Towards a Multi Target Quantum Computational Logic

Giuseppe Sergioli

Published online: 4 December 2018
© Springer Nature B.V. 2018

Abstract
Unlike the standard Quantum Computational Logic (QCL), where the carrier of information (target) is conventionally assumed to be only the last qubit over a sequence of many qubits, here we propose an extended version of the QCL (we call Multi Target Quantum Computational Logic) where the number and the position of the target qubits are arbitrary.

Keywords Quantum Computational Logic · Quantum circuit · Control and target qubits

1 Introduction
Both in classical and in quantum computation, a circuit is described in terms of a sequence of gates that transform an arbitrary state (input) into another state (output) (Kitaev et al. 2002). In classical computation these transformations are basically irreversible: the Boolean functions \( f \) that represent the classical gates are many-to-one, i.e. \( f : \{0, 1\}^n \mapsto \{0, 1\} \), where the dimension of the input state does not correspond to the dimension of the output. Hence, the same output may be obtained in correspondence with different inputs, producing an irreversible process. Anyway, this irreversibility is no longer essential. As proved by Toffoli (1980), through a very simple expedient any classical irreversible gate can be canonically converted into a respective reversible gate. Given an arbitrary irreversible gate, any input can be divided in two components: the first is the “genuine” input while the second is an ancilla (or target bit); on the other hand, also the output can be divided in a first component, which is a copy of the “genuine” input, and a second component that is obtained by the evolution of the ancilla. Basically, only the second part of the output contains the result of the computation, while the first part only contains redundant information, that represents a kind of memory of the input state. Using this expedient, the dimensions of the input and the output states are the same and the computation becomes reversible. The price to pay in order to obtain the desirable reversibility is an increasing of the dimension of the computational space.

Unlike classical computation, the theory of quantum computation is naturally reversible (Sergioli et al. 2011). The Schrödinger equation describes the dynamic evolution of quantum systems, showing how the state \( |\psi_{t_0}\rangle \) at the initial time \( t_0 \) evolves into another state \( |\psi_t\rangle \)
at the final time $t_1$ by the equation $|\psi_{t_1}\rangle \rightarrow |\psi_{t_1}\rangle = U|\psi_{t_0}\rangle$, where $U$ is a unitary operator that represents a reversible transformation. The Schrödinger equation is naturally applied also in quantum information theory (Nielsen and Chuang 2000). Indeed, quantum logical gates are unitary operators that describe the time evolution of a quantum input state into an output and to perform a quantum algorithm exactly means to apply a sequence of quantum gates to a quantum input state. Unlike the classical case, because of their unitarity, the dimensions of the input and the output of a quantum gate are the same and the quantum logical gate always represents a reversible transformation. Furthermore, another peculiar difference between classical and quantum information theory is given by the basic information quantity, that, in the classical framework, is stored by the classical bit while in quantum computation is given by the quantum bit (qubits for short), i.e. unitary vector in the complex Hilbert space $\mathbb{C}^2$ (Dalla Chiara et al. 2013; Nielsen and Chuang 2000). The qubits turns out to store a very larger amount of information with respect to its classical counterpart and this is the reason that makes, in principle, quantum computation more efficient with respect to the classical one. The input of a quantum circuit is given by a composition of qubits that is mathematically represented by the tensor product operation. Hence, given $k$ qubits $|x_1\rangle, |x_2\rangle, \ldots, |x_k\rangle \in \mathbb{C}^2$ the input state given by an ensemble of $k$ qubits is given by $|x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_k\rangle$ (that, for short, we call quantum register—or quregister—and we indicate by $|x_1x_2 \cdots x_k\rangle \in \otimes^k \mathbb{C}^2$). It is important to remark that, given the non-commutativity of the tensor product, the sequence in which any qubit appears in the state is not negligible; in other words, $|x_1x_2\rangle$ and $|x_2x_1\rangle$ generally represent two different states.

A quantum circuit is represented by the evolution of the input quregister under the application of some unitary quantum logical gates (Hirvensalo 2001; Nielsen and Chuang 2000). Obviously, it is often the case where a $n$-ary quantum gate $U^{(n)}$ is applied to $n$ qubits over a given input configuration, where these $n$ qubits can, in principle, be non-adjacent one another. This very common scenario is related to the topic of the architecture in quantum computer design, that plays a crucial role for the realization of advanced technologies in quantum computation (Linke et al. 2017). Even if physical architectures conveniently use particular constraints on the qubits distribution based on the nearest-neighbor couplings (Kumar 2013; Linke et al. 2017), in principle, these constraints have no incidence in the possibility to perform arbitrary computations, because the Swap operations can be suitably used; anyway, the use of the Swap operations is not free of any computational cost. On this basis, recent topics related to efficient quantum computing between remote qubits in nearest-neighbor architectures—such as the linear neighbor architectures (LNN) (Kumar 2013)—are active and important areas of research, also devoted to physical implementations (Fowler et al. 2004; Takahashi et al. 2007).\(^1\)

In addition, from a more theoretical point of view, the architecture in quantum computer design plays also a crucial role in the very general problem regarding the classical simulation of quantum circuits. As focused by Jozsa and Miyake (2008), the capability of a classical computer to efficiently simulate a quantum circuit is strictly related to the “distance” between the qubits on which a given quantum gate operates, i.e. the number of the Swap operators necessary to simulate the circuit.

