Beyond Quantum Computers

G. Chiribella*, G. M. D'Ariano*, P.Perinotti*, B. Valiron†

**QUIT Group, Dipartimento di Fisica "A. Volta", via Bassi 6, 27100 Pavia, Italy*
†LIX/INRIA, École Polytechnique, 91280 Palaiseau, France

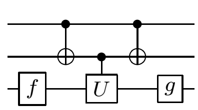
Quantum Computers are the computers of the future. They will be capable of computational tasks that are unfeasible by the "classical" computers that we use everyday. A quantum computer processes quantum bits (*qubits*), whose value can be not only 0 and 1 as for a classical bit, but also every possible "superposition" of 0 and 1.

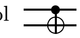
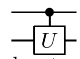
The "information" that is processed by this new type of computers is "quantum". It is made of *entanglement*—the most elusive feature of Quantum Mechanics. Entanglement provides correlations between separate systems that we cannot understand in classical terms as if they were due to unknown fluctuating local variables. For this reason Einstein used to address entanglement as *spooky action at a distance*. It is from the magic of entanglement and of superposition that the power of quantum computers comes from.

Quantum information processing, however, is not the ultimate physical model of computation that we can conceive in our quantum world. A computation always transforms an input into an output, but these do not have to be necessarily qubits: one can e.g. consider a computation where the input is a physical transformation provided as a black box, and the output is also a transformation, obtained from the input black box by means of suitable physical operations. This kind of "higher-order" quantum computation includes the basic quantum information processing as a special case and is potentially more powerful. Here we show that there are computations that are admissible in principle—i.e. their existence does not lead to any paradoxical or unphysical effect—and yet cannot be realized by a usual quantum circuit. In order to implement this new kind of computations one needs to change the rules of quantum circuits, also considering circuits where the geometry of the connections can be itself in a quantum superposition.

What is a quantum computer? And what does it compute? The first quantum computational model was the Quantum Turing Machine introduced by Deutsch in 1985 [1] in analogy with the classical Turing machine. Quantum Turing machines however were not very intuitive to deal with. Few years later, an alternative model was presented, namely the quantum circuit model [2], in which the computation is described as a sequence of transformations (logical gates) acting on a register of input qubits. The quantum circuit model was then proved to be equivalent to the quantum Turing machine for a class of computations that can be described as processing of input qubits [3]. Since then the quantum circuit model has grown definitely more popular, due the discovery of powerful quantum algorithms, like Shor's one for factoring integers in polynomial time, or Grover's one for searching a database of size N in $\sqrt{N}$ steps, which have been invented in the framework of quantum circuits.

Let us start by illustrating quantum circuits and their basic rules in a simple example:



Here each wire is drawn in space, but the path from left to right in the circuit does not represent a path in space: instead, it is the time evolution of a qubit from past to future. In the above example the boxes $\boxed{f}$ and $\boxed{g}$ implement a quantum processing on a single qubit, e. g. a binary function. The symbol  is a C-NOT (*controlled-not*) transformation: this transforms two qubits *jointly*, with the target qubit (wire with $\oplus$) which undergoes the identity transformation if the control-qubit (wire with $\bullet$) is in the state $|0\rangle$ and the NOT transformation $|0\rangle \leftrightarrow |1\rangle$ if the control-qubit is in the state $|1\rangle$. When the control-qubit is in the superposition state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ (and the target is in the state $|0\rangle$) the two qubits at the output become entangled in the state $\frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle)$. The symbol  is a C-U (*controlled-unitary*), a generalization of the control-not, with the transformation U replacing the NOT transformation of the C-NOT.

It is worth stressing that the quantum circuit is a *computational* circuit—not a physical one: while in the physical circuit we can have loops (for example when a system passes twice through the same physical device), in the computational circuit there are no loops (when we apply twice a transformation to the same system we just draw two times the same box). The computational circuit represents the actual flow of information during the run of a "program". It is also important to make clear the distinction between *program* and computational circuit, the former being a set of instructions to build up the latter. In the computational circuit the "wires" can never go backward, because this would mean to go *backward in time*, whereas, on the contrary, in the program code we can have commands pointing back to a previous instruction.

