When Turing Meets Deutsch: A Confrontation between Classical Computing and Quantum Computing

Ellie D'Hondt[1], Maja D'Hondt[2], and Theo D'Hondt[3]

[1] Center Leo Apostel (CLEA)
[2] System and Software Engineering Lab (SSEL)
[3] Programming Technology Lab (PROG)
Vrije Universiteit Brussel, Belgium,
eldhondt | mjdhondt | tjdhondt@vub.ac.be

1 Introduction

After a blitz evolution of about fifty years, computer science would today seem to be a stable discipline without – some would say – any real new challenges. All solvable problems are solved, some problems turned out to be far more complicated than expected and NP is still its old obnoxious self. Computer science research consists in some people's minds of finding new permutations of existing nuggets of knowledge. The main concern of computer science teaching would seem to be instilling undergraduate students with a minimum of historical consciousness so that they do not go around eternally reinventing the wheel.

This bleak – and obviously exaggerated – view on the field of computer science does however highlight one particular issue: is the future of computer science a smooth extrapolation of the past or is there some revolutionary new technology waiting around the corner to turn everything upside down? An adventurous streak – present in every human being – makes us root for the second option; and if we had to place bets there seems to be only one choice: the quantum.

The past ten years have seen quantum computation evolve from science fiction into something tangible: theory is being confirmed by experiment and extremely modest trials are actually conclusive. These limitations – far from dampening our fantasy – inspire us to imagine full-fledged quantum computers – and of course the software and associated programming paradigms for these machines.

Which brings us to the theme of this paper: if quantum computation is to be taken seriously, shouldn't we start thinking about bringing theoretical physics and computer science much closer together than they are today? What about the obsession with formalism of the former and the complacency of the latter? We would like to propose a gedanken experiment: confronting a theoretical physicist and a computer scientist, and exposing them to the qubit through queries from an historically conscious veteran.

The following two sections - an introduction to quantum computing and a close inspection of a quantum search algorithm - try to accomplish two goals. First of all there is the obvious goal of introducing the state of the art in quantum computing. Additionally, we try to communicate some results of our experiment which mostly involved knowledge transfer from the physicist to the computer scientist and vice versa. In this process we identified correspondences between the two disciplines as well as concepts that are hard to grasp and explain. These insights are used in the next two sections to cut some corners and provide a tailored briefing on quantum computing for computer scientists. In the last section we express some conclusions about our experiment.

2 Getting to Know Each Other

2.1 Qubits

The fundamental concept of quantum computing is the *quantum bit* or *qubit*, a counterpart of the bit in classical computing. In the same vein, the state of a qubit can be $|0\rangle$ or $|1\rangle$, but unlike classical

bits a qubit cannot be measured or "accessed" to determine its exact quantum state. When accessing a qubit we acquire much more limited information: we get either $|0\rangle$ with probability $|\alpha|^2$ or $|1\rangle$ with probability $|\beta|^2$ where α and β are complex numbers. In fact, before accessing a qubit its state is a superposition of $|0\rangle$ or $|1\rangle$: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where the *normalization condition* $|\alpha|^2 + |\beta|^2 = 1$ must hold. When accessing a qubit, a transformation takes place which collapses its superposition state (two dimensions) to one of the computational basis states (one dimension, hence the term collapse). Up to a certain point , this process can be compared to *lazy evaluation* in classical computation, where the computation of a value is delayed until the value is actually needed, at which point its computation is forced. There is a difference however: in a pre-accessed qubit all possible values are actually contained, whereas in classical computing an intensional representation is provided from which the value can be calculated.

An interesting question is how much information a qubit can contain. Apparently, if a computation is performed on a qubit without accessing it, all possible values are being kept track of, resulting in an enormous amount of hidden information. In other words, superpositions intrinsically allow an implementation of parallel computation. When accessing the qubit a certain result is forced with a probability determined by the coefficients, but it is not possible to know beforehand with certainty what the outcome will be. So the trick in quantum computing is not only to manipulate the (possible values of a) qubit in order to obtain results of a certain computation, but also to adapt the probabilities corresponding to those values to improve the chance that the desired result is forced.

As in classical computing, quantum computing uses multiple qubits resulting in 2^n possible states if n is the number of (qu)bits used. A two-qubit system results in the four possible computational basis states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$. A crucial difference with classical computing, however, is that a pair of qubits can also exist in superpositions of these four basis states, associating a (complex) coefficient with each basis state where again the normalization condition must hold. Therefore, the number of possible quantum states increases exponentially with the number of qubits.

Because a quantum state is a combination of basis states, a computer scientist's reflex is to create a structure which holds the basis states, instead of an superposition containing intricate coefficients. Some kind of probabilistic mechanism should be associated with such a structure to force the appropriate result. However, the coefficients do not only denote the probability that a certain basis state will be the result, they also indicate the nature of the internal organisation of the quantum state. In other words, they reveal the exact relationships between the basis states, such as partial or full entanglement, which will be explained in the next subsection.

