Prog Artif Intell (2015) 3:89–105
DOI 10.1007/s13748-014-0059-0

REGULAR PAPER

Can artificial intelligence benefit from quantum computing?

Vicente Moret-Bonillo

Received: 10 April 2014 / Accepted: 14 August 2014 / Published online: 13 September 2014
© Springer-Verlag Berlin Heidelberg 2014

Abstract In this article, we will try to present from an academic point of view some of the relevant characteristics of both artificial intelligence (AI) and quantum computing (QC) in order to explore the possibility of a meaningful cooperation between both areas of computer science (CS). The quantum part of this paper is based on "The Quantum Circuit Model" which, in the opinion of the author, could be easier understood by computer scientists and/or artificial intelligence researchers, rather than other approaches such as, for example, "Adiabatic Quantum Computation". Many fundamental questions will arise along this paper for which we will need to give an answer in order to analyze the basic principles that allow researchers and engineers to put AI and QC working together. With this in mind, we will briefly describe the behavior of the biological brain focusing on the identification of the singularities that allow them to perform in such an efficient manner. Energy consumption and parallelism are in the core of the above-mentioned efficiency. Then we will present some of the well-established artificial intelligence approaches that are potentially related to the operation of biological brains. After identifying common characteristics we will introduce basic concepts and issues related to the so-called reversible computing and QC that may eventually help to increase the efficiency of our current intelligent systems. In this respect, we will pay special attention to 'speed' and 'energy consumption'. Some examples, as well as an outline of algorithms from both quantum computing and artificial intelligence, will be used to illustrate the ideas presented in the paper. We conclude with a discussion about what can we expect from the cooperation between both, apparently unconnected fields of computer science, artificial intelligence and quantum computing.

Keywords Artificial intelligence · Reversible computing · Quantum computing

1 Introduction

In 2003 ACM published an article authored by Peter Shor entitled: "Why haven't more quantum algorithms been found?" [26]. Among the possible answers to this crucial question, Shor pointed out that perhaps one of the reasons is that computer scientists do not usually understand the basics of quantum mechanics [32]. This last statement equally applies to both, computer scientists and artificial intelligence practitioners. In fact Feynman used to call the field of artificial intelligence as the "Field of Advanced Computing Applications" [14]. However, it appears that we are "filling the gap" and continuous research in quantum computing (QC), computer science (CS), and artificial intelligence (AI), is starting to produce promising results that are suitable for a deep analysis. And what is even more encouraging is that there is a real interest in the development of this relatively new multidisciplinary field.

1.1 What is the situation in 2014?

Following the seminal efforts of IBM and Bell Labs and their long investments during many years in basic science, in May 2013 NASA and Google, in collaboration with a consortium of universities, set up an initiative to investigate how technology could drive advances in the artificial intelligence arena

Submitted to "Progress in Artificial Intelligence", Springer, ISSN: 2192-6352, March 2014.

V. Moret-Bonillo (✉)
Department of Computer Science, University of La Coruña,
15071 La Coruña, Spain
e-mail: vicente.moret@udc.es

[8]. The initiative is based on a synergy of interests. First, Google defends that QC could help to improve research in the Internet and in the technology of voice recognition. On the other hand, the university researchers could use QC to elaborate better and more efficient computational models. Finally, NASA could use QC to establish models of the space climate, to simulate planetary atmospheres, or to analyze enormous quantities of data.

Until very recently, and although the theoretical background was solid enough, the suitable technology that could allow to put everything together was not yet developed. The situation was quite similar to that what happened with the "Analytical Machine" defined by Babbage and Lovelace: the theory was much more advanced than the technology and Babbage and Lovelace failed [28]. On the contrary, it seems that nowadays the situation is clearly different, and although with some precaution and some serious criticisms on its true nature, the company D-Wave sold, in 2011, his first system designed and implemented following the quantum theory principles, the 128-qubit D-Wave One, to the company Lockheed Martin. And at the beginning of the year 2013 the same company D-Wave updated its first version to a 512-qubit D-Wave Two [12]. Finally, in January 2014, Google reported results comparing the performance of the D-Wave Two in the "Quantum Artificial Intelligence Lab" with that of classical computers. However, the results were ambiguous and provoked heated discussion on the Internet [24].

Perhaps Google and NASA strategy is not correct, in fact some quantum researchers do not believe D-Wave to be a real quantum computer. One can also think that Google and NASA have chosen a looser horse to start in the Quantum Artificial Intelligence race, but the fact is that such a race has already started.

1.2 A brief description of the scenario

Let us think a little bit about what it has been presented in the previous paragraphs. It is clear that not all the problems are of the same nature. Having that in mind, and in order to meet with the requirements of NASA, Google and the consortium of universities, it is necessary to have a very clear understanding of: (a) the theoretical foundations; (b) the advantages and disadvantages of the technology; (c) the problems to be resolved; and (d) the "essential limitations" of both the theory and the technology. We will try to illustrate our point of view with an example [7]. Biological brains are composed by neural systems highly interconnected. The neurons need energy and the neural systems need space. Our brain contains roughly 100 trillion nervous cells, 32 million kilometers of fibers, one million trillion connections, it occupies a volume of 1.5 l, it only weighs 1.5 kg, and it consumes around 10 W. If we try to build a similar brain with chips of silicon, our artificial brain would consume some 10 MW. Besides, the heat that would produce a brain of silicon of such characteristics would destroy soon our artificial brain. From the previously mentioned arguments the following question arises: how is it possible that a biological brain is able to work so efficiently and in such an economical way?

Obviously, and this is one of the aims of the artificial intelligence field, the objective could be the construction of machines structurally and/or functionally similar to the brain. However, in spite of the success and interest we are observing, much effort still has to be done. In the remainder of this paper we will explore some of the concepts that have been introduced in this section. Such an analysis could be of interest to better understand the potential of the synergy between Quantum Computing and Artificial Intelligence.

2 Human brains and computers

Much of the efficiency of a biological brain is related to energetic aspects. In fact, the energy cost of transmitting signals, or—in other words—the communication capability between two biological neurons, has probably been the most important factor in the biological evolution of the brain.

Approximately 50–80 % of the total energy consumption by the brain is due to conduction of action potentials through the fibers and synaptic transmission. In this regard, it is a proven fact that the transmission speed of information in today's computers is much greater than the rate of transmission of nerve impulses in biological brains. However, biological brains are designed as highly parallel networks. The majority of neurons are connected directly to thousands of other neurons. For this to be possible, the three-dimensional nature of the brain is essential [7].

On the contrary, the establishment of connections, even among a small number of silicon neurons, is limited by the two-dimensional nature of the chips and circuits. So, unlike the brain, the direct communication between the "neurons" of silicon is very restricted. It is, however, possible to exploit the high speed allowed by conventional electronics, but in a different way. The basic idea consists of using processes that make it feasible to handle multiple messages over the same cable. In this manner computer engineers can begin to emulate the connections of the biological networks. There are however practically insurmountable energetic consequences. In fact, to reduce energy and to increase speed, some computer engineers that are trying to simulate the behavior of biological systems have adopted the strategy of using analog encoding rather than digital encoding ("Neuromorphic Engineering") [23]. Thus, instead of digitally coding in classical bits, information is encoded in the analog circuitry through continuously changing voltage. Calculations can therefore be done in fewer steps that can also better exploit the basic physics of silicon devices.

In any case, the human brain behaves as a unit of massively parallel computing in which many instructions are executed simultaneously. The concept may seem trivial, and certainly works well in biological brains. The basic idea underlying is that often complicated problems can be divided into simpler problems that are then solved simultaneously (in parallel). The problem, however, is not as simple to our conventional computers. The reason is that the power consumption of a chip is given by the following equation: $P = C \times V^2 \times F$, where P is power, C is the capacitance change per cycle, which is proportional to the number of transistors whose inputs change, V is voltage, and F is the processor frequency (cycles per second). Therefore, an increase in frequency increments the amount of energy used in the processor. This situation, which is related to the increasing of power consumption, led Intel—in May 2004—to the cancellation of the processors "Tejas" and "Jayhawk". This fact is usually cited as the end of frequency scaling that was the dominant paradigm of computer architecture [30].

3 What about artificial intelligence, parallelism and energetic efficiency?

In this section, we will try to highlight some of the 'fundamental' problems of artificial intelligence that cannot be solved with conventional computing techniques, or with the hardware that is used today.

3.1 Some drawbacks of artificial intelligence

Artificial intelligence is a multidisciplinary field that benefits from other areas, such as computer science, logic and philosophy. Main interest of AI is the creation and design of systems that are capable to reason for themselves using the paradigm of human intelligence. AI can be envisaged from two different perspectives, namely (a) conventional artificial intelligence, and (b) computational intelligence. Conventional artificial intelligence is also known as deductive AI and is based on the formal and statistical analysis of human behavior when trying to solve different problems. On the other hand, computational intelligence (also known as inductive AI) involves development or interactive learning (interactive parameter changes in connectionist systems) using empirical data. The main criticism about artificial intelligence programs relates to their ability to mimic human thinking. However, these criticisms ignore that humans have not the capacity to solve all kinds of problems. In this context, authors such as Gardner have proposed that there are multiple intelligences [16].

