follows.

- $(|0_z\rangle, |1_z\rangle) = (|0\rangle, |1\rangle)$ is the *basis along* z;
- $(|0_x\rangle, |1_x\rangle) = (\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \frac{1}{\sqrt{2}}(|1\rangle - |1\rangle))$ is the *basis along* x;
- $(|0_y\rangle, |1_y\rangle) = (\frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle), \frac{1}{\sqrt{2}}(|1\rangle - i|1\rangle))$ is the *basis along* y.

These three bases have a peculiar property: measuring $|0_z\rangle$ and $|1_z\rangle$ against both the bases along x and y returns true and false with probability $\frac{1}{2}$: in other words, two quantum bits in state $|0_z\rangle$ and in state $|1_z\rangle$ cannot be distinguished by measurement against the basis x and y. This property is also true for the qubits $|0_x\rangle$ and $|1_x\rangle$ when measured against the bases along y and z, and for the

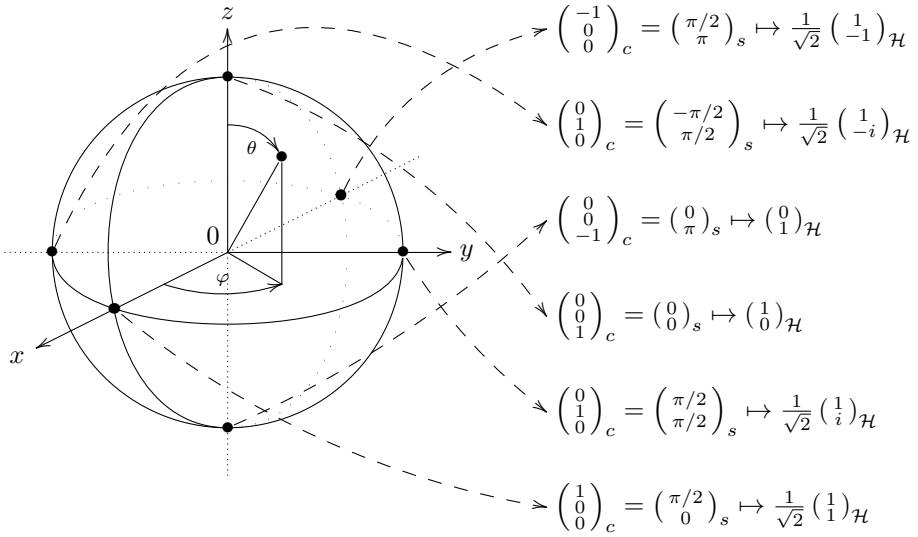


Fig. 2 The Bloch sphere.

qubits $|0_y\rangle$ and $|1_y\rangle$ measured against the bases along x and z.

In general, two rays in $\mathbb{C}^2$ are orthogonal if and only if their representations on the Bloch sphere are antipodal.

2.5 The BB84 algorithm

We can use two of the bases described in Section 2.4 to securely create secret keys for one-time pad encryption. This algorithm, called BB84 from the seminal paper that presented it[2)]. The algorithm reads as follows.

- Alice creates two sequences of n random bits. the first sequence specifies which of the basis x or z to use to encode each of the bit in the second sequence.
- Alice sends the encoded quantum bits to Bob
- Bob creates a sequence of n random bits: he measures each of the quantum bits he receive using the basis specified by its sequence of bits.
- On a public classical channel (e.g. internet), Alice and Bob share the basis for creation and measurement of quantum bits they used for each quantum bits. They keep only the bits in the sequence where the bases match: This is the shared secret key.
- If the quantum channel was noisy, or if an eavedropper tampered with it, the photons Bob received might not precisely match the one Alice

sent: This introduces errors in Bob's measurements. Using successive exchanges of parity information (called *reconciliation phase*[3)]), Alice and Bob can recover the complete bit-string.

- If the error-rate is low enough (about 15%), it is then possible to produce a shorter, but secret key using *privacy amplification*[4)]. If the error-rate is too high, they just start over.

This protocol is provably secure: on average, an evedropper cannot get more than a certain percentage of the secret key.

While quantum cryptography is not the subject of this tutorial, it is nonetheless interesting to note that the technology is well-developed and mature enough for companies to sell quantum-based cryptographic devices[16, 9)]. Commercial devices that might not be as secure as their theoretical counterpart[15)]...

§3 Two quantum bits

If one quantum bit is enough to build the foundation for a cryptographic system, it is not sufficient to perform general computation. In this section, we add a second quantum bit to our system

The state of a 2-qubit system lives in the *tensor product* $\mathbb{C}^2 \otimes \mathbb{C}^2$ of the two original spaces $\mathbb{C}^2$. The space $\mathbb{C}^2$ was generated by the base $\binom{1}{0}, \binom{0}{1}$, the space $\mathbb{C}^2 \otimes \mathbb{C}^2 = \mathbb{C}^4$ is generated by

$$\binom{1}{0} \otimes \binom{1}{0} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \binom{0}{1} \otimes \binom{1}{0} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad \binom{1}{0} \otimes \binom{0}{1} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad \binom{0}{1} \otimes \binom{0}{1} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

The vectors are respectively shorten in $|00\rangle, |01\rangle, |10\rangle, |11\rangle$: the space $\mathbb{C}^2 \otimes \mathbb{C}^2$ can be seen as the space of "quantum superpositions" of pairs of booleans.

3.1 Operations

The operations one can perform on 2-quantum bit system are quite similar to the one performed on only one quantum bit: One can create, measure, and apply unitary operations.

Unitary operations. As for one-qubit system, it is possible to change a quantum system through discrete, reversible operations that simply sends an orthogonal basis to another orthonormal basis. The simplest solution to construct a unitary operation on a 2-qubit system is to tensor two 1-qubit operations: $U \otimes V$ applied on $|x\rangle \otimes |y\rangle$ is $(U|x\rangle) \otimes (V|y\rangle)$. For example, the *mixing* operation is $H^{\otimes 2} = H \otimes H$: it sends $|00\rangle$ to $(H|0\rangle) \otimes (H|0\rangle) = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$. But

this is (thankfully) not the only possibility: when written in the canonical basis $|00\rangle, |01\rangle, |10\rangle, |11\rangle$, we have in particular the swap gate X and the control-not gate $\mathbf{N}_C$ defined by

$$X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad N_C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

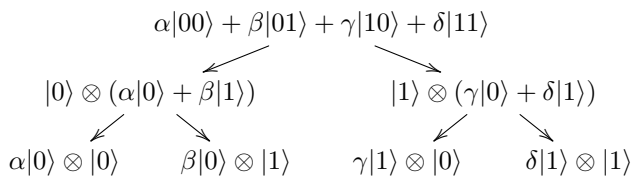
The N_C-gate comes from a generic process: If U is a one-qubit gate, one can construct $U_C = \left(\begin{smallmatrix} I & 0 \\ 0 & U \end{smallmatrix}\right)$, called *controlled* U, as the 2-qubit gate:

$$U_C\left(|0\rangle \otimes |\phi\rangle\right) = |0\rangle \otimes |\phi\rangle,$$

$$U_C\left(|1\rangle \otimes |\phi\rangle\right) = |1\rangle \otimes U|\phi\rangle.$$

The gate U is applied on the second qubit depending on the value of the first qubit.