The framework of quantum circuits is used in quantum computer science to evaluate the amount of computational resources used in an algorithm (*e. g.* number of oracle calls, number of qubits, length of the computation, etc.). We summarize here few basic rules that characterize ordinary quantum circuits and the associated resource counting. From now on, a circuit satisfying this set of rules will be referred to as a computational circuit.

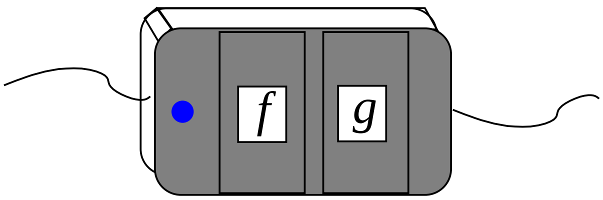
1. qubits are represented by wires;
2. a box on a single wire represents a transformation on the corresponding system, whence on multiple wires generally describes an interaction between the corresponding systems;
3. input/output relations proceed from left to right and there are no loops in the circuit;
4. each box represents a single use of the corresponding transformation.

As already mentioned, this framework has been a fertile ground for the development of quantum algorithms. In most algorithms the input data are encoded in a unitary transformation performed by a black box (the *oracle*), which is called as a subroutine during the computation. This is the case e.g. of Deutsch-Josza, Simon's, Shor's, and Grover's algorithms, to give just a few examples. The core of all this algorithms can be described as a "computation" that takes as input a certain number of calls to the oracle, and returns as output some classical data, like the period of a function, or the prime factors of a number. Despite the fact that the input data are encoded in a black box, however, all these algorithms are realized as the evolution of qubits through a quantum circuit which simply *contains* the available black boxes as elements. Is this a general rule? Do quantum circuits allow for the computation of all possible functions whose input is a box, rather than a qubit register? This question is inspired by Church's notion of computation [4], which allows one to compute functions of functions, rather than only functions of bits. We now show that the answer to the question is negative: *i)* there exist functions of boxes that are clearly computable (they are achieved by means of elementary operations), but their computation cannot be represented by a circuit obeying rules 1-4. Moreover, *ii)* there are functions that might be computed in principle without leading to any unphysical effect, which however cannot be achieved by an ordinary quantum circuit.

The key counterexample for point *i)* is provided by the following function of boxes $\boxed{f}$ and $\boxed{g}$, that depends on a control bit x:

$$S\left(x,\boxed{f},\boxed{g}\right)=\begin{cases} -\boxed{f}-\boxed{g}- & x=1\\ -\boxed{g}-\boxed{f}- & x=0\end{cases} \qquad (1)$$

The two boxes $\boxed{f}$ and $\boxed{g}$—along with the classical bit x—are the *input* of the function, and must be regarded as *single* calls to two different oracles during the computation. The above example can be generalized in various ways, for example by putting between f and g a third box $\boxed{U_x}$ that depends on the value of the bit x, or by leaving between f and g an open slot in which a third arbitrary transformation can be inserted. It is easy to imagine a physical device that implements the function S. Consider a machine with two slots, in which the user can plug two *variable* boxes $\boxed{f}$ and $\boxed{g}$ at his choice, as in the following figure.