In any case, whatever the approach, the efficiency of an artificial intelligence program is also conditioned by the lack of massive computational capacity. Some examples: (a) in rule-based expert systems is, computationally speaking, a very expensive task the pattern matching of rules; (b) in neural systems, data mining, and machine learning, the efficiency of the artificial intelligence program is limited by the large amount of data to be handled. We are faced once again with the difficulty of transforming data into information and information into knowledge.

3.2 Is "Parallelism" a solution?

But... why not make use of the paradigm of parallel computing to solve these problems? Certainly the answer is not easy, but it could be related to some energetic considerations and also to quantum constraints, both derived from the famous Moore's Law: "The number of transistors per inch on integrated circuit will double every 18 months and this trend will continue for at least two decades". This prediction was formulated by Gordon Moore in 1965. Today, Moore's Law still applies. The problem will arise when the new technologies allow to manufacture chips of around 5 nm. We will briefly discuss what this means.

Modern microprocessors are working on 64-bit architectures integrating more than 700 million transistors and they can operate at frequencies above 3 GHz. For instance, the third-generation Intel Core (2012) evolved from 32 nm wide to 22 nm, thus allowing duplication of the number of transistors per surface unit. Moreover, a larger number of transistors means that a given computer system could be able to do more tasks simultaneously. But, following the arguments of Feynman there is an "essential limit" for this to be done [13].

3.3 The limits of the so-called "Tunnel Effect"

The fact is that increasingly smaller microchips are manufactured. And the smaller is the device, the faster the computing process is reached. However, we cannot infinitely diminish the size of the chips. There is a limit at which they stop working correctly. When it comes down to the nanometer size, electrons escape from the channels where they circulate through the so-called "tunnel effect", a typically quantum phenomenon [20]. Electrons are quantum particles and they have wave-like behavior, hence, there is a possibility that a part of such electrons can pass through the walls between which they are confined. Under these conditions the chip stops working properly. In this context the traditional digital computing should not be far from the limits, since we have already reached sizes of only a few tens of nanometers. But... where is the real limit?

The answer may have much to do with the world of the very small. We will discuss this point in some detail, for which we will have no choice but to recover some ideas and concepts of physics, and the first concept to go over has to do with the size of the atom. In this respect, the various existing methods to estimate the size of the atomic radius give values between 0.5 and 5 Å. The size of current microprocessors is of the order of nanometers (nm), while the size of typical atoms is of the order of angstroms (Å). But 10 Å = 1 nm. We only have to go one order of magnitude further, prior to designing our computers considering the restrictions imposed by quantum mechanics!

3.4 Energetic restrictions of the information processing

In 1960 Rolf Landauer [19] began to wonder if the physical laws imposed some limitations on the computational process. Specifically he faced the problem of the origin of the heat dissipated by the computers and wondered if this fact was inherent to the laws of physics or it was due to lack of efficiency of the technology available. The subject looks really interesting if we remember that one of the problems of current high-speed computers is the removal of the heat produced during operation. The question is as follows: is there any relationship between energy, information and computing? The answer to this question seems trivial for those who nowadays work with computers (all of us). Energy in the form of heat, noise, etc., is omnipresent when we perform something with a computer (if you do not believe it, try to create something as simple as a sum using a computer without batteries). Indeed, for a computer system to be operational it is required the flow of electrons through cables and devices, thus generating frictional heat that has to be dissipated. To prevent this heat, we use a fan that needs energy to function and this also generates noise (another form of energy). Besides, the computer itself needs to be fed with an external power. There is energy everywhere. But this is not what concerns us here. And we said we had try to talk about the computer itself. To do this we will approach energy issues of computing from the point of view of the information we manipulate [3,4,14].

Let us consider a system which is constituted by two tanks connected together via a tube which can be opened or closed with a valve, in which there are two different and easily distinguishable particles: particles (x) and particles (o). Let us imagine such a system in two different situations. In 'situation 1' all particles (o) are on the left, all particles (x) are to the right, and the valve is closed. In 'situation 2' the valve is open and both types of particles (o) and (x) are homogeneously distributed in the two reservoirs. Let us now consider our system from a macroscopic perspective: if the number of both particles is large... how can we evaluate the amount of energy, which is related to the concept of 'information', in both situations of our system?

We will try to answer this question through statistical mechanics [14]. In statistical mechanics the concept of entropy is defined as follows: "The entropy is a measure of disorder in a system, equal to the Boltzmann constant times the natural logarithm of the number of microscopic states corresponding to the thermodynamic state of the system". According to the above definition:

$$S = k \ln(N)$$

In this equation S is the entropy, k is the Boltzmann constant, and N is the number of microscopic states corresponding to the thermodynamic state of the system. We recall now that the "amount of information" as defined by Shannon is:

$$\Psi = k \ln(N)$$

Perhaps k and N are not the same thing in both expressions, but the equations are identical... there must be some link between entropy and the amount of information of a system. Let us think a little bit: if entropy is a measure of disorder in a system, it is clear that situation 2 is much messier than situation 1. In other words, there are much more possible microscopic states in situation 2 than in situation 1. Thus entropy of the situation 1 is much smaller than that of situation 2. If $N(1)$ is the number of microstates of the situation 1, and $N(2)$ is the number of microstates of the situation 2, then:

$$S(1) = k \ln[N(1)] \quad \text{and} \quad S(2) = k \ln[N(2)]$$

But in physics it does not make much sense to speak about absolute entropies. The physical meaning relates to increments of entropy. Thus:

$$\begin{aligned}\Delta S &= S(2) - S(1) = k \ln[N(2)] - k \ln[N(1)] \\ &= k \ln[N(2)/N(1)]\end{aligned}$$

In this formulation the underlying assumption is that we start from situation 1, then we open the valve, and we let the system evolve. In this way the particles are disordered, and the entropy of the system increases, so that ΔS is positive. But now, what happened to our information? When we have our system in situation 1 we know many things... we know that all particles (o) are on the left side and all the particles (x) are on the right side. Now we open the valve and we let the system to evolve. Early in the process -that is, just after opening the spigot- we know that 'almost' all particles (o) are on the left side, and 'almost' all particles (x) are on the right side. We are losing information.

When the system reaches equilibrium (which roughly means that, macroscopically, the system no longer evolves) we have lost all the information. In fact, if the system is in situation 1, and someone want to take out a particle (x), he or she will go directly to the reservoir on the right and will take any of the particles. If someone asks for the same with the system in situation 2, he or she needs go to any of the deposits to take a particle, then it will be necessary to verify that the particle is a (x) particle and, otherwise, it will be necessary to return the particle to the system and continue testing until finally, by chance, a particle (x) appears.

To say that it is chance that guides the success of our mission with the system in situation 2 is the same as saying that in situation 2 our information is zero. Therefore, if $\Psi(1)$ is the information associated to situation 1, and if $\Psi(2)$ is the information associated to situation 2, and since we have already said that we can assume that our information in situation 2 is zero, then:

$$\Delta\Psi = \Psi(2) - \Psi(1) = -\Psi\,(1)$$

Now let us recap: starting from situation 1 and letting the system evolve spontaneously to reach the equilibrium state given by situation 2, an entropy increase occurs, but a loss of information also occurs. According to our previous formulation, we could say that the loss of information involved when we let the system spontaneously change is 'equivalent' to an increase of entropy of the system. However, we will be more cautious and will say that:

$$[\Delta\Psi = -\Psi(1)]\,\alpha\ \Delta\,S$$

We interpret this expression as follows: the loss of information that occurs when we let the system spontaneously evolve is related to the increase in entropy of the system. We can find many examples of what we have just established in everyday life. We will suggest to visualize what we have discussed trying to answer the same question in two different scenarios:

- Question: Where can I find my blue socks?
- Scenario 1: We are looking for them at our mother's home.
- Scenario 2: We are looking for them in our student's apartment.

So far we have tried to know something about the energy of computing, but the truth is that we have not fully achieved our goal. At best we have come to establish a link between the entropy of a system and its information. Anyway we are not far: there is a thermodynamic function, free energy (F), which relates the internal energy of a system (U) with its entropy (S). To illustrate how this works we will analyze an interesting problem formulated by Feynman [14]. The problem is: how much energy is needed to perform a 'basic' computation? We return to a previous question. What we are really trying to find out is what is the minimum energy required for an elementary computation.

Let us consider the structure of a typical message written on an endless belt that transports boxes with bits $|0\rangle$ and with bits $|1\rangle$, according to the arbitrary criteria that a bit $|0\rangle$ is a particle on the right of the box, and a bit $|1\rangle$ is a particle on the left of the box. Let us now 'reboot the system' by putting all the bits in the state $|0\rangle$.

...	$\vert 1\rangle$	$\vert 0\rangle$	$\vert 0\rangle$	$\vert 1\rangle$	$\vert 1\rangle$	$\vert 0\rangle$	...
...	x	x	x	x	x	x	...