Measurements. Measuring a 2-qubit system is similar as doing so in a 1-qubit system. And despite the probabilistic nature of the operation, measuring first the first qubit and then the second qubit is equivalent to doing it the other way around. If the system is in state $|\phi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$, measuring the first qubit then the second yields (modulo renormalization):



The norm of each of these vectors is the overall probability of getting to this state: from $|\phi\rangle$, in two steps we get (true,true) and the state $|00\rangle$ with probability $|\alpha|^2$; in one step, we get false and the state $|1\rangle \otimes (\gamma|0\rangle + \delta|1\rangle)$ with probability $|\gamma|^2 + |\delta|^2$.

3.2 Particular properties

In this section, we discuss various specific properties of quantum computation when more than one quantum bit are available.

No Cloning. Consider a quantum bit in some unknown state $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$; it is not possible to "clone" the state and get the two-qubit state $|\phi\rangle \otimes |\phi\rangle$. The reason is simple: the only operations we are allowed to perform are unitaries and

measurements, which are essentially linear maps. The cloning operation being non-linear, it is therefore not implementable.

It is however possible to build a "copying" map G

$$\alpha|0\rangle + \beta|1\rangle \longmapsto \alpha|00\rangle + \beta|11\rangle.$$

But it does not satisfy the usual commutativity rule for duplication: if f is any linear map $\mathbb{C}^2 \to \mathbb{C}^2$, then $G \circ f \neq (f \otimes f) \circ G$.

Entanglement. Another special property of quantum information is *superposition*: the 2-qubit state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is a valid state, but cannot be written as $|\phi\rangle \otimes |\psi\rangle$. We say the two quantum bits are *entangled*.

The interesting aspect of entanglement is that the measurements of the qubits are correlated. For example, measuring the first qubit in the state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ with respect to the standard basis yields 0 with probability $\frac{1}{2}$ and 1 with probability $\frac{1}{2}$. In the former case, the state of the 2-qubit system is $|00\rangle$, thus measuring the second qubit yields 0 with probability 1. In the latter, the state of the system is $|11\rangle$, and measuring the second quantum bit yields 1 with probability 1.

One can form a basis for $\mathbb{C}^2 \otimes \mathbb{C}^2$ out of entangled states. For example, an often used basis is the following set of orthogonal, entangled states:

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \quad \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle), \quad \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \quad \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle).$$

Probabilistic and quantum computation. From a classical perspective, a measurement is a special form of coin-toss: Quantum computation can therefore simulate probabilistic computation. Is quantum computation conservative over probabilistic computation? It turns out not to be. In the following, we propose a proof can be found using the protocol described by Bell in 1964[1)], yielding what is known as *Bell's inequalities*.

The argument goes as follows. Consider a quantum machine that maximally entangles two quantum bits A and B

$$|\phi_{AB}\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |1\rangle - |1\rangle \otimes |0\rangle)$$

and sends qubit A to Alice and qubit B to Bob. Suppose that they can independently choose one of the following axes in the Bloch sphere (see Figure 2) to measure:

$$a = (0, 0, 1), \qquad b = \left(\frac{\sqrt{3}}{2}, 0, -\frac{1}{2}\right), \qquad c = \left(-\frac{\sqrt{3}}{2}, 0, -\frac{1}{2}\right).$$

They live in the xz-plane of the Bloch sphere and correspond respectively to the bases in $\mathbb{C}^2$:

$$|0_a\rangle = |0\rangle, \qquad |0_b\rangle = \tfrac{1}{2}|0\rangle + \tfrac{\sqrt{3}}{2}|1\rangle, \qquad |0_c\rangle = \tfrac{1}{2}|0\rangle - \tfrac{\sqrt{3}}{2}|1\rangle,$$

$$|1_a\rangle = |1\rangle \qquad |1_b\rangle = \tfrac{\sqrt{3}}{2}|0\rangle - \tfrac{1}{2}|1\rangle, \qquad |1_c\rangle = \tfrac{\sqrt{3}}{2}|0\rangle + \tfrac{1}{2}|1\rangle.$$

What is the probability of obtaining the same output when measuring A and B with respect to two different bases?

If we forget about the quantum aspect of the protocol, we essentially build two isolated probabilistic computations inputting an element from the set $\{a, b, c\}$ and outputting a probabilistic boolean.

If we stay purely probabilistic, the result of measuring A and B along each of the possible axis is predetermined. Let $P_{x,y}$ be the probability of obtaining the same output while measuring A along x and B along y. We have

$$P_{a,b} + P_{b,c} + P_{c,a} \geqslant 1, \tag{2}$$

since for any possible distribution of measurement values, there will always be two values that will be equal. However, using the definition of measure against arbitrary basis given in Section 2.3,

$$P_{x,y} = |\langle \phi_{AB} | 0_x 0_y \rangle|^2 + |\langle \phi_{AB} | 1_x 1_y \rangle|^2.$$

The computation shows that $P_{x,y} = \frac{1}{4}$ for $x \neq y$, $x, y = a, b, c$. In particular,

$$P_{a,b} + P_{b,c} + P_{c,a} = \tfrac{3}{4} < 1,$$

which violates Equation (2): Quantum computation is not conservative over probabilistic computation.

§4 Many quantum bits

If the state of a 2-qubit system lives in $\mathbb{C}^2 \otimes \mathbb{C}^2$, not so surprisingly the state of a 3-qubit system lives in $\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$, that is, $\mathbb{C}^8$, and its canonical basis in lexicographic order is $|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle$. Working with a n-quantum bit system means manipulating vectors in $\mathbb{C}^{2^n}$, with a basis made of strings of n booleans.

Measurements and unitaries extend fluently to the case of n quantum bits. However, since the size of matrices becomes prohibitive, and since most unitary operations are built out of smaller one, we prefer to write operations on quantum bits using *quantum circuits*. At first sight, quantum circuit is to

quantum computation what classical boolean circuit are to classical computation: the description of an algorithm. However, it should be noted that unlike classical circuit, for most physical implementation one cannot just "get from a shelf" a component of a quantum circuit: the circuit is really a description of logical operations to be performed on the system, and should more be seen as the description of a procedure.

4.1 Universal set of gates

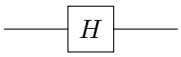
An interesting result coming from the theory of Lie algebras is that there exist sets of unitary operations from which one can construct (using composition and tensor) any given unitary gate up to a given error. These sets are called *universal sets of gates*. Some examples are

- the set of all the unitary gates on $\mathbb{C}^4$;
- the set of all the unitary gates on $\mathbb{C}^2$ together with N_C;
- the set consisting of H, N_C and $V_{\frac{\pi}{4}}$.

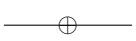
In particular, this means that for every unitary map on n qubits one can find a quantum circuit using H, N_C and $V_{\frac{\pi}{4}}$ that approximates it.

4.2 Quantum circuits

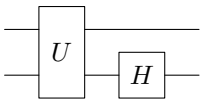
A quantum circuit is a graphical representation of a sequence of unitary operations to be performed on quantum bits. Graphically, a quantum bit is a wire, and a unitary is a (named) box. Several quantum bits are represented by several parallel wires, read from top to bottom. For example, the circuit



represents a one-quantum-bit system on which one applies the Hadamard gate. Two 1-qubit operations have special representation: the not-gate N and the identity function are respectively denoted with

 and .