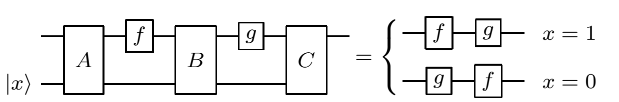
The machine is programmed with the following code:

```
PROGRAM "SWITCH"
if x = 1
  then
    do —[f]—[g]—
  else
    do —[g]—[f]—
endif
```

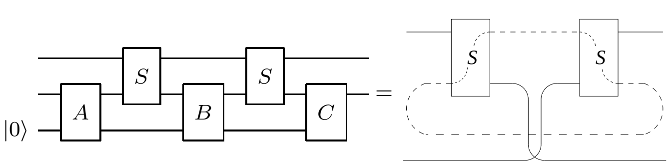
We can imagine that the machine has movable wires inside, that can connect the boxes $\boxed{f}$ and $\boxed{g}$ in two possible ways depending on the value of the classical bit x, thus implementing the SWITCH function. Ordinary quantum circuits, however, don't have such movable wires. They can have controlled swap operations, but once a time-ordering between $\boxed{f}$ and $\boxed{g}$ has been chosen in the circuit, there is no way to reverse it:

Theorem 1 (No-switch of boxes). The program SWITCH cannot be achieved deterministically by a computational circuit in which $\boxed{f}$ and $\boxed{g}$ represent single calls to variable oracles.

Proof. Suppose by absurd that there exists a deterministic circuit performing the program SWITCH. Then we must have



where A B and C are deterministic sub-circuits (possibly including the preparation of ancillary qubits). By linearity, for $x=0$ the fixed circuit also locally switches bipartite boxes. In the case of two swap gates, for $x=0$ the output would be a circuit containing a time loop, represented by a dashed line in the following diagram:



By simple algebra one can verify that right-hand side does not preserve normalization of states. This is in contradiction with the fact that the left-hand side is the composition of deterministic boxes, and, therefore, it preserves normalization. ■

The proof shows that a computational circuit implementing the program SWITCH would allow one to build a closed time-like curve within a computational circuit, thus breaking rule 4. This fact is not a coincidence: in the following we will see that also the converse is true, namely a qubit in a time loop would allow one to build a computational circuit for the program SWITCH.

The program SWITCH is the prototype of a *higher-order computation* of the kind described in the λ-calculus by Church [4], with the input being a function instead of a block of data. As a consequence of Theorem 1, the higher-order computation represented by the program SWITCH cannot be implemented by a quantum circuit that contains only one use of $\boxed{f}$ and $\boxed{g}$ in a pre-defined order.

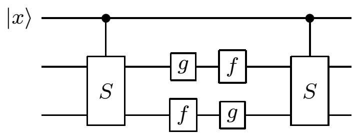
In fact, the realization of the program SWITCH by a computational circuit obeying rules 1-4 is impossible not only in the quantum world, but also in the classical one, where qubits are replaced by ordinary bits. However, in the classical case this realization problem arises only in distributed computation, when the input functions f and g are provided as "physical" machines connected in a circuit, rather than as sets of programming data defining two subroutines. Indeed, when functions are encoded into strings of bits, they can be processed by a circuit in the standard way. This is essentially the reason why in the classical case the Turing machine and the λ-calculus give rise to equivalent notions of computation (in particular, any higher-order function can be encoded on the tape of the Turing machine, which emulates the evaluation of the function).

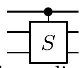
In the quantum case the equivalence between Church's and Turing's notions of computation is a much more delicate issue. A quantum version of λ-calculus was formulated in Ref. [6], and proved to be equivalent to the quantum Turing machine, in the same sense as in the classical case. However, this proposal treats only unitary boxes, and, for example, cannot describe the program SWITCH if the input functions f and g are not unitaries. Another quantum version of λ-calculus, presented in Ref. [7], describes quantum computation assisted by classical control, and, in particular, it is able to express the SWITCH function with arbitrary input boxes. Whether or not a quantum Turing machine can emulate all higher-order computations allowed by this language is still an open question.

It is worth stressing that the relevance of the no-switch theorem is independent of the problem of establishing a formal equivalence between different notions of computation: the main point here is the physical implementation of the SWITCH program as an actual transformation of physical boxes. From this point of view, having an abstract encoding of the functions f, g into quantum states $|f\rangle, |g\rangle$ is not a satisfying solution: one use of the unknown boxes $\boxed{f}$ and $\boxed{g}$ is a very different physical resource from one copy of the states $|f\rangle, |g\rangle$. Indeed, the conversion $\boxed{f} \leftrightarrow |f\rangle$ cannot be achieved in a physically reversible way, since if it were, it would violate the no-programming theorem [5].