Although qubits are inherently weird - from a computer scientist's point of view and even from most physicists' - they are undoubtedly real since their existence and behaviour have been confirmed by experiments. Examples of realizations of qubits are the two polarizations of a photon, the alignment of a nuclear spin in a uniform magnetic field, or two states of an electron orbiting a single atom.

2.2 Non-Determinism and Entanglement

Turing machines [...] are those quantum computers whose dynamics ensure that they remain in a computational basis state at the end of each step, given that they start in one.
– David Deutsch [1]

The two main characteristics of a quantum computer are non-determinism and non-locality. The latter is also referred to as entanglement, the term we will continue to use in this paper. The previous subsection on qubits and their quantum states (i.e. superpositions which are forced when accessed) shows the origin and possibilities of the non-deterministic nature of a quantum computer.

Entanglement, however, is altogether much more difficult to explain. A system of n qubits is entangled if and only if its representation with respect to basic states of dimension n cannot be

split up in a product[1] of (1-dimensional) qubits. Thus, we have the following possible levels of entanglement, illustrated by an example:

- no entanglement at all
  for example $\frac{|001\rangle+|011\rangle}{\sqrt{2}} = |0\rangle\frac{|0\rangle+|1\rangle}{\sqrt{2}}|1\rangle$ which is not entangled, since we can split it up in three 1-qubit products;
- partial entanglement
  for example $\frac{|001\rangle+|010\rangle}{\sqrt{2}} = |0\rangle\frac{|01\rangle+|10\rangle}{\sqrt{2}}$ is *partially* entangled, because we can split of the first qubit, but not the other two;
- full entanglement
  for example $\frac{|001\rangle+|110\rangle}{\sqrt{2}}$ is *fully* entangled, because we cannot partition this three-qubit state into a product of lesser-qubit states.

To shed some light on the repercussions of entanglement, consider that the above fully entangled state has the property that upon accessing the first qubit two possible results are obtained: 0 with probability 1/2 resulting in the state $|001\rangle$, and 1 with probability 1/2 resulting in the state $|110\rangle$. Thus, accessing the second qubit will give the same result as accessing the first qubit, and accessing the third qubit will give the opposite result as accessing the first qubit. Therefore, the accessing outcomes are *correlated*. A way to view this is that entanglement makes a horizontal crosscut of all the parallel threads in the qubit system, because local information of one thread leads to information of the other threads - using the correct operations. We would like to remark that we completely agree with Merriam-Webster's definition of "entanglement": something that confuses; or of "to entangle": to involve in a perplexing or troublesome situation, to make complicated.

To fully grasp entanglement a minimal knowledge of the mathematical foundation, in other words linear algebra, and quantum mechanics would be useful. Given the current evolution of the curriculum of computer science education, it will get increasingly difficult for computer scientists to contribute in a significant way to quantum computing.

3 Holding Hands

In order to grasp quantum computation and its obvious and less obvious concepts we will work out a specific case of Grover's quantum search algorithm [2]. As of today, the quantum search algorithm is one of the main results of quantum computation, next to Shor's factoring algorithm [4]. Moreover, it is an application of quantum computing with which a typical computer scientist is well acquainted. While the search algorithm does not provide that an impressive speedup over classical algorithms – $O(n)$ to $O(\sqrt{n})$ as opposed to an exponential speedup for the factoring algorithm – we felt it was more convenient to portray it here, being closer to the framework of a typical computer scientist.
Grover's search algorithm has been analysed and discussed over and over again, and it is not our goal to provide yet another permutation of previously noted results. In this section, one should view the search algorithm as a means and not a goal, one by which we can efficiently transfer quantum computation knowledge to a computer scientist.
In order not to clutter our thoughts unnecessarily with aspects such as input representation we chose a very simple instance of the search algorithm: finding a non-trivial factor of 35 which is less than 8. Note that a factoring problem can be solved much more efficiently through the factoring algorithm, but again we stress that the point is not to explain the algorithm nor efficiency matters, but to bestow the (computer science) reader with a clear idea of quantum computation and its main differences with classical computation.

Problem: Find a non-trivial factor of the integer 35 which is less than 8.
In order to represent all integers less than 8 we work with 3 qubits, such that the resulting $2^3 = 8$ computational basis states represent the integers less than 8 in binary form. The algorithm proceeds

[1] Actually, qubits are combined through the *tensor* product.

through the following states:

easy start: $|\psi_1\rangle = |0\rangle|0\rangle|0\rangle = |000\rangle$
We start out in a simple, easy to construct basis state.

allow parallel computation:
$|\psi_2\rangle = H^{\otimes 3}|\psi_1\rangle = \frac{1}{\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle)$
By applying the Hadamard gate[2] to each qubit separately we obtain an equally weighted superposition of states. This is exactly the functionality of the Hadamard gate: it creates superpositions, which are inherently nondeterministic and allow parallel processing.

differentiate solutions:
$|\psi_3\rangle = O|\psi_2\rangle = \frac{1}{\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle - |101\rangle + |110\rangle - |111\rangle)$
The oracle O differentiates between those basis states which represent solutions to the search problem and those that do not. In this case, in other words, it filters out those integers which are non-trivial factors of 35. We will not concern ourselves overly with the actual implementation of the oracle and instead view it as a black box - however, for suspicious readers, let us note that one can easily adapt the classical circuit for doing trial division to an efficient (quantum) oracle with the above functionality [3]. One could view this step in a way as *entangling* the solutions with one another – after all, the solutions obtain a correlation through their coefficient, i.e. the minus sign. However, while it is true that they become correlated, this is in a fundamentally different (classical) way as with quantum entanglement.