The notation for the gate N is due to its resemblance with the classical XOR operation. The circuit

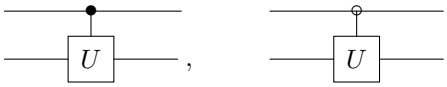


is a two-quantum-bit system on which one first apply the two-quantum-bit unitary U, followed by the Hadamard gate on the second quantum bit. The unitary map represented by this circuit is the map $|\phi\rangle \longmapsto (I \otimes H)(U|\phi\rangle)$.

Two classes of gates have a particular representation and deserve special attention. The first one is the class of swap gates. As we already saw in Section 3.1, the swap X on 2 qubits, refers to a "physical" swap. One therefore represents swaps as follows:

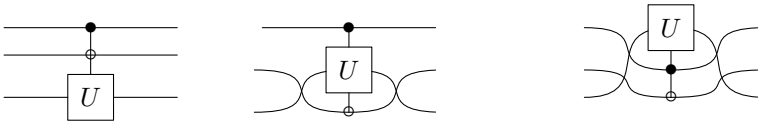
The former is X and sends $|x\rangle \otimes |y\rangle$ to $|y\rangle \otimes |x\rangle$, while the latter sends $|x\rangle \otimes |y\rangle \otimes |z\rangle$ to $|z\rangle \otimes |x\rangle \otimes |y\rangle$.

The second kind of gates to have a special representation in quantum circuits is the controlled gate. Since this gate acts conditionally on a quantum bit, we write respectively



for the gate U_C sending $|0\rangle \otimes |y\rangle$ to $|0\rangle \otimes |y\rangle$ and $|1\rangle \otimes |y\rangle$ to $|1\rangle \otimes (U|y\rangle)$ and for the gate $(N \otimes I)U_C(N \otimes I)$ sending $|0\rangle \otimes |y\rangle$ to $|0\rangle \otimes (U|y\rangle)$ and $|1\rangle \otimes |y\rangle$ to $|1\rangle \otimes |y\rangle$.

The notation is flexible. For example, the circuits



are all describing the map sending the basis element $|1\rangle \otimes |0\rangle \otimes |z\rangle$ to the state $|1\rangle \otimes |0\rangle \otimes (U|z\rangle)$ and every other basis element to itself.

Creation and measurement. It is often useful to express creation of quantum bits and measurements. The creation of a quantum bit in state $|\phi\rangle$ and the measurement, making a classical bit out of a quantum bit, are respectively represented with the notations

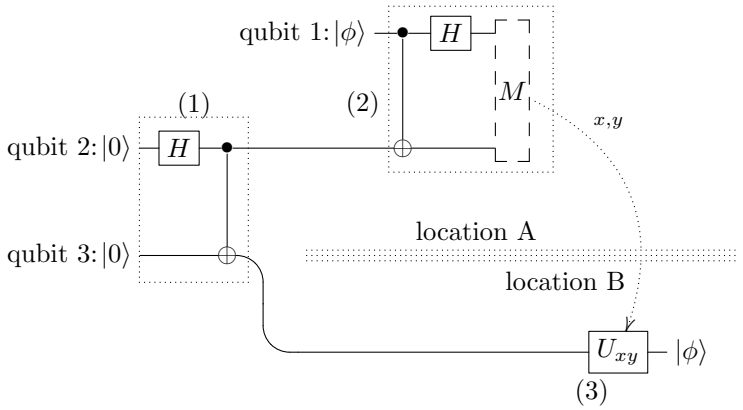
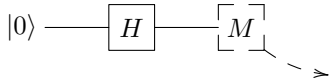


Fig. 3 Quantum teleportation protocol.

So for example, the coin-toss is represented by the quantum circuit



4.3 A simple algorithm: quantum teleportation

A standard example of algorithm that mix unitaries and measurements operations is the so-called quantum teleportation algorithm. It allows one to send the state of an unknown quantum bit from a location A to a location B without having to look at it (i.e. without having to measure it). The trick is to use an entangled pair of quantum bits between both locations. See Figure 3; we explain the 3 steps:

1. At location A, create the initial entangled state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ with the two last qubits. Alice keeps qubit 2 and stays at location A, while Bob takes qubit 3 and goes to location B.
2. Alice, to send qubit 1 in state $|\phi\rangle$ to Bob, applies a rotation on her two qubits 1 and 2. She then measures the 2-qubit resulting state, gets two classical bits (x, y) and sends them to Bob.
3. Provided that M outputs the bits x, y, to transform qubit 3 to state $|\phi\rangle$, Bob applies on it the transformation U_{xy}, where U_{00}, U_{01}, U_{10} and U_{11} are respectively $\left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$, $\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, $\left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$.

Note that the entanglement of qubits 2 and 3 can be done ahead of time (so long as they stay entangled).

Proof of the correctness of the protocol. If we apply the computation of Step (2) to the two first qubits of

$$(\alpha|0\rangle + \beta|1\rangle) \otimes \tfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \tfrac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle)$$

we get

$$\tfrac{1}{2}\Big(\alpha(|000\rangle + |100\rangle) + \alpha(|011\rangle + |111\rangle) + \beta(|010\rangle - |110\rangle) + \beta(|001\rangle - |101\rangle)\Big)$$

$$= \frac{1}{2}\begin{pmatrix} & |00\rangle \otimes (\alpha|0\rangle + \beta|1\rangle) & + & |01\rangle \otimes (\beta|0\rangle + \alpha|1\rangle) \\ + & |10\rangle \otimes (\alpha|0\rangle - \beta|1\rangle) & + & |11\rangle \otimes (\alpha|1\rangle - \beta|0\rangle) \end{pmatrix}$$

Measuring the two first qubits, the third qubit becomes

$$\begin{array}{ll} \alpha|0\rangle + \beta|1\rangle & \text{if 00 was measured,} \\ \beta|0\rangle + \alpha|1\rangle & \text{if 01 was measured,} \\ \alpha|0\rangle - \beta|1\rangle & \text{if 10 was measured,} \\ \alpha|1\rangle - \beta|0\rangle & \text{if 11 was measured.} \end{array}$$

Finally, if U_{xy} is applied in the case where x, y was the result of the measurement, then the state of the last qubit is $\alpha|0\rangle + \beta|1\rangle$, as desired. □

§5 Physical realization of quantum devices

Nowadays, more or less every physical realization of a computer uses silicon and transistors. In the realm of quantum computers, several competing physical implementations co-exist, both with their strengths and weaknesses. In this section, we briefly review the (wide) range of existing physical implementations of quantum devices.

5.1 Decoherence and errors

To encode a quantum bit, one needs to find a physical object with two modes that are both distinguishable by a physical operation and whose state can be regarded as being governed by the law of quantum physics. For example, a book can be either opened or closed, but one can hardly argue that these two modes can be placed into "quantum superposition".

However, for various physical objects, one can find such modes. Some simple examples are photons: polarization, or location; trapped ions: electronic state; electrons: spin nuclei: spin; in general: energy level, if discrete.

The main problem with an object governed by the law of quantum physics is that it tends to interact with its environment. The state of the particle

will therefore change along time, a phenomenon known as *decoherence*. Depending on the technology, the typical decoherence time for an quantum bit ranges from 1ms to several tens of seconds[14].

Together with imperfect gates, decoherence makes quantum information extremely unstable. It renders the task of building a device hosting quantum information very challenging.

5.2 Requirements

Requirements for a successful physical implementation are well summarized by DiVincenzo[8] and Ladd and al[14].