It is apparent that the 'amount of information' in a message is proportional to the free energy required to restart the entire tape to zero. The first thing to understand is that reset to zero is equivalent to compressing each cell of the tape to ensure that its constituent particle is at position $|0\rangle$ -that is to say, to the right of the box according to the established criteria-. In this case we can assume that we drive a piston from left to right so that when we travel just half way (and therefore have reduced the volume of each box in half) we have all the bits on the right side. Thus, if the initial volume of each box is V_1 and V_2 is the volume of the box after restarting, it follows that $V_2 = (1/2)V_1$. In thermodynamics, state changes are studied from two fundamental concepts: the free energy F and the entropy S. With the restrictions of our problem, the expression that relates both magnitudes is:

$$\partial F = -T\,\partial\,S$$

In the above equation, T is the temperature of the system, which we will keep constant by a suitable refrigeration system (recall that we only aim to find out how much it is the 'fundamental energy' of an elementary computation). Now, if N is the number of bits of the entire message, the kinetic theory predicts that:

$$-\partial F = W = NkT\,\ln(V_1/V_2)$$

where k is the Boltzmann constant. Obviously ∂F is the energy that we have to use in order to restart the message. Therefore, if we consider one single basic operation on a single bit, it follows that:

$$\Delta F = kT\,\ln(2)$$

and consequently:

$$\Delta S = -k\,\ln(2)$$

However, we are forgetting something very important: we spend free energy only if we do not know where the particle is. This is because the above-mentioned is the only circumstance in which the phase space is divided by two. Conversely, if we know in advance the position of the particle we do not expend energy to restart. The conclusion is obvious: the information contained in a message is in the unknown bits, and the elemental energy of a basic computation is defined only in the case of 'unknown bits'. Does it mean that we can perform computations without energy? Obviously not, but we will better understand the relevance of the energetic approach when we will face the quantum operators and the reversibility of the quantum logical gates.

4 Introducing quantum theory

In general, we can say that computing is creating sets of symbols (results) from certain sets of initial symbols (or data). If we interpret the symbols as physical objects, computing corresponds to the evolution of the states of the systems. Therefore, this development is an example of computation. If this evolution follows the laws of quantum mechanics then we have quantum computing [5,6,9,10,22]. Since quantum computations are based on the quantum properties of the so-called "quantum bits" (or simply *qubits*), and the manipulation of qubits must be in accordance with the laws and restrictions of quantum mechanics, we have to understand the basic principles and the language of quantum mechanics.

4.1 Basic principles of the quantum approach

Quantum computing is a type of computer technique that uses quantum mechanics tools for its development. For this reason, and although it is not the purpose of this paper to explore in depth the intricacies of quantum mechanics, it will be however necessary to understand some of its most basic and elementary principles. In this regard, we will introduce the axiomatic formulation of Heisenberg and Dirac which is based on the following postulates [20]:

- Postulate I. The state of a physical system is described by a function $\psi(q, t)$ of the coordinates (q) and time (t).
- Postulate II. The temporal evolution of the state of a system is given by the Schrödinger equation.
- Postulate III. Each physical observable in quantum mechanics corresponds to a linear and Hermitian operator **A**.
- Postulate IV. Whatever the state function of a system, the only values that can result from a measure of the physical observable A are the eigenvalues a of the equation: $\boldsymbol{A}\psi_i = a_i\psi_i$
- Postulate V. If **A** is a linear Hermitian operator which represents a physical observable, then the ψ_i eigenfunctions of the equation $\boldsymbol{A}\psi_i = a_i\psi_i$ form a complete set.
- Postulate VI. *If* $\psi_i(q, t)$ is the normalized function of a system state at time t, then the mean value of a physical observable A at time t is:

$$\langle A\rangle = \int \psi * \mathbf{A}\,\psi\,\mathrm{d}q$$

- Postulate VII. If **A** is a linear and Hermitian operator that stands for a physical observable, the eigenfunctions, a_i, of operator $\boldsymbol{A}$ form a complete set.

4.2 Quantum operators

Once defined the principles of quantum mechanics, we will have to know what to do with them. We need a set of operators. An operator (denoted by BOLD capital letters) is a rule or procedure that, given a function, calculates another corresponding function. In the quantum approach all operators are linear. What follows is a list of things, operations, and properties, featuring our friends the quantum operators. Thus, we can define the following basic operations of quantum operators:

- $(\mathbf{A} + \mathbf{E}) f(x) = \mathbf{A} f(x) + \mathbf{E} f(x)$
- $(\mathbf{A} \times \mathbf{E}) f(x) = \mathbf{A}\{\mathbf{E} f(x)\}$
- $(\mathbf{A} \times \mathbf{E}) f(x) \neq (\mathbf{E} \times \mathbf{A}) f(x)$

We can also define an algebra of quantum operators with the following elements:

- Given two operators **A** and **E**: $\mathbf{A} = \mathbf{E} \leftrightarrow \mathbf{A} f = \mathbf{E} f,\ \forall f$
- The unit operator is defined
- The null operator is defined
- The associative property is true: **A** (**E I**) = (**A E**) **I**
- Not always the commutative property holds: **AE** $\neq$ **EA**
- The commutator of two operators is defined as: [**A**, **E**] = **AE** − **EA**
- The square of an operator is defined as: $\mathbf{A}^2$ = **A A**.

4.3 Dirac notation

A drawback of all this mess is that lots of integral equations can appear. The solution comes from the hand of Paul Dirac, who proposed an alternative notation which we briefly outline. If a given state can be described as a column vector we use the notation 'ket', and if it can be described as a row vector we use the notation 'bra' as illustrated below:

$$\text{ket}\ (\psi) = |\psi\rangle = \begin{pmatrix} a \\ b \end{pmatrix} :\ \text{bra}\ (\psi) = \langle\psi| = (a \quad b)$$

On the other hand, it holds that:

$$\langle m|n\rangle^* = \langle n|m\rangle\ :\ \langle m|m\rangle^* = \langle m|m\rangle$$

We believe that the concepts presented here should be enough to understand what we intend to explain in the lines that follow. It is, however, an unfinished business because in quantum mechanics all the operations are reversible. As we have to operate under the constraints imposed by quantum mechanics, this circumstance leads us to formally address the problem of reversibility. It will be immediately.

4.4 Reversibility and computing

A logic gate whose output data is less than the input is irreversible because it has to discard information, which ultimately translates into a loss of energy of some kind. Conversely, a logic gate whose output information is the same as the input information will be reversible or not dissipative, and the information remains 'constant', which also carries a constant energy [15,19,29]. The first reversible logic operation that we encounter is the binary negation, which is implemented by the NOT logical operator, that will be denoted from now with either the letter N or the symbol (¬). The binary negation is clearly reversible. We only have to remember that:

$$|1\rangle = \neg|0\rangle \quad \text{and} \quad |0\rangle = \neg|1\rangle$$

Similar to the N gate we will now work on the CN gate or controlled-NOT operator. This new gate CN is a device with two inputs and two outputs where the top line (A) is the line of control. The bottom line (B) is a NOT but it is controlled by the top line. This gate can be interpreted as a binary disjunction of two inputs and two outputs. The operation of CN must respect the following restrictions:

1. $|A_{\text{out}}\rangle = |A_{\text{in}}\rangle$
2. $|B_{\text{out}}\rangle = |B_{\text{in}}\rangle \leftrightarrow |A\rangle = 0$
3. $|B_{\text{out}}\rangle = \neg|B_{\text{in}}\rangle \leftrightarrow |A\rangle = 1$

If the status of the line (A) is $|1\rangle$ then the value of the input line (B) is reversed, but if the input to the line (A) is $|0\rangle$ then the line (B) passes unchanged. The entry on line (A) activates the N operation on the bottom line (B), and the output of (A) is always the same as the input of (A). Clearly it can be interpreted $|B_{\text{out}}\rangle$ as the output of an XOR gate with inputs $|A_{\text{in}}\rangle$ and $|B_{\text{in}}\rangle$:

$$|B_{\text{out}}\rangle = \text{ XOR } (A_{\text{in}}, B_{\text{in}})$$

However, the device is not the same, as CN gate produces two outputs rather than one. This gate is reversible as well: once known the output, we can always play back the input. In any case, N and CN are not enough to 'compute everything'. We need a reversible gate that constitutes, by itself, a complete set of operators. We introduce for that the CCN operator (controlled-controlled-NOT), also called 'Toffoli Gate' [29]. The operation of the CCN operator is as follows:

1. There are two control lines, A and B such that: $|A_{\text{out}}\rangle = |A_{\text{in}}\rangle$, $|B_{\text{out}}\rangle = |B_{\text{in}}\rangle$
2. Line C is only activated when: $(|A\rangle = 1) \wedge (|B\rangle = 1)$
3. In this case $|C_{\text{out}}\rangle = \neg|C_{\text{in}}\rangle$
4. If we keep $|A\rangle = |B\rangle = 1$ then CCN behaves as N in the line C
5. If we only maintain $|A\rangle$ (or $|B\rangle$) = 1 then CCN behaves as CN.

We have already said that the CCN gate itself is a complete set of operators. For example, the logical reversible XOR operator can be constructed from CCN setting $|A\rangle = 1$ or $|B\rangle = 1$. Also, the logical operator AND can be constructed from CCN setting $|C_{\text{in}}\rangle = 0$ and playing with the states of lines A and B. Moreover, the NAND logic operator can be constructed from the CCN gate setting $|C_{\text{in}}\rangle = 1$ and playing with the states of lines A and B. It is interesting to remember that the classical NAND is, by itself, an entire set of universal gates. With CCN we have the same but in a reversible manner. A complete description of the above-mentioned reversible logical gates can be found in [14].