approach solutions: $|\psi_4\rangle = IAM(|\psi_3\rangle) = \frac{2}{\sqrt{8}}(|101\rangle + |111\rangle)$
One may wonder why we go through all the trouble of calculating and dragging around all these annoying and superfluous-seeming coefficients. However, these are crucial to the search algorithm, since the key idea is to manipulate them in such a way that the coefficients of all terms that are solutions approach 1 while those of the terms that are not solutions approach 0. This is accomplished through the *inversion about mean* transformation IAM, which is defined – rather ad hoc – by $IAM(\sum_k a_k|k\rangle) = \sum_k(-a_k + 2\text{average}(a_k))|k\rangle$. Applying this transformation distributes all terms in the superposition state in the desired way, as can be understood more clearly from figure 1. This results in the state $|\psi_4\rangle$, which, when measured, collapses with probability 1 to a solution of the posed problem!

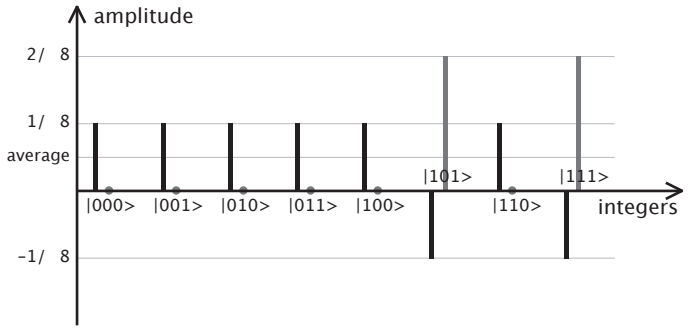

Fig. 1. Amplitudes before (black) and after (grey) the application of the IAM transformation. After this last step in the algorithm a solution is measured with probability 1.

[2] Transformations in quantum computation are also referred to as *gates*

We should note that in general the last two steps are to be iterated several times, such as to increase the probability to measure a solution with each iteration.

4 The Marriage Proposal

Witnessing a dialogue between a theoretical physics and a computer science Ph.D. student reveals a number of interesting facts, particularly in what one would expect to be a common area of interest: quantum computation. First of all: a minimal common language is required. On the one hand, a physicist should not consider that the Universal Turing Machine is the be all and end all of classical computing; programming paradigms and their expressiveness are all too often ignored. Our discussions revealed the distance that needs to be covered in order to abstract the features of quantum computers into suitable high level programming constructs. On the other hand, quantum mechanics (and hence quantum computation) is to such an extent counterintuitive that it can only be managed in a rational fashion by applying a rigorous mathematical formalism. Unfortunately, we live in an age where theoretical physics through inertia continues to be a stranger to computer science and where computer science is moving away from mathematics and other formal methods. Unless this evolution is reversed we will end up having fabulous quantum computers and nobody able to program them.

Our discussions also revealed that quantum computation in its current state can barely cope with a limited number of isolated, handcrafted algorithms. Their authors used their intimate knowledge of quantum phenomena to leapfrog the abstraction levels (that we are used to in classical computation) between the physical and the conceptual world. Surprisingly, whereas these algorithms systematically exploit the non determinism of quantum state, the same cannot be said of entanglement. That is, while entangled states appear frequently in existing quantum algorithms, it is not at all clear whether and how the fact that they are entangled matters. Unfortunately, the notion of a entangled state (which even to the non-expert eye is at least as important a feature as non-determinism) is visibly extremely difficult to transfer from a quantum physicist to a computer scientist. Again a general formalism covering the two disciplines is missing.

Our experiment was of course limited in time and scope. However it was very enlightening for all three parties involved. One need not be a genius to see the benefits of having the field of quantum physics and computer science establish a dialogue. However, the lack of common ground is disappointing and the realization that it is shrinking is frankly disquieting. Our experiment very clearly reveals that we are probably closing in on a revolutionary new technology and that we are ill prepared. One century after the discovery of quantum phenomena, we will probably see the first real quantum computers. However – although not as much as in the beginning – the quantum still is elusive, even to the specialist. It is high time that we stop the tendency of computer science to move away from the other sciences. It does not seem too improbable that future computer scientists will need to be familiar with (the formal model of) quantum mechanics. In that case we will need to review our educational system in a significant way.

References

1. David Deutsch. Quantum theory, the Church-Turing principle and the Universal Quantum Computer. *Proc. R. Soc. Lond.*, pages 97–117, 1985.
2. Lov K. Grover. A fast quantum mechanical algorithm for database search. In *ACM Symposium on Theory of Computing*, pages 212–219, 1996. classic paper.
3. Michael A. Nielssen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
4. Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *IEEE Symposium on Foundations of Computer Science*, pages 124–134, 1994. classic paper.