Scalability. Regardless of the physical apparatus, to be able to take advantage of the efficiency of a given quantum algorithm, it has to be applied on a large enough input size. This implies that the number of quantum bits is large enough. It also implies that one needs to be able to apply probably very many operations on the quantum bits in a single run: we need the implementation to scale both in space and in time, and to be highly parallel.

Set of universal gates. Any physical implementation will come with a set of allowed operations on the device. Sure enough, we need to be able to allocate and initialize quantum bits in a fixed state, and we need to be able to measure the quantum bits with respect to a particular basis. But also, we need to be able to construct an arbitrary unitary gate on a potentially arbitrary number of quantum bits: a universal set of elementary gates is required for a successful quantum device.

Timing issue: low error rate, fast gates and quantum bit stability. The algorithms require many gates to be applied on many quantum bits. In order to keep the probability of error low enough, the various gates need to be precise enough. Since the run-time of interesting algorithms will probably be non-trivial, the state of the quantum bits need to be stable enough in time and gates to be applied fast both to allow many gates to be applied and to allow quantum error-correction to succeed.

Ability to move quantum bits around. Finally, the quantum bits are represented by the state of physical objects: 2-qubit operations (such as a control-gate, for example) on non-adjacent quantum bits will generally require either moving the physical objects near each other, or moving the quantum state themselves to adjacent physical objects (e.g. by teleportation, or a series of swap operations).

The bottom line is that to realize a quantum computer one has to con-

ciliate two rather opposite properties. On one hand, quantum bits have to be isolated from the outer world; on the other hand, they have to interact with each other, and with the devices actually performing the computation.

5.3 Review of techniques

In the following, we list a few of the physical realizations of quantum devices, and discuss their relative advantages and drawbacks. For an in-depth, technical description, see Chapter 7 of Nielsen and Chuang's classic textbook[19], or the more recent book from Chen and al.[6]. For an rapid overview of the current state of the field, refer to Ladd and al[14].

Photons. Photons are reliable supports for quantum bits: they are relatively free from decoherence, and they are easy to move around. There are several ways to encode a quantum bit on a photon. The first one is simply by using polarization to hold the state of the quantum bit: vertical polarization means $|0\rangle$, horizontal means $|1\rangle$. But one can do other ways, for example by the location of the photon. Depending on how the paths are designed and how the photon is injected, the photon could experience one path or the other, or both at the same time, creating a superposition of the state $|0\rangle$ and the state $|1\rangle$.

Some experimental devices used to interact with photons are known tools from linear optics:

- Lasers. At very low intensity, they can produce photons one at a time.
- Polarizing beamsplitters. Separate incoming photons into separate paths, depending on their polarization state. Combined with a pair of single-photon detectors, the polarization state of a qubit can be measured.
- Mirrors. They are used to change the direction of a photon in space.
- Polarization rotators. It can be realized by a piece of birefringent material: the deeper the piece is, the bigger the phase shift is.
- (Regular) beamsplitter: A thin layer of semi-refracting material. It can be used for either coupling two polarization modes, or for separating a photon on two distinct paths.

Photons are therefore good candidates for moving as well as holding quantum information: they are relatively resilient to decoherence, they are easily routed, and many easy-to-deploy devices exist for performing one-qubit operations on them. Although the main caveat remains the interaction between two photons, recent progress[13, 14] tends to indicate how to overcome the prob-

lem. This makes quantum optics a potentially promising setting for a quantum computer.

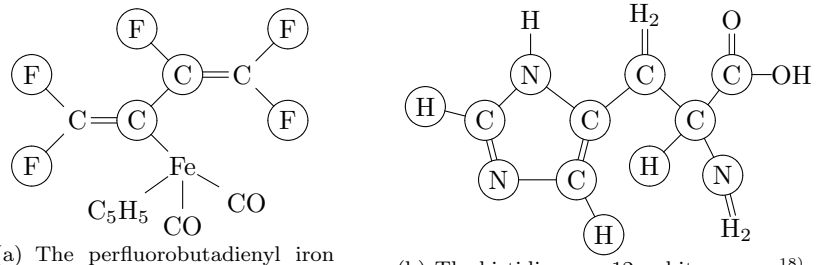
Trapped ions. Ions can be trapped in free space using suitable electric fields. Provided that the ions are cooled enough, they are relatively resistant to decoherence, and their decoherence time is several orders of magnitude longer than the time needed for the basic operation of initialization, unitary maps and measurement. This makes them good candidates as support for quantum computation. Various research groups[5] have experimented with using this technique to build quantum memories with a number of quantum bits ranging from 4 (in 2000)[5] to 14 (in 2011[17]).

Ions are arranged spatially as an array. Initialization to base state is realized by laser cooling. Interaction on single ion is performed using lasers pulses, and entanglement can be done either by local interaction between ions, or through photonic interaction. This has the advantage of allowing quantum entanglement between potentially far-away quantum bits, and hints at how to scale a quantum computation realized in this way. However, although at small scale one can keep a high-fidelity control, it remains to show how to scale this fidelity to many qubits.

Quantum dots. Instead of confining quantum particles in free space, it is possible to store them on a solid host, such as a semiconductor, a piece of silicon or a crystal. The binding onto the host can typically be done electrostatically. The advantage is that the quantum "dot", for example an electron, or a single impurity such as an atom, does not need to be "trapped" in free space, but is instead integrated once and for all into the structure and is arguably more stable. It is also potentially able to support room-temperature without hampering the coherence time.

The vast variety of supporting hosts and dots makes this technique very versatile and subject to extremely active research. Due to its wide range of application, it is also able to bring insight to other techniques.

Nuclei of atoms. A naturally well-isolated system is the nucleus of an atom: shielded behind many electronic layers, the nuclear spin of an atom is extremely stable. It can then be reached and acted upon using an electromagnetic field, a technique called nuclear magnetic resonance (NMR). The technique is the following: a liquid at room temperature containing a particular molecule at



(a) The perfluorobutadienyl iron complex as a 7-qubit memory[21].

(b) The histidine as a 12-qubit memory[18].

Fig. 4 Molecules as quantum memory

some concentration is designed. The molecule is chosen so that some of its atoms can be either independently addressed or entangled with other using a specific electromagnetic pulse. The molecule can be thought of as a quantum memory of a certain (usually small) size, the last record being 12 quantum bits[18] in 2006: see Figure4(b) for the molecule. The atoms used as memory registers are circled. Since the liquid contains many of these molecules, the computation is run in parallel on all of them; this partially solves the problem of the probabilistic nature of quantum computation: only one run is necessary.

A previous record[21] was a famous experiment where Shor's factoring algorithm has been successfully applied to factor... the number 15. The reason for the small input has to do with the small size of the quantum memory: the "computer" had only 7 quantum bits available, and Shor's algorithm requires quite a few auxiliary qubits (see Figure 4(a) for the molecule, the circled atoms forming the memory). However, the fact that a complete algorithm was implemented is a noteworthy milestone: albeit small, the apparatus can reasonably be called a quantum computer.

One of the problem in this liquid-state NMR technique is the problem of the initialization of the memory. The other problem lies in the fact that a particular molecule has to be chosen for holding the memory: the method is not really scalable.

Superconductor. As photons, electrons are good support for quantum information. The naïve hardware where they are found, that is, electrical circuit, does not form a good support for holding them, though: because of resistive power, decoherence is extremely high.