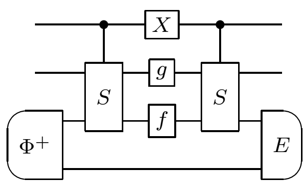
Focusing attention on the quantum circuit model, the origin of the problem in realizing the program SWITCH as a computational circuit obeying rules 1-4 is twofold. The first limitation arises from the fact that the oracles f and g are restricted to be called once, so that the circuit must contain boxes $\boxed{f}$ and $\boxed{g}$ only once (rule 4) and in a definite time order (rule 4). Indeed, a computational circuit that produces the *same output* of the program SWITCH actually exists, but it requires two calls to both oracles f and g, e. g.

as follows



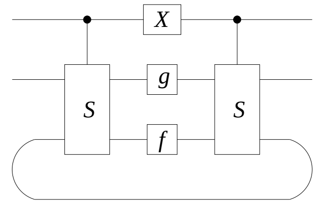
This circuit achieves the desired transformation over the qubit in the middle wire depending on the state of the controlling qubit at the top wire. Here  is a control-swap gate, exchanging the two input qubits depending on the state of the control qubit. On the other hand, if the input are two black boxes $\boxed{f}$, $\boxed{g}$, the possibility of achieving two uses from a single one is ruled out by the no-cloning theorem for boxes [8]. Again, the limitation due to the single call constraint is strictly related to the "physical" nature of the unknown black boxes $\boxed{f}$ and $\boxed{g}$. If we knew what f and g are, we would be able to duplicate them, thus making possible the computation of the function $S(x, \boxed{f}, \boxed{g})$ through the above circuit.

Another factor that prevents the implementation of the program `SWITCH` as a computational circuit is the requirement that the program succeeds deterministically. Indeed, rules 1-5 do not forbid achieving the task with some probability. In particular, a computational circuit that uses probabilistic teleportation succeeds in the task with probability 1/4

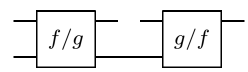


Here Φ^+ is a maximally entangled state of two qubits and E denotes the projection on Φ^+, which is a the outcome of a Bell measurement. When the outcome E occurs in this circuit, we may say that the third qubit (from the top) has been teleported from the future back to the past. In this case it is easy to see that if the control qubit is in state $|1\rangle$ one obtains the sequence "$\boxed{f}$ followed by $\boxed{g}$" acting on the second input qubit, while if the control qubit is in state $|0\rangle$ the boxes are exchanged. What's more, if one puts the control qubit in the superposition $|0\rangle + |1\rangle$ one would get the superposition of the two orderings of the boxes, namely the output of the circuit is proportional to $U_f U_g |\psi\rangle |1\rangle + U_g U_f |\psi\rangle |0\rangle$, where $|\psi\rangle$ is the input state of the qubit in the middle wire, and U_f and U_g denote the unitary operators corresponding to boxes $\boxed{f}$ and $\boxed{g}$, respectively. Note, however, that the probability of achieving the program `SWITCH` for $\boxed{f}$ and $\boxed{g}$ transforming N qubits goes to zero exponentially as 4^{-N} versus the number N of input qubits for each box.

If we artificially scale the projection E to achieve the `SWITCH` with unit probability of success, we introduce a loop in the circuit: The loop represents a qubit that travels backward in time, thus violating causality as expressed by rule 4. In a sense, this simple example is complementary to the results of Ref. [9], which showed that closed time-like curves do not improve tasks of first-order computation, like state discrimination. Here we have instead an impossible higher-order computation that would become realizable by a quantum circuit if a closed time-like curve were available. Note however, that the teleportation-based model of time travel considered here is different from the nonlinear model by Deutsch [10], which provided the framework for the results of Ref. [9].