On this basis, in Sergioli (2017) a simple mathematical representation of an arbitrary quantum circuit is provided. This representation turns out to be very beneficial for practical

\(^1\) As an example, the linear neighbor architectures LNN (Kumar 2013), offer an appropriate approximate method to approach to physical problems regarding trapped ions (Häffner et al. 2005), liquid nuclear magnetic resonance (Laforest et al. 2007) and the original Kane model (Kane 1998).
usages [for instance related to the implementation of a software package able to efficiently simulate a quantum circuit (Gerdt and Prokopenya 2011; Gerdt et al. 2009)]. But, in addition to the implementative purposes, this representation could be very useful also to provide a generalization of the quantum computational logic that takes into account a more realistic scenario. This paper is devoted to introduce a new version of the quantum computational logic that is strictly dependent on the architecture of the quantum circuit and that, at the same time, represents a generalization of the standard quantum computational logic (Dalla Chiara et al. 2004).

The paper is organized as follows: in Sect. 2 we discuss on the different roles that different qubits can assume along a computation, depending on the architecture of the quantum circuit. In Sect. 3 we briefly describe a formal model to represent an arbitrary quantum circuit by using a synthetic block matrix representation. This representation turns out to be particularly beneficial because it allows to avoid the reiterate involvement of multiple Swap operators. Section 4 is devoted to introduce a very fundamental background of the main features of the standard Quantum Computational Logic (QCL). Sections 5 and 6 constitute the real core of this paper: in Sect. 5 the so called Multi Target Quantum Computational Logic (MT-QCL) is introduced in details, while in Sect. 6 similarities and differences between QCL and MT-QCL are formally showed. The closing section is devoted to introduce some final comments and possible further developments.

## 2 Different Roles of the Qubits in a Quantum Circuit

In the standard computational framework, a quantum circuit can be described by three main ingredients: an input state, a logical gate (or a sequence of logical gates) and an output state. In the quantum computational context, the input state is an \( n \)-dimensional unitary vector \( |x⟩ \) belonging to the complex Hilbert space \( \otimes^n \mathbb{C}^2 \), the logical gates are unitary operators and the output state is again a \( n \)-dimensional unitary vector \( |y⟩ \in \otimes^n \mathbb{C}^2 \). Of course, the input state is arbitrary and the output is univocally determined by the application of the gate to the input state; indeed, the quantum circuit is identified just as a sequence (i.e. a product) of unitary operators (i.e. one unitary operator).

The input of a certain quantum circuit can be given by one qubit only or, more frequently, by many qubits. Let us consider an input state \( |x⟩ = |x_1⟩ \otimes |x_2⟩ \ldots |x_n⟩ = |x_1 \ldots x_n⟩ \).

Depending on the gates that are applied during the computation, not all the qubits play the same role. As an example, a very useful class of gates—used for many algorithmic implementations—is given by the following class of Controlled-\( U \) gates:

\[
CU^{(n)}|x_0 \ldots x_n⟩ = \begin{cases} 
|x_0 \ldots x_n⟩ & \text{if } x_0 = 0; \\
|x_0⟩ \otimes U^{(n)}|x_1 \ldots x_n⟩ & \text{if } x_0 = 1;
\end{cases}
\]

where \( U^{(n)} \) is an arbitrary \( n \)-dimensional unitary operator.

From a logical perspective, a \( CU^{(n)} \) gate can be interpreted as a kind of “double implication” that forces the application of \( U^{(n)} \) on \( |x_1 \ldots x_n⟩ \) if and only if \( x_0 = 1 \). For this reason, we conventionally say that \( |0⟩ \) plays the role of the “control” qubit (that keeps unchanged under the application of the \( CU^{(n)} \) gate), while \( |x_1⟩, |x_1⟩, \ldots, |x_n⟩ \), are called “target” qubits (that, accordingly to the value of \( |0⟩ \), can change under the application of the \( CU^{(n)} \) gate).

It is not hard to show (Sergioli and Freytes 2017) that, for any \( n \)-ary unitary operator \( U^{(n)} \), the respective \( CU^{(n)} \) operator (whose dimension is \( n + 1 \)) assumes the following useful matrix representation:
where $P_0$ and $P_1$ are the projector operators: $P_0 = |0\rangle\langle 0| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ and $P_1 = |1\rangle\langle 1| = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$.

Let us remark how a qubit can be neither a control nor a target qubit. Let us consider Fig. 1.

In this picture the line between the second and the third qubit follows the standard representation of a $CNot$ gate, where the qubit $|x_2\rangle$ plays the role of the control qubit, $|x_3\rangle$ plays the role of the target qubit and the unitary operator $U^{(n)}$ is represented by the standard negation $Not^{(1)}$. Further, the circuit also includes the qubit $|x_1\rangle$ that does play any role in the computation. Formally, the circuit can be represented as the following transformation:

\[
CU^{(n)} = \begin{bmatrix} I^{(n)} & 0 \\ 0 & U^{(n)} \end{bmatrix} = \begin{bmatrix} I^{(n)} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & U^{(n)} \end{bmatrix}
\]

where $P_0$ and $P_1$ are the projector operators: $P_0 = |0\rangle\langle 0| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ and $P_1 = |1\rangle\langle 1| = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$.