There is however a state of the matter where resistivity is minimized: superconductors. And indeed, quantum computing have been experimented using

this technology[14].

To date, the commercially most successful paradigm of computation using superconducting quantum computation is called adiabatic quantum computation. Although this is out of the scope of this review, it is worth noting that the company D-Wave[7] sells a quantum device with 128 quantum bits, encoded using superconducting techniques. Unfortunately, their device is not universal. But it is working, and computationally expressive enough to perform discrete optimization. Plus, it is no longer a lab experiment: it is a product that you can actually purchase (for a whopping 10 millions dollars).

§6 A tour of quantum algorithms

What can we do with quantum information, and what makes an algorithm manipulating quantum information really quantum?

As we'll see in Section 6.1, classical circuits can be efficiently simulated by quantum circuits: in particular, there is no particular "gain" in choosing quantum computation in this situation. The real gain is when the algorithm makes use of either entanglement, or more generally the interference brought by complex coefficients.

Most of the existing "really" quantum algorithms are based on a few quantum constructs described in Sections 6.2 to 6.4: quantum Fourier transform, quantum walk and amplitude amplification. The rest if made of classical pre and post-analysis, and possibly of an oracle: a quantum circuit corresponding to a classical reversible operation (Section 6.1).

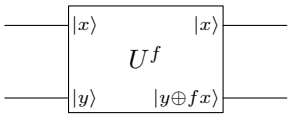
Despite the fact that only three main constructions are available, the richness of their possibilities makes the search for quantum algorithms and optimizations over classical algorithms a vibrant area: the *quantum algorithm zoo* of S. Jordan[10] refers 45 algorithms and 160 papers with no less than 14 written between 2011 and 2012.

6.1 Reversible classical computation

A restricted subset of unitaries (i.e. N, N_C, N_{CC}, X, ...) sends basis elements to basis elements: quantum circuits built from these components are effectively classical, boolean computation. What is the power of quantum computation with respect to classical computation?

Although quantum circuit are reversible operations, it turns out that one can simulate classical, boolean computation with quantum computation,

only using N_C and N_{CC}. If $\mathbb{B}$ is the set of booleans values, and if the map $f : \mathbb{B}^m \to \mathbb{B}^n$ is a boolean function taking m arguments and outputting a n-tuple of booleans, the map $\hat{f} : \mathbb{B}^m \times \mathbb{B}^n \to \mathbb{B}^m \times \mathbb{B}^n$ sending (x, y) to $(x, y \oplus (f\, x))$ is reversible. It is possible to build a quantum circuit

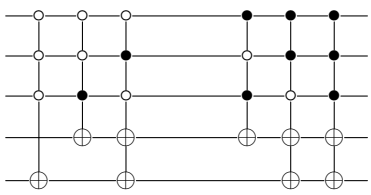


that is effectively computing the operation $\hat{f}$, sending

$$\textstyle\sum_i \alpha_i |x_i\rangle \otimes |y_i\rangle \longmapsto \sum_i \alpha_i |x_1\rangle \otimes |y_i \oplus f(x_i)\rangle.$$

A dumb implementation. The simplest implementation one could think of uses a series of multi-controlled operations implementing the truth table. For example, for a function $f : \mathbb{B}^3 \to \mathbb{B}^2$, a possible truth table and its circuit implementation is as follows.

0	0	0	0	1	1	1	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	1
1	0	1	0	0	0	1	1



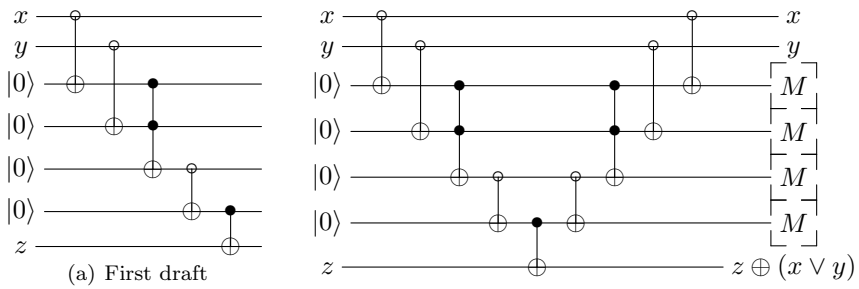
The circuit has one controlled-gate per non-zero output.

A compositional implementation. Although this encoding soundly encodes the operation $\hat{f}$, it is not really efficient: the number of gates is exponential on the number of inputs. It is possible to do a better, efficient implementation of $\hat{f}$, compositionally on the definition of f.

If f is the "and" map $\mathbb{B}^2 \to \mathbb{B}$, the operation $\hat{f}$, with 3 inputs and 3 outputs is simply the gate N_{CC} sending $|xyz\rangle$ to $|xy\rangle \otimes |z \oplus xy\rangle$. If f is the "not" map, the operation $\hat{f}$, with 2 inputs and 2 outputs is simply the gate $(N \otimes I)N_C(N \otimes I)$ sending $|xy\rangle$ to $|x\rangle \otimes |(\text{not } x) \oplus y\rangle$.

If the map f is $f(x, y) = \mathit{not}\ ((\mathit{not}\ x)\ \mathit{and}\ (\mathit{not}\ y))$, the operation $\hat{f}$ can be constructed in term of the elementary circuits $\hat{not}$ and $\hat{and}$. A first draft of circuit is as in Figure 5(a).

The problem with this circuit is that it has 3 input wires (x, y and z) and 7 output wires. One has to "close" the wires starting with $|0\rangle$. The



(a) First draft (b) Final draft

Fig. 5 Implementation of the "or" operation

obvious candidates for this is the measurement operation: it makes sure that the quantum bits that are measured are not anymore in superposition with the rest of the system. However, before actually performing the measure, one has to "undo" the operation we did on these wires. Indeed, we are working with quantum bits and not mere classical booleans. Suppose that the state of the system x, y, z are $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \otimes |0\rangle$. By linearity, since $|0000000\rangle \mapsto |0011100\rangle$ and $|1100000\rangle \mapsto |1100011\rangle$, then

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \otimes |0000\rangle \otimes |0\rangle \mapsto \frac{1}{\sqrt{2}}(|0011100\rangle + |1100011\rangle).$$

The quantum bits 3 to 6 in the output are not in a base state: measuring them will send the whole system to $|0011100\rangle$ with probability $\frac{1}{2}$ and to $|1100011\rangle$ with probability $\frac{1}{2}$. This is not what we were looking for: the overall circuit would send $\frac{1}{\sqrt{2}}(|000\rangle + |110\rangle)$ to a probabilistic distribution of states $|000\rangle$ and $|111\rangle$.

Thanks to the unitarity of the elementary operations, it is possible to "undo" the intermediate computations performed on the inner quantum bits. The actual $\hat{f}$ we want is the circuit of Figure 5(b). Now, if x, y, z were in state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \otimes |0\rangle$, right before the measurement the system is in the state

$$\frac{1}{\sqrt{2}}(|000000\rangle + |1100001\rangle).$$

Measuring (and forgetting) the inner quantum bits (number 3, 4, 5 and 6), we get the state $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$, which is precisely the expected result.

Of course, in the case of the implementation of the "or" function, it is easy to come up with a more resource-caring circuit. For example, the implementation in term of truth table is probably more efficient. But if you consider more sophisticated computation such as an adder, even this simplistic compositional implementation quickly becomes more interesting than the truth-table

one.