Now we are ready to construct a reversible computer. We will define the reversible computer as the one that outputs the actual result of a computation together with the original entry. Also, one can show that, theoretically, reversible computing can be performed with a null energy cost [14]. The only energy cost incurred appears when rebooting the machine to restart another computation. Furthermore, the energy involved is not dependent on the complexity of the calculation. It just depends on the number of bits of the response. We can have N components running on a machine, but if the answer we get is just a bit, the energy needed to make everything work is just $k \times T \times ln\, 2$. Therefore, speaking in terms of 'basic energy' we can say that reversible computing does not require the establishment of a minimum of energy.

4.5 Quantum information

We will continue our discussion by introducing the basic unit of information used in the quantum approach [6,9,22]. As we previously noted, in quantum computing the basic unit of information is the quantum bit or qubit. A qubit can be in any linear combination of the two different basic states that are denoted $|0\rangle$ and $|1\rangle$, respectively. Physically the situation is represented by a two-state quantum system. The best known quantum system of two states is the spin of an electron. In a system of this type we can represent the spin $-(1/2)$ as the state $|0\rangle$ and spin $+(1/2)$ as the state $|1\rangle$. The qubit is an element of a Hilbert space, generated by the kets:

$$\{|0\rangle, |1\rangle\}$$

These states can conventionally be represented as follows:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

The two vectors are orthonormal, which means that under the inner product $\langle x|y\rangle$ we find that:

$$\langle 0|0\rangle = \langle 1|1\rangle = 1 : \langle 0|1\rangle = \langle 1|0\rangle = 0$$

A given qubit, in general, is presented as a linear combination of the basic states $|0\rangle$ and $|1\rangle$ such that:

$$|\psi\rangle = \alpha|\ 0\rangle + \beta|\ 1\rangle$$

where probability amplitudes α and β and are generally complex numbers, that is, they contain phase information. As in any measurement in quantum mechanics, the squares of these coefficients, respectively, determine the probability of obtaining outcomes $|0\rangle$ and $|1\rangle$. Since the total probability must be 1, α and β must be related by the equation:

$$|\alpha|^2 + |\beta|^2 = 1$$

This equation simply says that, when measuring qubits, the system collapses, and one basic state or the other is obtained. Because of its quantum nature, any measurement of qubits inevitably alters our state, and $\{\alpha, \beta\}$ is transformed irreversibly in $\{0, 1\}$. In other words, we compute with qubits but the results of the computation are classical bits [18,27].

For practical purposes the information contained in a qubit is very small. To represent larger amounts of information, we have to make the tensor product of the n individual qubits. As with single qubits, a given n-qubit can be in an intermediate state.

4.6 The mystery of "Entanglement" and "Quantum Parallelism"

A given n-qubit is in an entangled state if we cannot describe it in terms of states of single qubits. For example, the 2-qubit described below:

$$\psi = (1/4)\ |00\rangle\ + (\sqrt{3}/4)\ |01\rangle\ + (\sqrt{3}/4)\ |10\rangle + (3/4)\ |11\rangle$$

is not in an entangled state because it can be written as a tensor product. In fact:

$$\psi = \left(\frac{1}{2}\,|0\rangle + \frac{\sqrt{3}}{2}\,|1\rangle\right) \otimes \left(\frac{1}{2}\,|0\rangle + \frac{\sqrt{3}}{2}\,|1\rangle\right)$$

However, there are sets of n-qubit states that cannot be described as a product of individual states of the n qubits. For example:

$$[1/\sqrt{2}]\ \{|00\rangle + |11\rangle\}$$

These are known as entangled states because the states of the two qubits are not independent. Also, we can say that:

$$|x_1 x_2 \ldots x_m\rangle \equiv |x\rangle$$

where:

$$x = x_1 2^{m-1} + x_2 2^{m-2} + \ldots + x_{m-1} 2^1 + x_m 2^0$$

In this way, the basis of a space formed by n qubits whose dimension is 2^n is formed by:

$$\{|0\rangle, |1\rangle, |2\rangle, \ldots, |2^n - 1\rangle\}$$

The string "$x_1 x_2 \ldots x_n$" can be interpreted as the natural number "x" represented in the binary numeral system. Thus the vectors of the basis B^n are identified with the natural numbers x that satisfy $0 \le x < 2^n$ (numbers with n bits). Having identified the bit string "$x_1 x_2 \ldots x_n$" with the natural number "x" we can write "x" in the decimal system. Thus, we can write:

$$B^n = [|\ 0\rangle\,, |\ 1\rangle\,, |\ 2\rangle\,, \ldots, |2^n - 1\rangle]$$

With this notation a given n-qubit can be written as follows:

$$\psi = \sum_{x=0}^{2^n-1} a_x\,|x\rangle \text{ with } \sum_{x=0}^{2^n-1} |a_x|^2 = 1$$

It should be emphasized that the size of space is exponential, namely 2^n. This is the key property called quantum parallelism which is the responsible of the enormous capacity of an n-qubit to store information.

4.7 The collapse of quantum information

A big problem that arises with qubits is that, generally speaking, is not possible to measure the state of a qubit in a deterministic manner. The postulates of quantum mechanics state that the probability P_0 or P_1 of the final state of the qubit to be $|0\rangle$ or to be $|1\rangle$ is equal to the square modulus of the amplitude of $|0\rangle$ or $|1\rangle$ in the linear combination.

In a 2-qubit we measure the first qubit and the second qubit. The process is similar in both cases. Suppose then that we measure the first qubit. Take for example the state:

$$\psi = (1/\sqrt{3})|\ 00\rangle\ + (1/\sqrt{3})|\ 01\rangle\ + (1/\sqrt{3})|\ 10\rangle$$

After measurement, the first qubit must be $|0\rangle$ or $|1\rangle$. Therefore the 2-qubit must be, after measurement, in any one of the following states:

$$|0\rangle \otimes [a|0\rangle + b|1\rangle] = a|00\rangle + b|01\rangle \text{ with } a, b \in C \text{ such that } |a|^2 + |b|^2 = 1$$

$$|1\rangle \otimes [a|0\rangle + b|1\rangle] = a|10\rangle + b|11\rangle \text{ with } a, b \in C \text{ such that } |a|^2 + |b|^2 = 1$$

Thus, the behavior of a quantum system is different whether the system is or is not in an entangled state. A system is in an entangled state if the measure of one component affects the measure of the other, and the system is in a non-entangled state if this does not happen.

4.8 Building quantum algorithms

In quantum computing an algorithm is a mechanism for manipulating n-qubits. One of the two possible mechanisms to do this is to measure qubits. The other is to transform an initial state ψ_1 in the corresponding final state ψ_2. The evolution and dynamics of a given n-qubit is determined by 'unitary operator' U on the Hilbert space, these operators are called 'evolution operators'. If we define the function $U : V^n \to V^n$ such that:

$$U\psi_1 = \psi_2$$

then the application of U transforms states into states preserving the norm and, according to the postulates of quantum mechanics, this is a linear process. Thus, U can only be a 'unitary transformation'. A given operator is 'unitary' if:

$$U^{\dagger}U = I$$

But the application of a generic unitary transformation is not possible in quantum computing [18]. Therefore, we need a sequence of elementary unitary transformations performed through quantum-reversible gates. Ultimately a quantum algorithm is a finite sequence of quantum gates and measurements.

Also, in the definition of quantum algorithm we have to include two restrictions. The first restriction affects the initial state, which is always the same: $\Psi_1 = |0\rangle$. The second restriction is that the algorithmic procedure has to be perfectly ordered. First a sequence of quantum gates operates, and then a sequence of quantum measurement applies. But ... how does it work? Let $|\ \psi\ (\mathrm{t})\ \rangle = |x_1, \ldots, \mathrm{x}_n\rangle$ be a n-qubit. We can establish that the evolution of the quantum system after applying the operator U on a single computation step is given by:

$$U|\ \psi\ (0)\rangle \to |\ \psi\ (1)\rangle$$

And, in general, the development of m computing steps is given by:

$$U_m|\Psi\ (0)\rangle \to |\ \psi\ (m)\rangle$$

In the context of quantum computing, operators that perform on n-qubits correspond to unitary matrices of dimension 2^n. On the other hand, as we have already said, computing is creating sets of symbols (results) from certain sets of initial symbols (or data). If we interpret the symbols as physical objects, computing corresponds to the evolution of the states of the systems. We have already seen that, in quantum computing, these developments materialize from the unitary operators. But it is very important to keep in mind that such unitary operators are merely matrix representations of reversible logic gates, or quantum gates, with which we can build quantum circuits [1,2]. Unlike conventional logic gates, which can operate in $n \times m$ bits, the quantum gates must operate in $n \times n$ qubits. This is a necessary, but not sufficient, condition to satisfy the reversibility property of quantum gates. The reversibility of quantum gates is a consequence of the unitary nature of the operators. If U_f is the unitary matrix associated with a gate f then for any states $|x\rangle$ and $|y\rangle$ it follows that:

$$U_f\,|x\rangle = |y\rangle \Rightarrow U_f^{\dagger}U_f\,|x\rangle = U_f^{\dagger}\,|y\rangle \Rightarrow |x\rangle = U_f^{\dagger}\,|y\rangle$$

This means that from the output information it is possible to obtain the input information. Furthermore, from a function f of n bits in m bits we can build a reversible, $f_{\text{reversible}}$ function of "$m + n$" bits in "$m + n$" bits according to the following procedure:

$$f : x \to f(x) \Rightarrow f_{\text{reversible}} : (x, y) \to (x, y \oplus f(x))$$

Thus, a function f can be implemented by a quantum circuit U_f, fulfilling the conditions of reversibility required to it, if U_f performs the transformation:

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

Each quantum gate of n-qubits can be represented by a unitary matrix of size 2^n, where the transformation performed by the quantum gate is performed by the corresponding matrix operator. Given the description of the transformation that takes a quantum gate on the elements of the base space, the unitary matrix should be obtained from the following procedure:

- The rows of the matrix correspond to the input.
- The columns of the matrix correspond to the output.
- The (j, i) position of the matrix corresponds, when the i_{th} vector is the input to the gate, to the coefficient of the j_{th} vector in the output of the gate.