3 Generalized Quantum Circuits

In the previous section particular attention has been devoted to the role of the $Controlled-U$ gates in the architecture of a quantum circuit. In this section we briefly provide the mathematical description of arbitrary quantum circuits, where arbitrary gates are involved. This representation turns out to be useful in the rest of the paper.
The simplest case is represented by a circuit where only a single unary gate is applied. However, this information is not enough to mathematically describe the quantum circuit; indeed, a crucial information regards the qubit to which the gate is applied to. Let us consider a unary gate $U^{(1)}$ applied to the $i$th qubit over a $k$-dimensional circuit (i.e., a circuit such that the input and the output live in the Hilbert space $\mathbb{C}^2 \otimes \mathbb{C}^k$). In this case, the circuit is simply represented by the following operator:

$$I^{(i-1)} \otimes U^{(1)} \otimes I^{(k-i)}.$$ 

Similarly, it is possible that $j$ unary operators $U^{(1)}_1, \ldots, U^{(1)}_j$ are applied to different $(i_1)_{st}, \ldots, (i_j)_{th}$ qubits, respectively, over a $k$-dimensional input state. In this case, the circuit is formally represented as:

$$U = I^{(i_1-1)} \otimes U^{(1)}_1 \otimes I^{(i_2-i_1-1)} \otimes U^{(1)}_2 \otimes \cdots \otimes U^{(1)}_j \otimes I^{(k-j)}.$$  \quad (3)

Anyway, a more complex quantum circuit is generally given by sequences of gates in such a way that any qubit can change under the action of several quantum gates. As an example, it is possible to the consider the operator $\hat{U}$ that corresponds to the application of $\tilde{j}$ unary operators $\hat{U}^{(1)}_1, \ldots, \hat{U}^{(1)}_j$ applied to different $(\tilde{i}_1)_{st}, \ldots, (\tilde{i}_j)_{th}$ qubits. Given a $k$-dimensional input state $|\psi_{in}\rangle$, it is possible to apply $\hat{U}$ not to the input, but to the $k$-dimensional state that arises from the application of $U$ to the input state $|\psi\rangle$. In this case the circuit is described by the product $\hat{U}U$ and, trivially, the output is $|\psi_{out}\rangle = \hat{U}U|\psi\rangle$.

We have considered only the special case of a circuit given by unary operators only. By a suitable arrangement of the dimension of the identity operators, it is easy to generalize the argument above to the case of circuits given by compositions of gates of different and arbitrary arity.

As an example, let us consider a 3-dimensional input state $|\psi\rangle = |x_1, x_2, x_3\rangle$ where we first apply a CNot between the first and the second qubit and an Hadamard gate $H$ to the third qubit. After, we apply a CNot between the second and the third qubit. In this case, the circuit is given by the following composition:

$$(CNot \otimes H)(I^{(1)} \otimes CNot).$$

This example is also useful to realize how the role of a qubit of the input state is not designed, in principle, to be a control or a target bit; indeed, the qubit $|x_2\rangle$ plays the role of target during the first “step” of the computation and the role of control during the second “step”.

All the arguments above contain the implicit assumption that a $n$-ary gate is applied to $n$ adjacent qubits. In a more general scenario, a $n$-ary gate $U^{(n)}$ could also be applied to $n$ qubits arbitrary placed over a $k$-dimensional Hilbert space (with $k \geq n$). In this case, the very standard quantum computational procedure requires a full involvement of the Swap operator, by applying the following procedure. Let us consider a $k$-dimensional input state $|x\rangle^m = |x_1 \ldots x_m \ldots x_{m+n} \ldots x_k\rangle$. For the sake of the simplicity, let us consider a binary gate $U^{(2)}$ that is applied to the two non adjacent qubits $|x_m\rangle$ and $|x_{m+n}\rangle$, leaving the other qubits of the input state $|x\rangle^m$ unchanged.

First, we apply $(n-1)$ Swap gates in order to “move” the $|x_m\rangle$ in the $(m+n-1)$th position. Formally:

$|x\rangle^m = Swap_{k:m,m+n-1}[x]\rangle^m,$

where $Swap_{k:m,m+n-1}$ represents the composition of all the $(n-1)$ Swap gates we need to “move” the $|x_m\rangle$ in the $(m+n-1)$th position. At this stage it is possible to apply the binary
operator $U^{(2)}$ to the two qubits $|x_m\rangle$ and $|x_{m+n}\rangle$ (that are now adjacent). Formally (by referring to Eq. 3):

$$|x\rangle^b = (I^{m+n-2} \otimes U^2 \otimes I^{(k-m-n)}) |x\rangle^a.$$

Finally, we need to apply the inverse2 of Swap$_{[k,m,m+n-1]}$ in order to retrieve the initial configuration: $|x\rangle^{out} = $ Swap$_{[k,m,m+n-1]} |x\rangle^b$.

We can formally summarizing this procedure by writing:

$$|x\rangle^{out} = U^{(2)}_{[k,m,m+n]} |x\rangle^{in},$$

where $U^{(2)}_{[k,m,m+n]}$ indicates the $k$-dimensional operator that acts as a binary operator $U^{(2)}$ on the $m$th and the $(m+n)$th qubits of $|x\rangle^{in}$ an leaves all the other qubits of $|x\rangle^{in}$ unchanged; the extended representation of $U^{(2)}_{[k,m,m+n]}$ is now simply obtainable by composition of Swap, I and $U^{(2)}$ gates, suggested by the previous description. Without any lost of generality, all the arguments above can be easily generalized for the case of $n$-ary operators applied to $n$ qubits arbitrarily allocated over a $k$ dimensional input state. Of course, the “distance” between the qubits where the gates are applied, the arity of these gates and the number of these gates are elements that could provide a relevant increasing of the number of the Swap gates necessary to make the required computation. This problem is fully investigated in the general context of architecture in quantum computer design. Indeed, recent topics related to efficent quantum computing among remote qubits in nearest-neighbor architectures are currently under investigation (Kumar 2013; Takahashi et al. 2007). From a purely mathematical viewpoint, it is easy to realize how the using of multiple Swap gates turns out to be particularly uncomfortable. For this reason we provide the following block matrix representation (Sergioli 2017) that are useful in order to formally manage quantum circuits with different gates applied to non adjacent qubits.