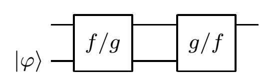
The question that we address now is: what rule in the theory of computational circuits can be modified in order to recover the physical implementation of the function $S(x, \boxed{f}, \boxed{g})$ of Eq. (1), whose computation is achieved through the program `SWITCH`? One possibility is to modify rule 4, and to allow for circuits containing certain time loops. However, introducing time travels in the model seems a rather drastic solution. A more moderate approach is to modify rule 4: In particular, we may assume that the resource provided by a single call to each of the two physical oracles—that would be separately described as $\boxed{f}$ and $\boxed{g}$—*in a causal succession that can be decided by the user*, is described in circuital terms as a single oracle with classical control:



where the wire on the bottom left denotes the control qubit, whose general state is $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$ with $|\alpha|^2 + |\beta|^2 = 1$. The input x is encoded on the state $|\varphi\rangle$ as follows: for $x = 0$ we prepare $|\varphi\rangle = |0\rangle$, for $x = 1$ we prepare $|\varphi\rangle = |1\rangle$. If the two qubits on the top lines are in the states ρ_1 and ρ_2, respectively, the action of the oracle is given by

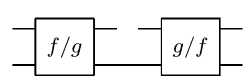
$$\begin{aligned}\mathscr{O}_{f,g}(|\varphi\rangle\langle\varphi| \otimes \rho_1 \otimes \rho_2) =& |\langle 1|\varphi\rangle|^2\ U_f\rho_1 U_f^\dagger \otimes U_g\rho_2 U_g^\dagger \\ &+ |\langle 0|\varphi\rangle|^2\ U_g\rho_1 U_g^\dagger \otimes U_f\rho_2 U_f^\dagger\end{aligned} \quad (2)$$

This way of representing the oracle is consistent with the basic properties that one expects for the resource, namely that it perform two successive transformations, one being a call of the box $\boxed{f}$ and the other a call of the box $\boxed{g}$, with the order of such calls can be controlled by the variable x encoded in the state $|\varphi\rangle$. During the time interval between the calls to the oracle, any transformation can happen, including evolutions transforming the first output into the second input. Exploiting the latter representation of the oracle one can clearly implement the program `SWITCH`, just by connecting the output of the first box with the input of the second one, and encoding the bit x in the state $|\varphi\rangle$ as follows



If we assume that the oracle of Eq. (2) translates the resource provided by a single use of the physical boxes corresponding to $\boxed{f}$, $\boxed{g}$ with classical control of the causal ordering, we can then consider the function $S(x, \boxed{f}, \boxed{g})$ as computable by a quantum circuit exploiting this resource.

Such an oracle can be achieved in practice, for example, by a physical circuit in which the connections between wires are movable, as in Fig. 1. While representing automated *classical control* of causal sequences of operations through the above oracle allows to recover the description of the program `SWITCH` within the computational circuit model, it leaves unanswered the question how *quantum control* of causal sequences of operations can be described. We can of course imagine a further generalization of the oracle, allowing for quantum control, with the control qubit that preserves coherence and becomes entangled with the causal ordering of boxes $\boxed{f}$ and $\boxed{g}$ as follows



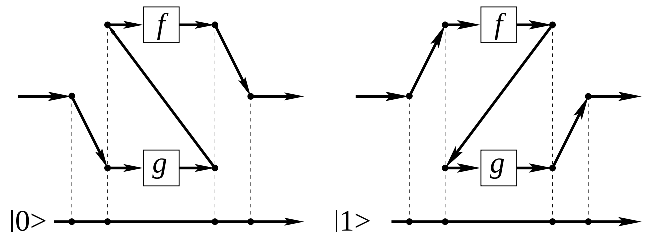


FIG. 1: Quantum machine with classical control over movable wires.