We shall see now how we can operate with some of the quantum gates. We begin with the "Identity" (I) that will serve to illustrate the process of construction of the unitary matrix associated to its operation.

$$U_{\text{identity}}\quad |0\rangle \to |0\rangle : U_{\text{identity}}\quad |1\rangle \to |1\rangle$$

$$\begin{array}{c|cc} & |0\rangle & |1\rangle \\ \hline |0\rangle & 1 & 0 \\ |1\rangle & 0 & 1 \end{array} \to U_{\text{identity}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Of course, the behavior of the unit matrix of identity is as follows:

$$U_{\text{identity}}\,|0\rangle = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle$$

$$U_{\text{identity}}\,|1\rangle = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$$

Let us now to look at the functioning of the Hadamard gate, which transforms a qubit in a superposition of the elements of the basis $\{|0\rangle, |1\rangle\}$. This operation is very important and it is present in the classical quantum algorithms we will see later. The description and the behavior of the Hadamard gate are illustrated below.

$$U_{\text{hadamard}}\,|0\rangle \rightarrow \frac{1}{\sqrt{2}}\,(|0\rangle + |1\rangle)$$

$$U_{\text{hadamard}}\,|1\rangle \rightarrow \frac{1}{\sqrt{2}}\,(|0\rangle - |1\rangle)$$

$$\begin{array}{c|cc} & |0\rangle & |1\rangle \\ \hline |0\rangle & \frac{+1}{\sqrt{2}} & \frac{+1}{\sqrt{2}} \\ |1\rangle & \frac{+1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{array} \rightarrow U_{\text{hadamard}} = \frac{1}{\sqrt{2}}\begin{pmatrix} +1 & +1 \\ +1 & -1 \end{pmatrix}$$

$$\begin{aligned} U_{\text{hadamard}}\,|0\rangle &= \frac{1}{\sqrt{2}}\begin{pmatrix} +1 & +1 \\ +1 & -1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\,(|0\rangle + |1\rangle) \\ U_{\text{hadamard}}|1\rangle &= \frac{1}{\sqrt{2}}\begin{pmatrix} +1 & +1 \\ +1 & -1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= \frac{1}{\sqrt{2}}\begin{pmatrix} +1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}}\,(|0\rangle - |1\rangle) \end{aligned}$$

Now let us do a little work on systems of two qubits. First, we should remember that a 2-qubit $|x, y\rangle = |xy\rangle$ is constructed as $|x\rangle \otimes |y\rangle$. Therefore, considering the basic 1-qubit vectors $|0\rangle$ and $|1\rangle$, it follows that:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} : |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$|0\rangle \otimes |0\rangle = |00\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$|0\rangle \otimes |1\rangle = |01\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|1\rangle \otimes |0\rangle = |10\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$|1\rangle \otimes |1\rangle = |11\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

The corresponding unitary matrices applicable to a 2-qubit system should have dimension 4 × 4. A trivial operation that can be defined in a 2-qubit system is the exchange operation (EX), which is limited to exchange the state of two lines. The truth table associated with this operation is as follows:

EX	A	B	C	D
	0	0	0	0
	0	1	1	0
	1	0	0	1
	1	1	1	1

Wherein *A* and *B* represent the inputs while *C* and *D* are the outputs. With conventional bits, EX behavior is as follows:

$$\begin{aligned} &\text{EX}(0\ 0) \rightarrow (0\ 0) \\ &\text{EX}(0\ 1) \rightarrow (1\ 0) \\ &\text{EX}(1\ 0) \rightarrow (0\ 1) \\ &\text{EX}(1\ 1) \rightarrow (1\ 1) \end{aligned}$$

Let us now operate with qubits and analyze the transformation:

$$\begin{aligned} &U_{\text{EX}}\,|00\rangle \rightarrow |00\rangle \\ &U_{\text{EX}}\,|01\rangle \rightarrow |10\rangle \\ &U_{\text{EX}}\,|10\rangle \rightarrow |01\rangle \\ &U_{\text{EX}}\,|11\rangle \rightarrow |11\rangle \end{aligned}$$

In terms of unitary matrices the above expressions are nothing more than the following:

EX	$\|00\rangle$	$\|01\rangle$	$\|10\rangle$	$\|11\rangle$
$\|00\rangle$	1	0	0	0
$\|01\rangle$	0	0	1	0
$\|10\rangle$	0	1	0	0
$\|11\rangle$	0	0	0	1

$$U_{\text{EX}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{Obviously} : U_{\text{EX}} \times U_{\text{EX}} = U_{\text{I}}$$

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

$$\text{And} : U_{\text{EX}}^{\dagger} = U_{\text{EX}}$$

The last two conditions ensure the reversibility of the door and the consistency with the postulates of quantum mechanics. In this respect, it should be clear that:

$$U_{\text{EX}}\,|00\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = |00\rangle$$

$$U_{\text{EX}}\,|01\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |10\rangle$$

$$U_{\mathrm{EX}}\,|10\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = |01\rangle$$

$$U_{\mathrm{EX}}\,|11\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |11\rangle$$

Another gate that has also been discussed when talking about reversibility was CN. This gate can be interpreted as follows:

$$|\ B_{\mathrm{out}}\rangle = \mathrm{XOR}(A_{\mathrm{in}}, B_{\mathrm{in}})$$

However, this device is not the same as a classical XOR gate, since CN gate produces two outputs rather than one. Exploring this in quantum terms:

$$U_{\mathrm{XOR}}|x, y\rangle \rightarrow |x, x \oplus y\rangle$$

U_{XOR}	$\|00\rangle$	$\|01\rangle$	$\|10\rangle$	$\|11\rangle$
$\|00\rangle$	1	0	0	0
$\|01\rangle$	0	1	0	0
$\|10\rangle$	0	0	0	1
$\|11\rangle$	0	0	1	0

$$U_{\mathrm{XOR}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$U_{\mathrm{XOR}}\ |00\rangle \rightarrow |0, 0 \oplus 0\rangle = |00\rangle$$
$$U_{\mathrm{XOR}}\ |01\rangle \rightarrow |0, 0 \oplus 1\rangle = |01\rangle$$
$$U_{\mathrm{XOR}}\ |10\rangle \rightarrow |1, 1 \oplus 0\rangle = |11\rangle$$
$$U_{\mathrm{XOR}}\ |11\rangle \rightarrow |1, 1 \oplus 1\rangle = |10\rangle$$

$$U_{\mathrm{XOR}}\ |00\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = |00\rangle$$

$$U_{\mathrm{XOR}}\ |01\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = |01\rangle$$

$$U_{\mathrm{XOR}}\ |10\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |11\rangle$$

$$U_{\mathrm{XOR}}\ |11\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |10\rangle$$

What we have briefly discussed in this section give us enough background to understand the main characteristics of some of the most relevant quantum algorithms, which will be faced in the next section.

5 An overview of some relevant quantum algorithms

In the following paragraphs we are going to present some quantum algorithms that ‘do something in a quantum manner’. We are not going to discuss any of them in deep, but a quick overview will pave the way for further thinking about the possibility of using the quantum paradigm in the field of artificial intelligence. In the description of the algorithms we will use the matrix representation of serial and parallel operations, having in mind that a serial computation corresponds to a multiplication of matrices that represent the gates. On the other hand, parallel operations are represented through the corresponding tensor product of matrices. The ‘classical quantum algorithms’ we will introduce are: (a) the Deutsch–Jozsa algorithm, (b) the Simon’s algorithm, (c) the Quantum Fourier Transform, (d) the Shor’s algorithm, and (e) the Grover’s algorithm.