**Theorem 1** (Sergioli 2017) Let us consider the projectors operators $P_0$ and $P_1$ and the Ladder operators (Fuchs 1992) $L_0 = |0\rangle\langle 1| = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ and $L_1 = |1\rangle\langle 0| = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$ and let $P^{(n)}_0 = I^{(n-1)} \otimes P_0$ (with $I^{(n-1)} (n-1)$-dimensional identity matrix); similarly for $P^{(n)}_1, L^{(n)}_0$ and $L^{(n)}_1$.

The block-matrix representation of the operator Swap$_{[k,m,m+n]}$ (that swaps the $m$ th qubit with the $(m+n)$th qubit over a $k$-dimensional input state) is given by:

$$\text{Swap}_{[k,m,m+n]} = I^{(m-1)} \otimes \text{Swap}_{[n+1,1,n+1]} \otimes I^{(k-m-n)}$$

where

$$\text{Swap}_{[n,1,n]} = \begin{bmatrix} P^{(n-1)}_0 & P^{(n-1)}_1 \\ P^{(n-1)}_0 & P^{(n-1)}_1 \end{bmatrix}.$$}

This theorem allows us to provide also a confortable block matrix representation of an arbitrary binary gate $U^{(2)}$ (applied to two non adjacent qubits) without incurring in the annoying involvement of multiple Swap gates.

---

2 It is easy to see that Swap$_{[k,m,m+n-1]}^{-1} = \text{Swap}_{[k,m,m+n-1]}$ for any value of $k$, $m$ and $n$. 

Springer
Theorem 2 (Sergioli 2017) Let \( U^{(2)} \) a binary unitary operator given by the following block-matrix representation \( U^{(2)} = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \), where \( U_{ij} \) are 2-dimensional square matrices given by \( U_{11} = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} \), \( U_{12} = \begin{bmatrix} u_{13} & u_{14} \\ u_{23} & u_{24} \end{bmatrix} \), \( U_{21} = \begin{bmatrix} u_{31} & u_{32} \\ u_{41} & u_{42} \end{bmatrix} \) and \( U_{22} = \begin{bmatrix} u_{23} & u_{24} \\ u_{43} & u_{44} \end{bmatrix} \).

The block-matrix representation of \( U^{(2)}_{[k;m,m+n]} \) is given by:

\[
U^{(2)}_{[k;m,m+n]} = I^{(m-1)} \otimes \left[ \begin{bmatrix} U^{(n)}_{11} & U^{(n)}_{12} \\ U^{(n)}_{21} & U^{(n)}_{22} \end{bmatrix} \right] \otimes I^{(k-m-n)},
\]

where \( U^{(n)}_{ij} = I^{(n-1)} \otimes U_{ij} \).

It is not hard to convince that, by following very similar arguments, it is possible to obtain a similar result for an arbitrary \( n \)-ary gate applied to \( n \) non adjacent qubits (Sergioli 2017).

4 The Standard Approach to the Quantum Computational Logic

The theory of Quantum Computation has naturally inspired new forms of quantum logic, the so called Quantum Computational Logic (QCL) (Dalla Chiara et al. 2004; Sergioli et al. 2011). From a semantic point of view, any formula of the language in the QCL denotes a piece of quantum information, i.e. a density operator living in a complex Hilbert space whose dimension depends on the linguistic complexity of the formula. Similarly, the logical connectives are interpreted as special examples of quantum gates. Accordingly, any formula of a quantum computational language can be regarded as a logical description of a quantum circuit. The initial concept at the very background of the QCL is the assignment of the truth value of a quantum state that represents a formula of the language. Conventionally, the QCL assumes to assign the truth value “false” to the information stored by the qubit \( |0\rangle \) and the truth value “true” to the qubit \( |1\rangle \). Unlike the classical logic, QCL turns out to be a probabilistic logic, where the qubit \(|\psi\rangle = c_0|0\rangle + c_1|1\rangle \) logically represents a “probabilistic superposition” of the two classical truth values, where the falsity has probability \(|c_0|^2\) and the truth has the probability \(|c_1|^2\). As in the qubit case, in the standard approach to QCL a probability function \( p \) assigns a probability value \( p(\rho) \) to any density operator \( \rho \) living in the space of the \( n \)-ary dimensional density operators (we denote this space by \( D(\otimes^n\mathbb{C}^2) \)). Intuitively, \( p(\rho) \) is the probability that the quantum information stored by \( \rho \) corresponds to a true information.

In order to define the function \( p \), we first need to identify in any space \( \otimes^n\mathbb{C}^2 \) the two operators \( P_0^{(n)} \) and \( P_1^{(n)} \) as the two special projectors that represent the falsity and the truth properties, respectively. Before this, a step is very crucial. In order to extend the definition of true and false from the space \( \mathbb{C}^2 \) of the qubits to the space \( \otimes^n\mathbb{C}^2 \) of the tensor product on \( n \) qubits (i.e. on an arbitrary quregister), the standard approach of the QCL accords with the following convention: a quregister \(|x\rangle = |x_1 \ldots x_n\rangle \) is said to be false if and only if \( x_n = 0 \); conversely, it is said to be true if and only if \( x_n = 1 \). Hence, the truth value of a quregister only depends on its last component. On this basis, it is natural to define the
property falsity (or truth) on the space $\otimes^n \mathbb{C}^2$ as the projector $P_0^{(n)}$ (or $P_1^{(n)}$) onto the span of the set of all false (or true) registers. Now, accordingly with the Born rule, the probability that the state $\rho$ is true is defined as:

$$p(\rho) = \text{Tr}\left( P_1^{(n)} \rho \right).$$

(4)

In QCL the evolution of a quregister is dictated by the application of a unitary operator (that represents a reversible transformation) while the evolution of a density operator is dictated by the application of a quantum operation (that represents a transformation that, in general, is not reversible). Of course, for any quantum gate $U$ there exists the correspondent quantum operation $^D U$ that replaces the behavior of the quantum gate in the context of the density operators (in particular, $^D U(\rho) = U\rho U^\dagger$), but the other way generally does not hold.