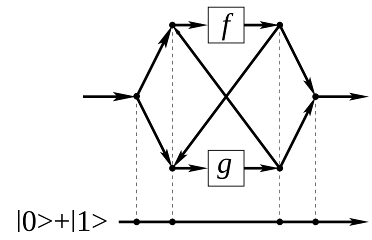


FIG. 2: Quantum machine with quantum control over movable wires.

The corresponding unitary operator is the following

$$W_{f,g} = |0\rangle\langle 0| \otimes U_f \otimes U_g + |1\rangle\langle 1| \otimes U_g \otimes U_f \tag{3}$$

The above construction can be suitably generalized when f and g are not unitary boxes, but noisy quantum channels: in this case, it is enough to use the above formula to define the Kraus operators of the channel with quantum control in terms of the Kraus operators of the input channels.

The oracle with quantum control is more general and more powerful than the classically controlled one introduced in Eq. (2). Indeed, having $W_{f,g}$ at disposal one can implement the classically controlled oracle $\mathcal{O}_{f,g}$ by using $W_{f,g}$ and then discarding the control qubit.

How can we build the controlled gate $W_{f,g}$ if we have at disposal one use of the black boxes for f and g? Again, this is a question that the circuit model is unable to answer. In principle, there is no physical reason to forbid the computability of the higher-order function defined by $W : f \otimes g \mapsto W_{f,g}$. This function is defined not only on product boxes, but also on the more general class of *non signaling* bipartite boxes, i.e. boxes for which the output state of each qubit is independent of the input state of the other. The function is linear in its argument, transforms deterministic boxes into deterministic boxes, and can also be applied locally to multipartite boxes without giving rise to unphysical effects like negative probabilities. The computation of this function is in principle admissible, according to the notion of admissibility originally developed in Ref. [11] for functions that are compatible with a pre-defined causal ordering of all quantum systems. Here, although the computation of $W_{f,g}$ is compatible with quantum mechanics, it cannot be implemented by a circuit with the rules 1-4, due to the lack of a pre-defined causal ordering. Moreover, it is also possible to prove that no circuit using the oracle with classical control $\mathcal{O}_{f,g}$ can simulate the oracle with quantum control $W_{f,g}$.

To imagine a way to build up the controlled gate $W_{f,g}$ from the boxes $\boxed{f}$ and $\boxed{g}$, we need to go beyond the usual language of quantum circuits, and to consider also circuits with movable wires that can be also in quantum superpositions. For example, we can consider a thought experiment where the physical circuit with movable wires depicted in Fig. 1 can be controlled by a qubit in a way that preserves superpositions, with the control qubit interacting with switches and controlling them in a correlated way, as represented in Fig. 2. Like in the Schrödinger cat thought experiment, in this case we would have a mechanism producing entanglement between a microscopic system (the control qubit) and a macroscopic one (the position of the switches). Notice however that quantum control of transformations is even more powerful than quantum entanglement, which is the feature giving rise to the classical Schrödinger cat experiment. Indeed, a control-unitary gate can be always used to generate a certain amount of entanglement. It is worth stressing that the subcircuit described by the oracle with quantum control has not to be meant as describing operations performed by a human observer: This would sound highly paradoxical, since in such a case, we would have to cope not just with cats in a superposition of dead/alive, but, even more dramatically, with operators whose free will can be in superposition of taking decision A or decision B.

The open question now is whether quantum control over the geometry of the connections in the circuit is enough to physically implement all possible functions of boxes that are compatible with Quantum Mechanics.

A circuit model in which the states of quantum systems can control the structure of a causal network immediately suggests the analogy with a quantum gravity scenario, in which the spacetime geometry can be entangled with the state of physical systems. We argue that exhaustive analysis of higher-order transformations in quantum mechanics will provide some insight in the relation between quantum mechanics and general relativity, within a framework similar to that of Ref. [12]. Moreover, the theory of higher-order quantum computation with quantum control will provide a computational model—which should be formalized by a suitable generalization of quantum λ-calculus with classical control [7]—that could possibly describe a wider range of computations compared to the Quantum Turing Machine operating only at first-order, thus breaking the equivalence between Church's and Turing's notions of computation [13].