Control structures and size of circuits. Unlike usual, classical computation, classical simulation using quantum circuit has some drawbacks.

The first one is that all classical loops have to be completely unwinded: there is no "jump" operation, as the circuit only goes from left to right. This makes quantum circuits similar to classical boolean circuits.

However, quantum circuits are even more constrained than classical, boolean circuits: there is no true branching on test. Since tests are implemented with controlled gates, both branches of the circuit will have to be computed.

Technically, this means that *every single gates* in a quantum circuit will be run, making a relatively large overhead to the transformation classical-to-quantum.

6.2 Quantum phase estimation

Certain interesting algorithms in algebra and number theory reduce to the problem of order finding. For example, factorization is such a problem and the acclaimed Shor's factorization algorithm is using order finding as its core component. Some other examples are the search for a discrete log and the general hidden subgroup problem.

In classical mathematics, the order finding problem is not known to have an efficient algorithm. Thus, in general we do not associates the aforementioned problems with it. However, in quantum computation there exists such an algorithm: this is why we focus on it.

Order finding. The problem reads as follows: choose two coprime integers a and N. A theorem states that there exists a number $p > 0$ such that $a^p = 1 \bmod N$. What is p?

Interestingly enough, this problem relates to quantum computation because the property that unitaries, as matrices, can be decomposed into

$$U = \sum_j \lambda_j u_j u_j^*.$$

where λ_j's are complex numbers and u_j's are normalized, orthogonal column vectors. As we saw in Section 2.1 and in Figure 1, $Uu_j = \lambda_j u_j$: we call u_j an *eigenvector* of U and λ_j a *eigenvalue*.

To see how one can relate this notion to order finding, assume for simplicity that $N = 2^n$ and consider the operation U_a on n qubits sending $|j\rangle$ to $|j \cdot a \bmod N\rangle$. This operator is unitary since a and N are coprime: the image of $\{0 \dots N-1\}$ under $j \mapsto j \cdot a$ is the whole set. In particular, since U_a^k sends $|j\rangle$ to $|j \cdot a^k \bmod N\rangle$, if $a^p = 1 \bmod N$ then the map U_a^p is the identity map. Therefore, the eigenvalues of U_a are p^{th} roots of the unity. A p^{th} root of unity is of the form $e^{2\pi\, ik/p}$.

Thus, an algorithm for eigenvalue estimation should be enough to retrieve p (with some classical post-processing): we left the world of number theory and we are back with operators: This is the realm of quantum computation.

Quantum phase estimation. Before solving the problem of estimating an eigenvalue, we first discuss a similar problem: quantum phase estimation. the question is to estimate the parameter ω in the state $|\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i\omega y}|y\rangle$.

The trick consists in decomposing ω as a floating point number in binary notation. Note that without loss of generality, one can assume that $0 \leqslant \omega < 1$ since $e^{2\pi\, i} = 1$. Then write ω as $\omega = \sum_{i=1}^{\infty} \frac{x_i}{2^i}$ where the x_i's are 0 or 1. We write $\omega = 0.x_1x_2x_3\dots$, assuming a binary representation.

First, suppose that ω is equal to $0.x_1$. In this case, $n = 1$ and

$$|\phi\rangle = \tfrac{1}{\sqrt{2}} \textstyle\sum_{y=0}^{1} e^{\pi i(x_1 y)}|y\rangle,$$

that is, $|\phi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_1}|1\rangle)$. To retrieve the value x_1, it is enough to use the Hadamard gate, since $H|\phi\rangle = |x_1\rangle$.

If now we are interested in $\omega = 0.x_1x_2$, then $\omega = \frac{x_1}{2} + \frac{x_2}{2^2}$ and $y = 2$. The state $|\phi\rangle$ is

$$|\phi\rangle = \frac{1}{\sqrt{2^2}} \sum_{y=0}^{3} e^{2\pi\, i(\frac{x_1}{2} + \frac{x_2}{2^2})y}|y\rangle = \frac{1}{2} \sum_{y=0}^{3} e^{\frac{2}{2^2}\pi\, i(2x_1 y)} e^{\frac{2}{2^2}\pi\, i(x_2 y)}|y\rangle.$$

Remembering that $e^{2\pi\, i} = 1$, we can rewrite this sum as

$$|\phi\rangle = \frac{1}{2}(|00\rangle + e^{\frac{1}{2}\pi\, i(2x_1)} e^{\frac{1}{2}\pi\, ix_2}|01\rangle + e^{\frac{1}{2}\pi\, i(2x_2)}|10\rangle + e^{\frac{1}{2}\pi\, i(2x_1)} e^{\frac{1}{2}\pi\, i(3x_2)}|11\rangle)$$

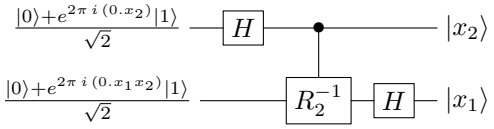
This factors as

$$|\phi\rangle = \frac{1}{2}(|0\rangle + e^{\frac{1}{2}\pi\, i(2x_2)}|1\rangle) \otimes (|0\rangle + e^{\frac{1}{2}\pi\, i(2x_1)} e^{\frac{1}{2}\pi\, ix_2}|1\rangle)$$

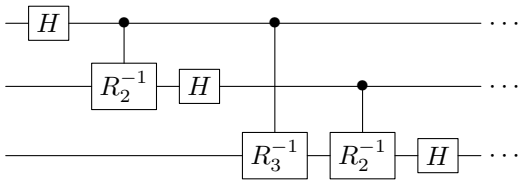
and it rewrites to

$$|\phi\rangle = \frac{1}{2}(|0\rangle + e^{2\pi\, i\,(0.x_2)}|1\rangle) \otimes (|0\rangle + e^{2\pi\, i\,(0.x_1x_2)}|1\rangle) \tag{3}$$

Remember that $H\frac{2}{\sqrt{2}}(|0\rangle + e^{2\pi\, i\,(0.x_2)}|1\rangle) = |x_2\rangle$. If we write R_2 for $\left(\begin{smallmatrix} 1 & 0 \\ 0 & e^{2\pi i(0.01)} \end{smallmatrix}\right)$, from $|\phi\rangle$ we can retrieve the values x_1 and x_2 with the circuit



This scheme generalizes, and if R_n is $\left(\begin{smallmatrix} 1 & 0 \\ 0 & e^{2\pi i(0.0\ldots01)} \end{smallmatrix}\right)$ and $\omega = 0.x_1 \ldots x_n$, then the circuit



computes $|x_n \ldots x_1\rangle$ out of $|\phi\rangle$.

Quantum Fourier transform. The above general circuit computes the phase estimation, denoted QFT^{-1}: $\frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1} e^{2\pi i \frac{x}{2^n}y}|y\rangle \mapsto |x\rangle$. The reverse operation is written QFT:

$$|x\rangle \longmapsto \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \frac{x}{2^n}y}|y\rangle$$

and is called *quantum Fourier transform* because of its resemblance to the discrete Fourier transform. It is realized by inverting the circuit for the phase estimation.

Estimation of eigenvalues. Suppose that the eigenvalue of an eigenvector for a specific unitary maps is of the form $e^{2\pi i\omega}$. We are now ready to estimate the value ω.