In the language of the QCL it is usual to distinguish between semiclassical gates (called semiclassical because, when they are applied to the elements of the computational basis $\mathbb{B} = \{|0\rangle, |1\rangle \}$, they replace the behavior of their corresponding classical logical gates) and genuinely quantum gates (called genuinely quantum because their application to the elements of the computational basis has not any classical counterpart). The semiclassical gates usually involved in the QCL are: the Identity $I$, the Negation $\text{Not}$, the control-negation (or Xor) $C\text{Not}$ and the Toffoli gate $T$, while the genuinely quantum gates are: the Hadamard gate $\sqrt{I}$ (also named square root of the identity) and the square root of the negation $\sqrt{\text{Not}}$.

In particular, the $T$ and the $\text{Not}$ gates allows us to provide a probabilistic replacement of the classical logic in virtue of the following properties:

- $p(^D \text{Not}(\rho)) = 1 - p(\rho)$, for any $\rho \in \otimes^n \mathbb{C}^2$;
- $p(\text{AND}(\rho, \sigma)) = p(^D T(\rho \otimes \sigma \otimes P_0)) = p(\rho)p(\sigma)$, for any $\rho, \sigma \in \otimes^n \mathbb{C}^2$.

Let us notice how the conjunction is obtained by the expedient to use the ternary Toffoli gate equipped by the projector $P_0$ that plays the role of an ancilla.

Basing on this approach and inspired by the intrinsic properties of the quantum systems, the semantic of the QCL turns out to be strongly non-compositional and context dependent (Dalla Chiara et al. 2016). This approach, that may appear prima facie strange, leads to the benefit to reflect many of informal arguments that are currently used in our rational activity (Dalla Chiara et al. 2015). A detailed description of the QCL and its algebraic properties are summarized in (Dalla Chiara et al. 2004, 2013, 2016; Ledda and Sergioli 2010).

## 5 A New Definition of Probability in QCL

Following the brief description of a quantum circuit given in Sect. 3, we can easily realize that the actual carrier of information is given by the target bit, while the control bit remains unchanged under the application of a given quantum gate; furthermore, the set of the qubits that play the role of target can change during the computation, depending on the gates that, step by step, are applied. On the other hand, despite its remarkable expressive power, the very preliminary notions of the QCL introduced in Sect. 4 seem to do not take into suitable account this fact, assuming that the “useful” information is always stored by the last qubit only. Indeed, the assignment of the truth value of a given composition of qubits (a quregister), accordingly with what we have previously introduced, only depends on the last
component. For this reason, all the gates that are involved in the language of the QCL are only one-target gates. This restriction is basically unnecessary and it could also seem to be at distance from the architecture of a real quantum circuit. For this reason, this section is devoted to introduce an extension of the QCL (that we will call Multi Target QCL, briefly MT-QCL) and to show some preliminary results. The immediate benefit of the MT-QCL with respect to the QCL is given by the fact that the MT-QCL can also involve in the language multi target gates (such as the Swap gate, the square root of Swap gate \( \sqrt{\text{Swap}} \) and the Fredkin gate \( F \)) without any lost of generality. Furthermore, in this framework the standard QCL can be seen as a particular (one-target) case of the MT-QCL.

Similarly to the case of QCL, in order to introduce the MT-QCL the essential step is given by the definition of probability. First, let us consider a very simple circuit given by the definition of probability. First, let us consider a very simple circuit given by the standard QCL can be seen as a particular (one-target) case of the MT-QCL.

Let us give a slight abuse of the terms \( \text{target} \) and \( \text{control} \) remarking that, for example, if we refer to the picture of the Sect. 3, the qubit \( |x_i\rangle \) is not a control qubit but, accordingly with this notation, it assumes a \( \text{control position} \) over the quantum circuit where it is located.

**Proof** Trivially follows by Definition 1. 

---

3 Let us give a slight abuse of the terms \( \text{target} \) and \( \text{control} \) remarking that, for example, if we refer to the picture of the Sect. 3, the qubit \( |x_i\rangle \) is not a control qubit but, accordingly with this notation, it assumes a \( \text{control position} \) over the quantum circuit where it is located.
As discussed in Sect. 3, when more quantum gates are involved along a computation, it is possible that the same qubit plays, as an example, the role of target during one “step” of the computation and the role of control during another “step”. Reasonably, we accord to fix that if a qubit $|x_i\rangle$ assumes a target position at list one time during the computation, then the position $i$ will be considered as a target position in the evaluation of the probability given by Definition 1.

On this basis, it is possible to naturally generalize the Definition 1 to the case where $U^{(n)}$ is not only one gate quantum circuit, but it is an arbitrary circuit given by an arbitrary composition of quantum gates.

**Example 1** Let us consider a quantum circuit given by the following composition of gates

$$U^{(3)} = (\text{CNot} \otimes H)(I^{(1)} \otimes \text{CNot}).$$

and let $\rho^{(n)} = \rho \otimes \sigma \otimes \tau$, where $\rho = \frac{1}{2} \begin{bmatrix} 1 + r_3 & r_1 - ir_2 \\ r_1 + ir_2 & 1 - r_3 \end{bmatrix}$, $\sigma = \frac{1}{2} \begin{bmatrix} 1 + s_3 & s_1 - is_2 \\ s_1 + is_2 & 1 - s_3 \end{bmatrix}$ and $\tau = \frac{1}{2} \begin{bmatrix} 1 + t_3 & t_1 - it_2 \\ t_1 + it_2 & 1 - t_3 \end{bmatrix}$.