Besides the problem of abstract computational equivalence, the physical implementation of higher-order functions discussed has an interesting relation to the paradigm of the universe as a quantum computer [14]. Indeed, one can wonder what kind of quantum computer the universe is: It could be a gigantic quantum circuit, or a quantum Turing machine, or also be a higher-order computer, that processes information encoded in transformations (e.g. in scattering amplitudes) rather than in states. Even if these three models turned out to be equivalent from an abstract computational point of view, they would nevertheless remain very different from the physical one, as they are based on different physical mechanisms. Moreover, as we already mentioned, the third model has still to be completely formulated: what is presently lacking is a complete physical theory that specifies all transformations of boxes that are possible in nature. A piece of Quantum Mechanics still needs to be written.

In conclusion, after summarizing the main rules of computational circuits, we exhibited a higher-order function—namely a function of physical boxes—that is computable by elementary operations but whose computation cannot be described by a quantum circuit obeying the usual rules. We proposed a minimal change of the rule for describing the oracles, introducing classical control of causal sequences of operations, in such a way that the computation of the class of higher-order functions including the `SWITCH` can be expressed in circuital terms. We then discussed a further level of generality, accounting for quantum control of the causal sequence of operations. A complete physical theory of higher-order computation has not been developed yet, we expect it to reveal unexplored aspects of quantum theory in a non-fixed causal framework.

We wish to thank P. Selinger for stimulating criticisms and discussions, during which he independently devised the realization of the `SWITCH` program by a machine with movable wires. This work was supported by EC through the project COQUIT.

[1] D. Deutsch, *Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer*, Proc. Roy. Soc. Lond. A **400**, pp. 97-117 (1985).

[2] D. Deutsch, *Quantum Computational Networks* Proc. Roy. Soc. Lond. A **425**, pp. 73-90 (1989).

[3] Andrew Yao *Quantum circuit complexity*, Proceedings of the 34th Annual Symposium on Foundations of Computer Science, 352 (1993).

[4] H. Barendregt, *Lambda Calculi with Types*, in *Handbook of Logic in Computer Science, Volume 2: Computational Structures*, S. Abramski, D. M. Gabbay and T. S. E. Maibaum eds., (Oxford University Press, New York, 1993).

[5] M. A. Nielsen and I. L. Chuang, Phys. Rev. Lett. **79**, 321 (1997).

[6] A. van Tonder. *A lambda calculus for quantum computation*, SIAM Journal of Computing, **33** 1109 (2004).

[7] P. Selinger and B. Valiron, *A lambda calculus for quantum computation with classical control*, Math. Struct. in Comp. Sci., **16** 527 (2006)

[8] G. Chiribella, G. M. D'Ariano, P. Perinotti, *Optimal Cloning of Unitary Transformation*, Phys. Rev. Lett. **101** 180504 (2008).

[9] C. H. Bennett, D. Leung, G. Smith, and J. A. Smolin, *Can Closed Timelike Curves or Nonlinear Quantum Mechanics Improve Quantum State Discrimination or Help Solve Hard Problems?*, Phys. Rev. Lett. **103**, 170502 (2009).

[10] D. Deutsch, *Quantum Mechanics near Closed Timelike Lines* Phys. Rev. D **44**, 3197 - 3217 (1991).

[11] G. Chiribella, G. M. D'Ariano, and P. Perinotti, *Theoretical Framework for Quantum Networks*, Phys. Rev. A **80**, 022339 (2009).

[12] L. Hardy, *Towards quantum gravity: a framework for probabilistic theories with non-fixed causal structure*, J Phys A: Math. Theor. **40**, 3081-3099 (2007).

[13] B. J. Copeland, *The Church-Turing Thesis*, in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta ed., (Fall 2008 Edition).

[14] S. Lloyd, *Programming the Universe: A Quantum Computer Scientist Takes On the Cosmos* , (Alfred A. Knopf, NewYork, 2006).