5.1 The Deutsch–Jozsa algorithm

The Deutsch–Jozsa algorithm is one of the first algorithms designed to run on a quantum computer and has the potential of being more efficient than classical algorithms. The idea is to exploit the inherent parallelism of quantum algorithms and the superposition of states [18,27]. In the Deutsch–Jozsa problem we are given a quantum function (which for us is a black box) that takes n input bits $x_1, x_2, ..., x_n$ and returns a binary value $f(x_1, x_2, ..., x_n)$. We know that the function is constant (0 for all inputs or 1 on all inputs) or balanced (returns 1 for half of the entries and 0 for the other half). The problem is then to determine how the function is (constant or balanced) using inputs to the black box and observing the outputs. A peculiarity of the Deutsch–Jozsa algorithm is that the sign of the corresponding amplitudes is determined by $(-1)^{f(x)}$. A simplified version of the Deutsch–Jozsa algorithm (which will be used to illustrate how this algorithm works) is the Deutsch algorithm, that refers to a single target input or, in other words, $n = 1$ and $f(x_1, x_2, ..., x_n) = f(x)$. In this case, we need a quantum gate of two inputs and two outputs that performs the following operation:

$$U_{\mathrm{DJ}}\ \ |x\rangle|y\rangle = U_{\mathrm{DJ}}|x, y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle = |x, y \oplus f(x)\rangle$$

In the matrix representation the entire algorithm for $n = 1$ is as follows:

$$|\psi_{\text{out}}\rangle = (U_{\text{H}} \otimes U_{\text{I}})U_{\text{DJ}}(U_{\text{H}} \otimes U_{\text{H}})|0\rangle|1\rangle$$

The previous algorithm must be read from right to left, U_{H} is the Hadamard gate, U_{DJ} is the Deutsch–Jozsa operation and U_I is the identity gate. After applying this algorithm we have to measure de first qubit of $|\psi_{\text{out}}\rangle$, which collapses to 0 or 1. It can be demonstrated that if we obtain 0 then f is constant. On the other hand, if we obtain 1 then f is balanced. Detailed and pedagogic descriptions of this algorithm can be found in [11,32].

The algorithm has almost no practical use, but it is one of the earliest examples of a quantum algorithm that has been shown to be exponentially faster than any possible deterministic classical algorithm.

Quantum computing allows to solve the problem since it is capable of simultaneously evaluating $f(0)$ and $f(1)$. This possibility stems from the so-called 'quantum parallelism'. The quantum parallelism allows to compute 2^n entries for a state consisting of n-qubits. That is: from a linear growth in the number of qubits, exponential growth in computing space is achieved.

5.2 Simon's algorithm

This algorithm is about finding the period of a Boolean vector function of the type:

$$f : \{0, 1\}^n \to \{0, 1\}^n$$

The period will be a Boolean vector $\boldsymbol{c}$ that meets with the following restriction:

$$f(\boldsymbol{x}) = f(\boldsymbol{x} \oplus \boldsymbol{c})/\boldsymbol{c} \in \{0, 1\}^n$$

This can be rewritten as follows:

$$\boldsymbol{c} = c_0c_1 \ldots c_{n-1} \to \forall \boldsymbol{x} \forall \boldsymbol{y} \in \{0, 1\}^n$$
$$f(\boldsymbol{x}) = f(\boldsymbol{y}) \leftrightarrow \boldsymbol{x} = \boldsymbol{y} \oplus \boldsymbol{c}$$

It is not difficult to see that conventional assessment of f involves (in the worst case) evaluation of more than half of the elements of the domain ($2^{n-1} + 1$). This means that the computational cost would be exponential. The quantum approach proposed by Simon to resolve the problem is as follows:

- Definition of the unitary transformation:
  $U_{\text{S}}|\,\mathbf{x}\rangle\,|\,\mathbf{y}\rangle = |\,\mathbf{x}\rangle\,|\mathbf{y} \oplus f(\mathbf{x})\rangle$
- Definition of the algorithm:
  $\langle\psi_{\text{out}}\rangle = (U_{\text{H}}^{\otimes\text{n}} \otimes U_{\text{I}})U_{\text{S}}(U_{\text{H}}^{\otimes n} \otimes U_{\text{I}})|\,\mathbf{0}\rangle\,|\,\mathbf{0}\rangle$

Inputs are now strings, U_{H} is again the Hadamard gate, $U_{\text{H}}^{\otimes n}$ is the tensor product $U_{\text{H}} \otimes \ldots U_{\text{H}}$ (n times). After applying the algorithm, measurement of the first string of $|\psi_{\text{out}}\rangle$ gives a system of linear equations from which we can evaluate $\boldsymbol{c}$. It can be demonstrated that using Simon's algorithm we only need to perform n evaluations of the elements of the domain instead of the $2^{n-1} + 1$ evaluations needed when using the conventional approach. A detailed and pedagogic description of the Simon's method can be found in [32].

5.3 Quantum Fourier transform (QFT)

From the previous examples we can easily infer that one of the most powerful tools to perform quantum computations is the Hadamard gate. Truly speaking the Hadamard gate can be considered as a particular case of the so-called quantum Fourier transform (QFT). On the other hand, QFT is the quantum version of the discrete Fourier transform (DFT). The complexity of DFT is $O(N^2)$. In any case, if $N = 2^n \to O(N^2) = O(2^{2n})$. The performance of DFT can be improved by using the fast Fourier transform (FFT), whose complexity is $O(N \log N) = O(n^{2n})$ but still remains exponential. It can be demonstrated that the exponential complexity of the classical approaches, DFT and FFT, is considerably improved in the quantum version, QFT. In fact, the complexity of the QFT is $O(n^2)$. Quantum Fourier transform can be defined as follows:

$$U_{\text{QFT}}|x\rangle = N^{-1/2}\sum\nolimits_y \text{e}^{2\,\pi\,ixy/N}|y\rangle$$

This important operation plays a fundamental role in one of the most successful algorithms of quantum computing, namely Shor's algorithm, which we will present immediately. A detailed and pedagogic description of the Quantum Fourier Transform can be found in [32].

5.4 Shor's algorithm

Perhaps the most important and well-known quantum algorithm is Shor's algorithm for factoring numbers. The problem to be resolved can be defined as follows: given a number $N \in Z$ the goal is to find another number $p \in Z$verifying $1 < p < N$ such that p divides N.

The procedure proposed by Shor to resolve the problem involves both classical and quantum approaches. The classic part of the algorithm can be described as follows:

1. Choose a pseudo-random number $a < N$
2. Use the "Euclidean Method" to calculate the greater common divisor gcd $(a,\ N)$
3. If $\gcd(a, N) \neq 1$ then a is a non-trivial factor of $N \to$ End
4. If $\gcd(a, N) = 1$ then find the period of the function: $f(x) = a^x \text{mod}\, N$. We will denote this period as r. The number $r \in Z$ is the smallest number such that:
   $f(x + r) = f(x)$

5. If r is an odd number then go back to step 1
6. If $a^{r/2} \equiv -1(\text{mod } N)$ then go back to step 1
7. The factors of N are gcd $(a^{r/2} \pm 1, N) \rightarrow$ End

The quantum part of the algorithm is the subroutine designed to find the period of the function, and can be outlined as follows:

1. Define two input/output quantum registers, each of them with $log_2 N$ qubits
2. Given $0 \leq x \leq N-1$ define the following initial state: $|\psi_0\rangle = N^{-1/2} \sum_x |x\rangle|0\rangle$
3. Define the quantum function $f(x)$ and apply it to $|\psi_0\rangle$ to obtain $N^{-1/2} \sum_x |x\rangle|f(x)\rangle$
4. Apply the Quantum Fourier Transform to the input register.
5. Perform a measurement. We will obtain a given result y in the input register and $f(x_0)$ in the output register.
6. Operate on y/N to obtain an irreducible fraction. The corresponding denominator r' could be the period r we are looking for.
7. If $f(x) = f(x+r')$ then r' is the period $r \rightarrow$ End of the subroutine.
8. If $f(x) \neq f(x+r')$ then try to obtain more candidates of r using either some other values close to y or some multiples of r'.
9. If any of the new candidates satisfies the requirements to be the period r then → End of the subroutine.
10. If none of the new candidates satisfies the requirements to be the period r then → Go back to step 2.

The main interest of Shor's algorithm is in cryptography. In fact RSA, for example, would become useless if Shor's algorithm is implemented in a practical quantum computer. An encrypted message can be decrypted factoring the public key N, which is the product of two primes. The known classical algorithms cannot do this in $O(N_k)$ for any k, so these algorithms become unpractical as N increases. Conversely, Shor's algorithm can break RSA in polynomial time. Shor's algorithm is probabilistic since it gives the correct answer with a probability, and the probability of error can be reduced by repeating the algorithm. A detailed description of this algorithm can be found in [25].

5.5 Grover's algorithm

One outstanding algorithm that could be useful for our purposes of putting together quantum computing and artificial intelligence is the so-called Grover's search algorithm. A search algorithm is a procedure that allows to find an element x_0 in a possible set of solutions given a proposition $f(x)$. Among such problems is, for example, searching a database. If we know nothing about the structure of the solution space we are facing an unstructured problem. It is not difficult to demonstrate that the best conventional randomized algorithm would lead to a cost of $O(N)$ for a database of size N. This result is improved using Grover's algorithm by a quadratic gain $O(\sqrt{N})$. It has to be taken into account that Grover's algorithm is also probabilistic, that is, it only gives the correct answer with a certain probability. However, the probability of an incorrect answer can be as small as desired given a sufficient number of iterations of the algorithm. In any case Grover's algorithm is not trivial and its description is far beyond the scope of this article. A detailed explanation of Grover's algorithm can be found in [17].

6 Quantum computing and artificial intelligence

Obviously quantum computing shows significant potential alternative solutions to problems currently 'intractable', however little progress has been made in this regard so far. Perhaps this is because not much effort has been spent, maybe because quantum computing technology is still in its infancy, and it could be too early to consider how quantum computing can be used in the field of artificial intelligence. In order to present the reader with some possible connections between quantum computing and artificial intelligence, what we will try to do in this section is: (1) to face a classical problem of artificial intelligence from the viewpoint of quantum computing; and (2) to explore some of the possible synergies between quantum computing and artificial intelligence.