We observe that only the first qubit $\rho$ never assumes a target position. Hence, by Definition 1, we have that:

$$P[U^{(3)}, \rho^{(n)}] = \text{Tr}[(I^{(1)} \otimes P_1 \otimes P_1)^D U^{(3)} \rho^{(n)}] \quad = \frac{1}{4}[(1 - 2p_\rho)(1 - 2p_\sigma) - 1](t_1 - 1).$$

Let us immediately notice that the main difference between the definition of probability given in the standard QCL and the one given in the MT-QCL is that the first is related to the quantum state, while the second is related to the quantum circuit. Indeed, given a certain output state $|x\rangle^{out}$, the QCL assigns a probability to $|x\rangle^{out}$ independently to the computation from which the state comes from; conversely, the probability assigned by the MT-QCL is strictly dependent on the history of $|x\rangle^{out}$. This turns out to be very noticeable by the next example.

**Example 2** Let us consider the input state $\rho \in D(\otimes^2 \mathbb{C}^2)$ and let also consider the two circuits dictated by the following compositions of quantum gates: $U_1 = (\sqrt{I} \otimes I)(\sqrt{I} \otimes I)$ and $U_2 = I^{(2)}$. Obviously, $U_1 = U_2$ and $D U_1(\rho) = D U_2(\rho) = \rho$. But,

$$P[D U_1, \rho] = \text{Tr}(P_1 \otimes I)\rho \neq \text{Tr}(I \otimes I)\rho = P[D U_2, \rho].$$

Indeed, even if $U_1 = U_2$, the first position of $U_1$ is a target position, while the first position of $U_2$ is a control position. This difference is captured by the MT-QCL in virtue of the new definition of probability.

---

4 Notice that the probability $p$ we refer to is the one introduced in the context of the standard QCL (see Eq. 4.1). On this basis, we are using this example to express the MT-QCL probability $P$ in terms of the QCL probability $p$. 

Springer
The fact that two (let’s say) “equivalent” but not “identical” circuits (such as $U_1$ and $U_2$ of the previous example) provide different probabilistic results in the framework of MT-QCL is not so surprising. Indeed, even in the real quantum computation, two equivalent (but not identical) circuits applied to the same input state can provide as output a different distribution. In Fig. 2 we show the result of the performing of two equivalent (but not identical) identity circuits applied to the same input state $|\psi\rangle = |00000\rangle$. The results are obtained by running the output on a large number of computations on the real IBM quantum computer (Devitt 2016). As we can see, even if the expected result for both cases is, obviously, $|\psi\rangle$, the probabilities induced by the frequency distributions in the two cases are different. This difference arises from the fact that, even if from an operational point of view the two circuits are exactly equivalent, the computations that are executed by the quantum computer are essentially different.

6 Some New Results

Now let us generalize the notation that we have introduced in Sect. 3, and let us indicate by $U(\alpha)_{[k,m,m+n_1,\ldots,m+n_{n-1}]}$ an $\alpha$-arity gate that is applied to the $m$th, $(m+n_1)$th, $\ldots$, $(m+n_{n-1})$th qubits, respectively, and let calculate the value that assumes the new definition of probability associated to some particular gate.

**Theorem 4** Let $\rho_1, \rho_2, \ldots, \rho_k$ be density operator leaving in the two dimensional Hilbert space $D(C^2)$ such that $\rho_i = \frac{1}{2} \left[ \begin{array}{cc} 1 + r_{i_3} & r_{i_1} - ir_{i_2} \\ r_{i_1} + ir_{i_2} & 1 - r_{i_3} \end{array} \right]$. We have that:

1. $P^D[\text{Not}^{(1)}_{[k,i]}](\rho_1 \otimes \cdots \otimes \rho_i \otimes \rho_k) = \frac{1+r_{i_3}}{2}$;
2. $P^D[\sqrt{I}^{(1)}_{[k,i]}](\rho_1 \otimes \cdots \otimes \rho_i \otimes \rho_k) = \frac{1-r_{i_3}}{2}$;
3. $P^D[\sqrt{\text{Not}}^{(1)}_{[k,i]}](\rho_1 \otimes \cdots \otimes \rho_i \otimes \rho_k) = \frac{1-r_{i_3}}{2}$;
4. $P^D[\text{CNot}^{(1)}_{[k,m,m+n]}](\rho_1 \otimes \cdots \otimes \rho_k) = \frac{1-r_{m+n(m+n+1)}}{2}$;
5. $P^D[T^{(1)}_{[k,m,m+n+m+p]}](\rho_1 \otimes \cdots \otimes \rho_k) = \frac{1}{4} (2 + (r_{m_3}(r_{(m+n)3} - 1) - r_{(m+n)3} - 1) \cdot r_{(m+n+p)3})$. 