Suppose again that $\omega = 0.x_1x_2$. Note that $2 \cdot 0.x_1x_2 = x_1.x_2$ and that $e^{2\pi\, ix_1.x_2} = e^{2\pi\, i0.x_2}$. We have

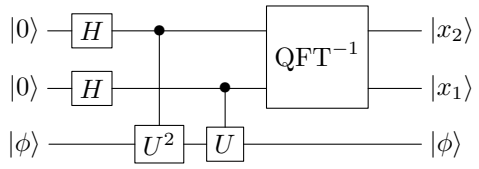
$$(U_C)^k \left(\tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |\phi\rangle\right) = \tfrac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(k\omega)}|1\rangle) \otimes |\phi\rangle.$$

In particular,

$$U_C\left(\tfrac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\otimes|\phi\rangle\right)=\tfrac{1}{\sqrt{2}}(|0\rangle+e^{2\pi i(0.x_1x_2)}|1\rangle)\otimes|\phi\rangle.$$

$$(U_C)^2\left(\tfrac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\otimes|\phi\rangle\right)=\tfrac{1}{\sqrt{2}}(|0\rangle+e^{2\pi i(0.x_2)}|1\rangle)\otimes|\phi\rangle.$$

Using Equation (3) and the map QFT^{-1}, we solve the problem:



This circuit generalizes for arbitrary ω (albeit probabilistically). Adding more $(U_C)^k$ to the circuit allows the estimation of more digits of ω.

6.3 Quantum walk

Another useful construction is the notion of *quantum walk*. It is similar to the random walk: instead of flipping a classical coin that creates a probabilistic superposition, we flip a "quantum coin" creating quantum superposition instead. To illustrate the difference, let us define a random walk, and then its quantum counterpart.

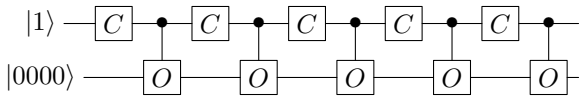
Consider a ring of 16 connected nodes as follows:

$$\begin{array}{ccccccccccccccc}
0000 & \rightarrow & 0001 & \rightarrow & 0010 & \rightarrow & 0011 & \rightarrow & 0100 & \rightarrow & 0101 & \rightarrow & 0110 & \rightarrow & 0111 \\
\uparrow & & & & & & & & & & & & & & \downarrow \\
1000 & \leftarrow & 1001 & \leftarrow & 1010 & \leftarrow & 1011 & \leftarrow & 1100 & \leftarrow & 1101 & \leftarrow & 1110 & \leftarrow & 1111
\end{array}$$

Random walk. Start from the node 0000. At each step, toss a fair coin: if tail, do not move, else move to the next node. The probability distributions over the 5 first steps shows a wave of probabilistic values moving right and getting diluted along the way:

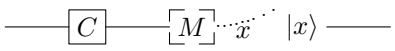
0000	0001	0010	0011	0100	0101	0110	...
1	0	0	0	0	0	0	...
0.5	0.5	0	0	0	0	0	...
0.25	0.5	0.25	0	0	0	0	...
0.125	0.375	0.375	0.125	0	0	0	...
0.0625	0.25	0.375	0.25	0.0625	0	0	...
0.03125	0.15625	0.311	0.311	0.15625	0.03125	0	...

Quantum walk. A quantum walk[12] is performed similarly, except that everything is in superposition, including the coin. The state of the computation consists then of 4 quantum bits for the spacial location and one quantum bit for the coin. The evolution is unitary: the coin-toss is a one-qubit unitary matrix C and the spacial step-forward O is the map sending $|i\rangle$ to $|i+1\rangle$ is $i \neq 1111$ and $|1111\rangle$ to $|0000\rangle$. This operation is unitary: it sends a base to a base, and it is obviously reversible. Also note that is is a purely classical operation: it is an oracle (which is the reason why we choose O for its name), and one can therefore implements it using the technique of Section 6.1. A sequence of 5 steps is represented by the circuit



To retrieve a classical information, as usual one simply measures the output.

What is a "fair" coin in this situation? If in the "quantum coin" computation we were measuring the quantum bit right after C



(4)

then we would recover a classical, random walk: a reasonable notion of quantum fair coin is a unitary matrix C such that the quantum coin (4) is a usual fair coin. An example of such a fair coin is the Hadamard gate $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. After 5 steps the probability distribution over the space is described by

0000	0001	0010	0011	0100	0101	0110	...
0.03125	0.15625	0.125	0.125	0.53125	0.03125	0	...

This time the wave keeps its cohesion and the tip of the wave goes faster than its probabilistic counterpart.

It has to do with the choice of the quantum coin and its initial value. Intuitively, the idea is that the Hadamard gate tends to "mix", or "annihilate" the tail of the wave but strengthen the step-forward part of the computation.

If instead we had chosen the initial value $|0\rangle$ for the coin, the tail of the wave would have been preserved instead. An indeed, the computation of the probability distribution after 5 steps gives

0000	0001	0010	0011	0100	0101	0110	...
0.03125	0.53125	0.125	0.125	0.15625	0.03125	0	...

Use in algorithms. The quantum walk is a tool that can be used to explore a graph. A made-up example where a quantum walk is more efficient than a classical algorithm is the *binary welded tree*: two binary trees of the same height are joined by a random welding. Starting at the root of one tree and not knowing the welding, can you find an algorithm that will find the root of the other tree? A random walk can of course be used, but as for the simple example we saw, the probability gets diluted very fast. It is possible to do better with a quantum walk, using the fact that interference can "cancel-out" some nodes and concentrates the wave function on the exit node.

Of course, the algorithm is not off-the-shelf: the step function (the oracle), the coin and the number of steps need to be fine-tuned. However, various algorithms successfully make use of the quantum walk[10].

6.4 Amplitude amplification

A technique related to quantum walks is called amplitude amplification. As for quantum walks, interference plays a central role in increasing the amplitudes of the states we are interested in and lowering the amplitudes of the other ones.

Grover's algorithm. The older and most known algorithm based on this technique is due to Grover, and aim at answering the search problem: given a boolean function of n inputs and one output, find an input to the function whose image is 1. We assume that there are not so many such inputs. This problem can be applied to a wide variety of problem. In this paragraph, we follow the very clear explanation given by Kaye and al.[11].

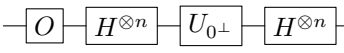
The tools for the algorithm are 3 unitary maps operating on n qubits: an oracle O, computing $O(|x\rangle) = (-1)^{f(x)}|x\rangle$; a n-shift operator $U_{0^\perp}$ sending $|00\ldots0\rangle$ to itself and all other base states $|x\rangle$ to $-|x\rangle$; the mixing operator $H^{\otimes n}$, applying the Hadamard gate on all n qubits.