6.1 The 'Search' problem

What follows is based on some ideas widely developed in the textbook "Principles of Quantum Artificial Intelligence", by Andreas Wichert [31]. According to Wichert, 'Search' is a classical problem of AI that can be treated from the viewpoint of quantum computing. Let us formulate the problem as follows:

Given a function $f(x)$ such that:

- $f_\varepsilon(x) = 1 \quad \text{if } x = \varepsilon$
- $f_\varepsilon(x) = 0 \quad \text{otherwise}$

what we try is to find x for which $x = \varepsilon$, that is to say: we are looking for $f(x) = 1$. This is nothing more than a decision problem with a binary answer and one instantiation x. In this context, the search for ε can be done by considering three restrictions of quantum computing in the circuit approach:

1. The function $f(x)$ will be represented by a quantum circuit U_f.
2. The properties of $f(x)$ are determined by the superposition principle and also by a given unitary transform.

3. The sign of the corresponding amplitudes is determined by $(-1)^{f(x)}$.

When we presented the reader with some of the 'classical quantum algorithms' (e.g., Deutsch–Jozsa algorithm) the function $f(x)$ was defined as a black box instead of being defined as a quantum circuit. If this is the case, we have to consider U_f as a 'quantum oracle' that performs an unitary operation of the type: $U_f | x\rangle\ | y\rangle = | x\rangle | f(x) \oplus y\rangle$. If the operator U_f is built from $O(m)$ gates then it is easy to see that we can get an increase in speed of $O(\sqrt{n})$ steps. In fact, Grover's algorithm implements exhaustive search in $O(\sqrt{n})$ steps in the corresponding n-dimensional Hilbert space. This fact can be used to improve the efficiency of, for example, the generate-and-test method. This could be done by means of the mapping of the so-called 'Generator' into a superposition of states and the so-called 'Tester' into an oracle implemented as a quantum circuit.

These ideas can be generalized taking into account that problem-solving in AI can be modeled by a production system that implements a search algorithm. The 'search' can then be represented as a tree. If we want to face the problem from the quantum computing perspective then we have to remind the following ways that allow us to speed up the computation:

- The quantum Fourier transform (QFT) can be used to determine the period of a periodic function exponentially faster than any known classical algorithm.
- The algorithm proposed by Grover can achieve an increase in speed of $O(\sqrt{n})$ steps.

On the other hand, in a tree, we search for a leaf ε. Let us use Grover's algorithm to illustrate the potential of quantum computing in this classical problem of AI. First, we need to represent the search tree by means of a quantum circuit U_{st}. Then we will see how this works in two types of 'uninformed tree search': (a) breadth-first search (BFS), and (b) depth-first search (DFS).

In a classical BFS the root node, which defines the level 0, is expanded first generating level 1. Then all nodes in level 1 are expanded generating level 2, and so on the process continues until the goal is reached or until we cannot generate a new deeper level. Main restriction of BFS is that all nodes at level N have to be reached before generation of a new node at level $N + 1$. Let us now consider a constant branching factor, b. In this case b^m nodes are expanded at level m, with $k = 0$ being the root. It is no difficult to see that with classical BFS the computing costs and the memory requirements are in the worst case $O(b^m)$.

Different from classical BFS, classical DFS always expands the deepest node in the search tree until the goal is reached or until no more valid transitions are possible in the branch under consideration. In the last case backtracking to the previous state is done and the new branch is again explored in a DFS manner. Again, it is easy to verify that if m is the maximum depth of the search tree, and if b is the constant branching factor, then the computational costs of DFS are in the worst case $O(b^m)$, and the memory requirements are $O(m)$.

Let us now apply the quantum computing approach to the tree search. Suppose a constant branching factor b for a tree whose depth is m. In this case we have $b^m = n$ leaves. Our goal is to explore all the leaves. According to Wichert [31] we will use the concept of 'ideal entropy' to evaluate the minimum number of optimal questions that describe the results of a given 'experiment'. This experiment represents n leaves of a search tree with equal probabilities: $p = (1/n, 1/n, \ldots, 1/n)$. Thus, the maximal ideal entropy that corresponds to the depth of the search tree is:

$$m = -\sum_i p_i \log_b p_i = \log_b n$$

Assume now that $b = 2$, then each of the m questions has the answer "true" or "false" and may be represented by a conventional bit. The m possible answers are represented by a binary register of length m, and there are n different binary registers representing all possible binary numbers of length m. On the other hand, each binary number represents a path from the root to a leaf. For each possible goal, a given binary number indicates the solution. In this context, using Grover's algorithm we can search through all possible paths and verify, for each path, whether or not it leads to a goal state. For this purposes, a quantum circuit U_{p} with a polynomial number of quantum gates can verify whether each path corresponds to a sequence of productions that lead from the initial state to the goal state. In this case it can be demonstrated that the computational costs are: $O(\sqrt{b^m})$ which is much less than the cost of uninformed tree search algorithms that, as we previously mentioned, is $O(b^m)$.

6.2 Exploring the potential synergies between quantum computing and artificial intelligence

As far as artificial intelligence is concerned much of the early research is related to artificial intelligence searching techniques. In fact, Grover's algorithm shows that quantum computing can behave faster than the best classical approaches. As a consequence of the latter, some artificial intelligence researchers believe that quantum search is one of the first techniques of quantum computing which play an important role in AI, but until now few successful applications of quantum search in AI have been reported. Let us think a little bit about what has been said in the previous paragraph.

Most of the so-called ‘decision problems’ can be formulated in terms of decision trees. Quantum algorithms can resolve decision problems represented by a class of decision trees exponentially faster than classical random approaches. But this does not necessarily mean any fundamental advantage of quantum computing over the ‘classical approach’, and this is because the decision problems can also be resolved very fast by means of the classical algorithms. In this context, perhaps the most important area in which quantum computing and artificial intelligence already converge is ‘machine learning’. Here, the goal is to find quantum procedures that perform better than existing classical learning algorithms, which have demonstrated to be highly inefficient.

The use of some ideas of quantum theory to face certain problems of artificial intelligence should be carefully analyzed. In part because there is a growing interest in the artificial intelligence community to develop computational models devoted to address the problems in the classical world for which classical approaches are not sufficient. New techniques are required, and quantum computing could be an interesting issue to be considered.

We will now change the perspective. There are several problems concerning quantum systems for which artificial intelligence approaches could be potentially used. We are now trying to establish a synergy between these two paradigms, quantum computing and artificial intelligence. For example, ‘Statistical Inference’ is related to the intrinsic probabilistic nature of quantum systems. On the other hand, ‘Bayesian networks’ are graphical models of probabilistic information that are widely used in the artificial intelligence arena. It is not difficult to see that this could be a matter that should be explored.

Another example concerns ‘Pattern Recognition’, which is an important area of artificial intelligence. In this context, discrimination of objects can be seen as a special case of pattern recognition. However, only recognition and discrimination of classical objects have been considered by artificial intelligence researchers. On the other hand, a lot of work on discrimination and recognition of quantum states and quantum operations has being carried out by physicists without knowing much about the existing artificial intelligence techniques. Again, it is not difficult to see that this could be a matter that should be explored.

These two last examples show that there is a huge interdisciplinary field to be investigated, and also that lots of efforts still need to be done to put quantum computing and artificial intelligence working together.

7 Discussion

Undoubtedly quantum computing could be a field of great interest and future in all fields of computer science in general and, specifically in artificial intelligence. In this respect it has been suggested the use of quantum computing as an efficient alternative to classical computing to resolve multiple problems, including integer factorization, discrete logarithm, or the simulation of processes. On the other hand, the always great Feynman conjectured in 1982 that quantum computers would be effective as universal quantum systems simulators, and in 1996 it was shown that the conjecture was correct. However, as in any emerging discipline, there are still many unresolved issues.

Quantum computing is based on the use of qubits instead of bits, and gives rise to new logic gates that enable construction of new algorithms which are based on totally different principles. The same task may have different complexity in classical computing and in quantum computing. This possibility has led to great excitement, as (at least theoretically) some intractable problems could become tractable. While a classical computer is equivalent to a Turing machine, a quantum computer is equivalent to a quantum Turing machine. Thus, in classical digital computing one bit can take only two values: 0 or 1. However, in quantum computing, involving the laws of quantum mechanics, the qubit can be in coherent superposition and it can be either 0 or 1 or 0 and 1 at the same time. This allows that several operations can be performed simultaneously. The number of qubits indicates the number of bits that can be in superposition. With conventional bits, if we have a record of three bits we have eight possible values and the record can only take one of these eight possible values. However, if we have a vector of three qubits, the particle can take eight different values simultaneously through quantum superposition. Thus, a vector of three qubits allows a total of eight parallel operations. As expected, the number of operations is exponential with respect to the number of qubits.

From a practical perspective, we are interested in some issues of quantum computing, still unresolved, that can be summarized in the following points: building strategies, data transmission, quantum algorithms and architectures and models. Regarding the first point, it has not yet completely solved the problem of which the ideal strategy for quantum computing is, however, a quantum system must satisfy the following hardware requirements:

- The quantum system has to be initialized, that is, brought into a known and controlled state of departure.
- It must be possible to tamper the qubits in a controlled manner with a set of operations performed through a universal set of logic gates.
- The system must maintain quantum coherence during the computations.
- It must be possible to read the final state of the system after the calculation.