Fig. 2 Two equivalent circuits with different frequency distributions
Proof

1. \[ P[DNot_{[k,l]}(\rho_1 \otimes \cdots \otimes \rho_i \otimes \cdots \otimes \rho_k)] = \text{Tr}[(I^{(i-1)} \otimes P_1 \otimes I^{(k-i)}) \cdot D(I^{(i-1)} \otimes \text{Not} \otimes I^{(k-i)}) \cdot (\rho_1 \otimes \cdots \otimes \rho_i \otimes \cdots \otimes \rho_k)] = \text{Tr}[(I^{(i-1)} \cdot D(I^{(i-1)}(\rho_1 \otimes \cdots \otimes \rho_{i-1})) \otimes (P_1 \cdot DNot \rho_i) \otimes (I^{(k-i)} \cdot D(I^{(k-i)}(\rho_{i+1} \otimes \cdots \otimes \rho_k)))] = \frac{1 + r_{i3}}{2}; \]

2. follows in a similar way;
3. follows in a similar way;
4. \[ P[D\text{CNot}_{[k,m,m+n]}(\rho_1 \otimes \cdots \otimes \rho_m \otimes \cdots \otimes \rho_{m+m} \otimes \cdots \otimes \rho_k)] = \text{Tr}[(I^{(m+n-1)} \otimes P_1 \otimes I^{(k-n)}) \cdot D\text{Swap}_{[k,m,m+n-1]}(D(I^{(m+n-2)} \otimes \text{CNot} \otimes I^{(k-n-1)})) \cdot (D\text{Swap}_{[k,m,m+n-1]}(\rho_1 \otimes \cdots \otimes \rho_i))] = (\text{by Theorem } 2) = \text{Tr}(P_1(C\text{Not}(\rho_m \otimes \rho_{m+n}))) = \frac{1 - r_{m3}s_{(m+n)3}}{2}; \]

5. follows in a similar way. □

Let us notice that the previous results are in accord with the standard QCL (Dalla Chiara et al. 2013) in the cases (1), (2) and (3). Also for the case (4), after considering to “move” the qubit \( \rho_k \) in the position \( m + n - 1 \) by a suitable availment of Swap (in accord with Theorem 1) in order to have \( \rho_m \) and \( \rho_{m+n} \) in two adjacent positions, we are perfectly recovering the probability values obtained by the standard QCL. It is easy to think in a very similar way regarding the item (5).

As we have discussed in Sect. 3, the action of the Swap gate allows to generalize the action of \( n \)-ary gates to the case where the qubits to which the gates are applied are not adjacent one another. Anyway, it has not any influence in the calculation of the probability; as an example: \( P[D\text{CNot}_{[k,m,m+n]}(\rho_1 \otimes \cdots \otimes \rho_m \otimes \cdots \otimes \rho_{m+n} \otimes \cdots \otimes \rho_k)] \) it is simply equal to \( P[D\text{CNot}_{[k,m,m+n]}(\rho_m \otimes \rho_{m+n})] \); hence, without lost of generality, we can confine ourselves in calculating the probability in the special case in which the qubits where the gate acts on are adjacent.

**Theorem 5** Let \( \rho_1, \ldots, \rho_k \) as defined in Theorem 4.

1. \[ P[D\text{Swap}_{[1,2]}(\rho_1 \otimes \rho_2)] = P(\rho_1)P(\rho_2); \]
2. \[ P[D\sqrt{\text{Swap}}_{[1,2]}(\rho_1 \otimes \rho_2)] = P(\rho_1)P(\rho_2); \]
3. \[ P[D\text{F}_{[1,2,3]}(\rho_1 \otimes \rho_2 \otimes \rho_3)] = P(\rho_2)P(\rho_3); \]

**Proof** Trivially follow from Definition 1 and by straightforward calculations. □

Let us notice that, by referring to the definition of probability of the standard QCL, the previous theorem assumes a remarkable different form.
By the way, the most remarkable results should regard the case where the input state is not a product state. On this basis, many results have been obtained regarding one-target gates (Dalla Chiara et al. 2016; Sergioli and Freytes 2017). Following similar arguments and according with the new definition of probability, we show similar results also for the Swap, \(\sqrt{\text{Swap}}\) and \(F\) gates.

**Theorem 6** Let \(U\) a binary operator \(U = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}\) (as represented in Theorem 2) and let us consider that \(U\) is not a control-target operator. Let \(\rho \in \mathcal{D}(\mathbb{C}^{k+2})\). Then:

\[
P[D_{U[k,m,n+1]}\rho] = \text{Tr}\left(\left\{ A_U^{(m+n)} \otimes I^{(k-m-n)}\right\}\rho \right),
\]

where \(A_U^{(n+1)} = \left( U^{(n-1)} \otimes \left( U_{21}^\dagger P_1 U_{21} \right) \right) \left( U^{(n-1)} \otimes \left( U_{22}^\dagger P_1 U_{22} \right) \right) \left( U^{(n-1)} \otimes \left( U_{22}^\dagger P_1 U_{22} \right) \right)\).

**Proof** By Definition 1, we have:

\[
P[U_{k,m,n+1}, \rho] = \text{Tr}\left( P_1^{(m)} \otimes P_1^{(n)} \otimes I^{(k-m-n)} U_{k,m,n+1} \rho U_{k,m,n+1}^\dagger \right)
\]

\[
= \text{Tr}\left( U_{k,m,n+1}^\dagger (P_1^{(m)} \otimes P_1^{(n)} \otimes I^{(k-m-n)}) U_{k,m,n+1} \rho \right)
\]

\[
= (\text{by Theorem 2}) \text{Tr}\left( U^{(m-1)} \otimes \left[ \begin{bmatrix} U^{(n)}_{11} & U^{(n)}_{12} \\ U^{(n)}_{21} & U^{(n)}_{22} \end{bmatrix} \right] \otimes I^{(k-m-n)} \right)
\]

\[
\cdot (f^{(m-1)} \otimes (P_1 \otimes P_1^{(n)} \otimes I^{(k-m-n)}) \rho)
\]

\[
= \text{Tr}\left( (f^{(m-1)} \otimes \left[ \begin{bmatrix} U^{(n)}_{11} & U^{(n)}_{12} \\ U^{(n)}_{21} & U^{(n)}_{22} \end{bmatrix} \right] \otimes I^{(k-m-n)} \right) \rho \right),
\]