Note that O is not of the canonical form $U_f(|x\rangle \otimes |y\rangle) = |x\rangle \otimes |y \oplus f(x)\rangle$. But from U_f we can construct O easily: since

$$U_f(|x\rangle \otimes \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle)) = (-1)^{f(x)}|x\rangle \otimes \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

we can omit the last register and consider O as the map acting on the n first qubits. The algorithm is a succession of the *Grover iterates* G

where G is



We know the action of O. What is the action of $H^{\otimes n}U_{0^\perp}H^{\otimes n}$? Consider the maximally entangled state $|\phi\rangle = H^{\otimes n}|00\ldots0\rangle = \sum_{i=0}^{2^n-1}|i\rangle$, and let $V_{|\phi\rangle}$ be the one-dimensional space generated by $|\phi\rangle$ and $V_{|\phi\rangle^\perp}$ its orthogonal subspace. Since HH is the identity,

$$H^{\otimes n}U_{0^\perp}H^{\otimes n}|\phi\rangle = H^{\otimes n}U_{0^\perp}|00\ldots0\rangle = H^{\otimes n}|00\ldots0\rangle = |\phi\rangle.$$

Now, sure enough for any element $|\psi\rangle$ in $V_{|\phi\rangle^\perp}$, $H^{\otimes n}|\psi\rangle$ is a superposition of base states which does not contain $|00\ldots0\rangle$. Therefore $U_{0^\perp}(H^{\otimes n}|\psi\rangle) = -H^{\otimes n}|\psi\rangle$, and $H^{\otimes n}U_{0^\perp}H^{\otimes n}|\psi\rangle = -|\psi\rangle$. So really, $H^{\otimes n}U_{0^\perp}H^{\otimes n}$ could be written as $U_{\phi^\perp}$, to keep the same intuition as the notation for $U_{0^\perp}$.

We are now ready to see the intuition behind the algorithm. Let us decompose it up to the first iteration. The very first step is to create the maximally entangled state $|\phi\rangle$. This state can be decomposed into a "good" subspace, the subspace of all the $|i\rangle$ such as $f(i) = 1$ and a "bad" subspace, the subspace of all the $|i\rangle$ such as $f(i) = 0$. We can then write

$$|\phi\rangle = |\phi_{\text{good}}\rangle + |\phi_{\text{bad}}\rangle.$$

The term $|\phi_{\text{good}}\rangle$ can be decomposed into $\epsilon|\phi\rangle + \delta|\phi^\perp\rangle$, with $|\phi^\perp\rangle$ in $V_{|\phi\rangle^\perp}$. Since there are not so many solutions to f, the amplitude of $|\phi_{\text{good}}\rangle$ is small, and so is ϵ.

Applying G to $|\phi\rangle$ means first applying O: this sends $|\phi\rangle$ to $|\phi_{\text{bad}}\rangle - |\phi_{\text{good}}\rangle$, that can be re-written as $(1-2\epsilon)|\phi\rangle - 2\delta|\phi^\perp\rangle$. Applying the operation $U_{\phi^\perp}$, the state $U_{\phi^\perp}(G|\phi\rangle)$ becomes

$$\begin{aligned} U_{\phi^\perp}((1-2\epsilon)|\phi\rangle - 2\delta|\phi^\perp\rangle) &= (1-2\epsilon)|\phi\rangle + 2\delta|\phi^\perp\rangle \\ &= (3-4\epsilon)|\phi_{\text{good}}\rangle + (1-4\epsilon)|\phi_{\text{bad}}\rangle. \end{aligned}$$

If ϵ is small enough (i.e. there are not so many i's such that $f(i) = 1$), we effectively increased the amplitude of the "good" states and decreased the amplitude of the "bad" states. Each iteration will emphasize the difference up to an optimal number of iterations.

In simple cases, this optimal can be computed exactly; in general, it can only be approximated: again, the classical information we can retrieve from this technique is probabilistic.

Use in algorithms. For many algorithms such as the NP-complete ones, the only available classical algorithm is a brute-force search. With amplitude amplification, one can gain quadratic speedup on a (classical) brute-force search algorithm.

6.5 Size of circuits

It should be noted that for non-trivial size of inputs, the number of gates in the circuits can quickly becomes daunting. If one take into account that each oracle is devised as shown in Section 6.1, the number of auxiliary quantum bit also grow very fast, yielding complex data-structures.

The fact that the complexity of the quantum algorithm is more manageable than the one of a corresponding classical algorithm is only saying that there exists an input size for which the quantum algorithm outperform the classical algorithm. In the event of an actual quantum computer with a finite memory, a thorough resource estimation needs to be performed, and heavy code optimization needs to be performed.

Quantum programming language may bring a benefit to this problem.

§7 Conclusion

We are nowhere close to real, complete, scalable quantum computers. However, many experimentations are realized in a very wide spectrum of approaches, letting us dream for the possibility of a programmable device at the edge between the classical and the quantum world.

Despite the strange nature of quantum information and its probabilistic interaction with the classical world, various quantum algorithms make it possible to compute differently and, in some case, theoretically faster than classical algorithms.

This is where we close this first part of the tutorial. The second and last part will present quantum computation from a programmer's perspective: why we need quantum programming languages in the first place, what are the challenged and what is the state of the art.

Acknowledgment We would like to thank Thomas Chapuran for his careful proof-reading of Section 5 and his helpful comments.

References

1) Bell, J. S.. "On the Einstein Podolsky Rosen paradox," *Physics, 1* pp. 195–200, 1964.

2) Bennett, C. H. and Brassard, G., "Quantum cryptography: Public key distribution and coin tossing," in *Proc. of the IEEE Int. Conf. on Computers, Systems, and Signal Processing*, pp. 175–179, 1984.

3) Bennett, C. H. et al., "Experimental quantum cryptography," *J. Cryptology, 5, 1*, pp. 3–28, 1992.

4) Bennett, C. H. et al. "Generalized privacy amplification," *IEEE Transactions on Information Theory, 41, 6*, pp. 1915–1923, 1995.

5) Blatt, R. and Wineland, D., "Entangled states of trapped atomic ions," *Nature, 453*, pp. 1008–1015, 2008.

6) Chen, G. et al. *Quantum Computing Devices: Principles, Designs, and Analysis*, Chapman and Hall/CRC, 2007.

7) D-Wave, `http://www.dwavesys.com/`, electronic resource.

8) DiVincenzo, D. P., "The physical implementation of quantum computation," *Fortschr. Phys, 48* pp. 771–783, 2000.

9) IDQ, `http://www.idquantique.com/`, electronic resource.

10) Jordan, S., `http://math.nist.gov/quantum/zoo/`, electronic resource.

11) Kaye, P., Laflamme, R. Mosca, M., *An Introduction to Quantum Computing*, Oxford University Press, 2007.

12) Kempe, J., "Quantum random walks: An introductory overview," *Contemporary Physics, 44, 4* pp. 307–327, 2003.

13) Knill, E., Laflamme, R., and Milburn, G. J., "A scheme for efficient quantum computation with linear optics," *Nature, 409*, pp. 46–52, 2001.

14) Ladd, T. D. et al., "Quantum computers," *Nature, 464*, pp. 45–53, 2010.

15) Lydersen, L. et al., "Hacking commercial quantum cryptography systems by tailored bright illumination," *Nature Photonics*, 2010.

16) magiQ, `http://www.magiqtech.com/MagiQ/Home.html`, electronic resource.

17) Monz, T. et al. "14-qubit entanglement: Creation and coherence," *Phys. Rev. Lett., 106*, p. 130506, 2011.

18) Negrevergne, C. et al., "Benchmarking quantum control methods on a 12-qubit system," *Phys. Rev. Lett., 96*, p. 170501, 2006.

19) Nielsen, M. A. and Chuang, I. L., *Quantum Computation and Quantum Information*, Cambridge University Press, 2002.

20) Valiron, B., "Quantum computation: from a programmer's perspective," *New Generation Computing*, 2012. To appear.

21) Vandersypen L. M. K.et al. "Experimental realization of shor's quantum factoring algorithm using nuclear magnetic resonance," *Nature, 414* pp. 883–887, 2001.