- The system must be scalable and it must be defined the way of increasing the number of qubits in order to deal with problems of greater computational cost.

But one of the main obstacles of quantum computing is the problem of decoherence, which causes loss of the unitary character (and, more specifically, 'reversibility') of the steps of the quantum algorithm. Error rates are typically proportional to the ratio of operating time versus decoherence time, so that any operation should be completed in a much shorter time than the decoherence time.

Another major problem is scalability, a problem which is related to the substantial increase in the number of qubits needed for any calculation involving error correction. For none of the currently proposed systems is trivial to find a design capable of handling enough number of qubits. However, after the announcement of Google and NASA, it seems that some (if not many) of these problems are on the way of being resolved. In fact, in an article published by Ying in 2010 [33], some potential "meeting points" between quantum computing and artificial intelligence are discussed. Among these, we found:

- Quantum algorithms for learning
- Quantum algorithms for decision-making
- Quantum search
- Quantum game theory
- Semantic analysis
- Natural language
- Quantum Bayesian networks
- Quantum neural networks
- Quantum genetic algorithms

In the particular opinion of the author of this paper, and at the moment this paper has been conceived, Machine learning could definitely benefit from quantum computing. Just to justify the above statement: in machine learning it comes the definition and implementation of processes by means of which computers are able to adapt to the environment and to perform some tasks that, until then, they did not know how to do. In some cases the computer does all the work of learning without human supervision, but in others the system must be adjusted to achieve a better performance. A new paradigm emerges in this field to apply quantum computing. Unlike classical computing, the quantum phenomenon gives the computer the ability to use simultaneously all the problem data. This is why we believe that, by now, the best applications of quantum computing in artificial intelligence will be found in the field of machine learning.

However, it is clear that nowadays the link between quantum computing and artificial intelligence is more a guess than a reality. There are several arguments for this last statement: (1) quantum computing is still an emerging discipline; (2) artificial intelligence is an interdisciplinary field and several approaches (symbolic or sub-symbolic) can be undertaken; (3) problems in artificial intelligence are usually difficult to resolve since, essentially, they imply the construction of a 'model of human behavior'; (4) there is not a common language between quantum physicists and artificial intelligence researchers; and (5) it is not possible to run a quantum algorithm without a quantum computer and, although the technology is evolving fast, it seems that we have not yet a 'real quantum computer'.

Considering the described scenario: should we forget about the possibility of a fruitful collaboration between artificial intelligence and the quantum computing approach? In the particular opinion of the author of this paper the answer is 'no'. On the contrary, it seems that there are enough arguments to continue working on it and, although the technology is not yet ready, we have to achieve a better understanding, from an artificial intelligence perspective, about how to take advantages from all the wonderful things that quantum computing offers (i.e., quantum superposition, quantum parallelism, etc.). We are not far from having the required technology. In fact, Feynman (again) predicted that before 2050 we would have a computer that we could not even see [14]. In any case the theory is already here, and the fascinating possibilities of such a theory deserve, in our opinion, the attention of the artificial intelligence community.

8 Conclusion

We are going to conclude this 'position paper' recalling on the ideas we started with. In this respect, NASA's interest in quantum computing lies in trying to solve extremely complex problems in areas such as optimization of air traffic control, navigation, communication and robotics. Google, meanwhile, see in quantum computing the potential of solving high-level scientific computations in the area of machine learning. "Machine learning is about building better models of the world to make more accurate predictions", wrote Hartmut Neven, Google Engineering Director Research, in his blog. "If we want to cure diseases, we need better models about how they develop, if we want to create effective environmental policies, we need better models of what happens with our climate, and if we want to build a more useful search, we need to know what's on the Web in order to get the best response" [21]. In the opinion of the author of this article, the potential of quantum computing in artificial intelligence will be evident soon, but still we do not know how to translate that potential into reality. Undoubtedly, time will put things in place.

Acknowledgments The author of this article graduated in chemistry, specializing in physical chemistry, in July 1984, at the University of Santiago de Compostela, Spain. There he had the great fortune to be

a student of a professor who was able to teach him how beautiful it is to understand the intricacies of Nature. This professor is Don Julio Casado Linarejos. Life decided to take me to the Computer Science and Artificial Intelligence fields. However, I want to dedicate this work to Professor Casado ... I think he knows why.

References

1. Barenco, A., Benett, C.H., Cleve, R., Divincenzo, D.P., Margolus, N., Shor, P.W., Sleator, T., Smolin, J., Weinfurter, H.: Elementary gates for quantum computation. Phys. Rev. A **52**, 3457–3467 (1995)
2. Bennett, C.H.: Logical reversibility of computation. IBM J. Res. Dev. **17**, 525–532 (1973)
3. Bennett, C.H.: The thermodynamics of computation: a review. Intern. J. Theor. Phys. **21**, 905–940 (1982)
4. Bennett, C.H., Landauer, R.: Fundamental physical limits of computation. Sci. Am. **253**, 48–56 (1985)
5. Bennett, C.H., Shor, P.W.: Quantum information theory. IEEE Trans. Inf. Theory **44**, 2724–2742 (1998)
6. Bennett, C.H., Divincenzo, D.P.: Quantum information and computation. Nature **404**, 247–255 (2000)
7. Brain Basics: Know Your Brain. http://www.ninds.nih.gov/disorders/brain_basics/know_your_brain.htm (From Internet March 2014)
8. Brandom, R.: A first look inside google's futuristic quantum lab. http://www.theverge.com/2013/10/10/4824026/a-first-look-inside-googles-secretive-quantum-lab (From Internet March 2014)
9. Desurvire, E.: Classical and quantum information theory. Cambridge University Press, Cambridge (2009)
10. Deutsch, D.: Quantum theory, the Church–Turing principle and the universal quantum computer. Proc. Royal Soc. Lond. **A400**, 97–117 (1985)
11. Deutsch, D., Jozsa, R.: Rapid solution of problems by quantum computation. Royal Soc. Lond. Proc. Ser. A **439**, 553–558 (1992)
12. D-Wave, disentangled: google explains the present and future of quantum computing. http://www.extremetech.com/extreme/177316-d-wave-disentangled-google-explains-the-present-and-future-of-quantum-computing (From Internet March 2014)
13. Feynman, R.P.: Simulating physics with computers. Int. J. Theor. Phys. **21**, 467–488 (1982)
14. Feynman, R.P., Pines, D., Hey, A., Hey, J.G., Allen, W.: Feynman Lectures On Computation. Westview Press (1996)
15. Fredkin, E., Toffoli, T.: Conservative logic. Int. J. Theor. Phys. **21**, 219–253 (1982)
16. Gardner, H.: Multiple intelligences: the theory in practice. Basic Books, New York (1993)
17. Grover, L.K.: A fast quantum mechanical algorithm for database search. Los Alamos Physics Preprint Archive. http://xxx.lanl.gov/abs/quant-ph/9605043 (1996)
18. Grupo De Computación Cuántica, Departamento De Matemática Aplicada, E.U. Informática, U. Politécnica Madrid, "Introducción Al Modelo Cuántico De Computación", Technical report N^{o} 19 (2003)
19. Landauer, R.: Irreversibility and heat generation in the computing process. IBM J. Res. Dev. **5**, 183–191 (1961)
20. Levine, I.N.: Quantum Chemistry, 7th edn. Pearson Education (2013)
21. Neven, H.: Launching the quantum artificial intelligence lab. http://googleresearch.blogspot.com.es/2013/05/launching-quantum-artificial.html (From Internet March 2014)
22. Nielsen, M.A., Chuang, I.L.: Quantum computation and quantum information. Cambridge University Press, Cambridge (2000)
23. Poon, C., Zhou, K.: Neuromorphic silicon neurons and large-scale neural networks: challenges and opportunities. Front. Neurosci. doi:10.3389/Fnins.2011.00108 (2011)
24. Quantum Artificial Intelligence Lab. http://en.wikipedia.org/wiki/quantum_artificial_intelligence_lab (From Internet March 2014)
25. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. Los Alamos Physics Preprint Archive. http://xxx.lanl.gov/abs/quantph/9508027 (1994)
26. Shor, P.W.: Why haven't more quantum algorithms been found? J. ACM **50**, 87–90 (2003)
27. Sicart, A., Elkin, M.: Algunos Elementos Introductorios Acerca De La Computación Cuántica. Departamento De Ciencias Básicas. Universidad EAFIT. Medellín, Colombia. Junio de 1999
28. The Babbage Engine. http://www.Computerhistory.Org/Babbage/ (From Internet March 2014)
29. Toffoli, T.: Reversible computing. MIT Technical Report MIT/LCS/TM-151 (1980)
30. Vance, A.: Intel says Adios to Tejas and Jayhawk chips. http://www.theregister.co.uk/2004/05/07/intel_kills_tejas/ (From Internet March 2014)
31. Wichert, A.: Principles of quantum artificial intelligence. World Scientific, New York (2013)
32. Yanofsky, N.S., Mannucci, M.A.: Quantum computing for computer scientists. Cambridge University Press, Cambridge (2008)
33. Ying, M.: Quantum computation, quantum theory and AI. Artif. Intel. **174**, 162–176 (2010)