where \(0^{(n)}\) is the \(n\)-dimensional null matrix. Let us notice that

\[
\left[ \begin{bmatrix} U^{(n)}_{11} & U^{(n)}_{12} \\ U^{(n)}_{21} & U^{(n)}_{22} \end{bmatrix} \right] \cdot \left[ \begin{bmatrix} 0^{(n)} \\ 0^{(n)} \end{bmatrix} \right] = \left[ \begin{bmatrix} 0^{(n)} \\ p^{(n)} \end{bmatrix} \right]
\]

\[
= \left[ \begin{bmatrix} 0^{(n)} \\ 0^{(n)} \end{bmatrix} \right] \cdot \left[ \begin{bmatrix} 0^{(n)} \\ p^{(n)} \end{bmatrix} \right] = \left[ \begin{bmatrix} U^{(n)}_{11} & U^{(n)}_{12} \\ U^{(n)}_{21} & U^{(n)}_{22} \end{bmatrix} \right] \cdot \left[ \begin{bmatrix} 0^{(n)} \\ 0^{(n)} \end{bmatrix} \right] = \left[ \begin{bmatrix} U^{(n)}_{11} & U^{(n)}_{12} \\ U^{(n)}_{21} & U^{(n)}_{22} \end{bmatrix} \right] \cdot \left[ \begin{bmatrix} 0^{(n)} \\ 0^{(n)} \end{bmatrix} \right] = \lambda_U^{(n+1)}
\]

Hence,

\[
P[D_{U[k,m,n+1]}, \rho] = \text{Tr}\left( (f^{(m-1)} \otimes \left[ \begin{bmatrix} U^{(n)}_{11} & U^{(n)}_{12} \\ U^{(n)}_{21} & U^{(n)}_{22} \end{bmatrix} \right] \otimes I^{(k-m-n)} \right) \rho \right) = \text{Tr}\left( \left[ \begin{bmatrix} U^{(n+1)} & 0^{(n+1)} \end{bmatrix} \right] \otimes \left[ \begin{bmatrix} 0^{(n)} & 0^{(n)} \\ 0^{(n)} & p^{(n)} \end{bmatrix} \right] \right) = \lambda_U^{(n+1)}
\]
In this theorem we have considered the case where the binary operator $U$ is not a control-target operator. Accordingly with Definition 1, this assumption is essential in order to establish the correct expression of the probability of the circuit. In the case where the binary operator $U$ is a control-target gate, this expression will change, as showed in the following theorem.

**Theorem 7** Let $C\hat{U}$ be a binary control-target operator $C\hat{U} = \begin{bmatrix} I & 0 \\ 0 & U \end{bmatrix}$ (as represented in (1)), with $\hat{U}$ arbitrary unary gate. Let $\rho \in D(\otimes^k \mathbb{C}^2)$. Then:

$$P\left[P_{C\hat{U}}, \rho \right] = Tr\left[ (A^{(n+1)}_{C\hat{U}} \otimes I^{(k-m-n)}) \rho \right],$$

where $A^{(n+1)}_{C\hat{U}} = \begin{bmatrix} P^{(n)}_1 & 0^{(n)} \\ 0^{(n)} & I^{(n-1)} \otimes (U^\dagger P_1 U) \end{bmatrix}$.

**Proof** By Definition 1, we have:

$$P\left[P_{C\hat{U}}, \rho \right] = Tr\left[ (I^{(m)} \otimes P^{(n)}_1 \otimes I^{(k-m-n)}) C\hat{U}_{[k,m,m+n]} \rho (C\hat{U}_{[k,m,m+n]})^\dagger \right]$$

$$= Tr\left[ ((C\hat{U}_{[k,m,m+n]})^\dagger (I^{(m)} \otimes P^{(n)}_1 \otimes I^{(k-m-n)}) C\hat{U}_{[k,m,m+n]} \rho \right]$$

$$= (by \ Theorem \ 2) = Tr\left[ (I^{(m-1)} \otimes \left[ \begin{array}{c} I^{(n)} \\ 0^{(n)} \end{array} \right] I^{(n-1)} \otimes \frac{U^\dagger}{U} \right. \otimes I^{(k-m-n)}) \right]$$

$$\cdot \left( \begin{array}{c} I^{(m)} \otimes P^{(n)}_1 \otimes I^{(k-m-n)} \\ I^{(m)} \otimes I^{(n-1)} \otimes \frac{U^\dagger}{U} \end{array} \right) \cdot \left[ \begin{array}{c} I^{(n)} \\ 0^{(n)} \end{array} \right] \rho \right]$$

$$= Tr\left[ \left( I^{(m-1)} \otimes \left[ \begin{array}{c} I^{(n)} \\ 0^{(n)} \end{array} \right] I^{(n-1)} \otimes \frac{U^\dagger}{U} \right) \cdot \left[ \begin{array}{c} I^{(n)} \\ 0^{(n)} \end{array} \right] \rho \right]$$

$$\cdot \left( \begin{array}{c} I^{(m)} \otimes P^{(n)}_1 \otimes I^{(k-m-n)} \\ I^{(m)} \otimes I^{(n-1)} \otimes \frac{U^\dagger}{U} \end{array} \right) \cdot \left[ \begin{array}{c} I^{(n)} \\ 0^{(n)} \end{array} \right] \rho \right].$$

Let us notice that (by the mixed-product property of the tensor product):

$$\begin{bmatrix} I^{(n)} \\ 0^{(n)} \end{bmatrix} I^{(n-1)} \otimes \frac{U^\dagger}{U} \begin{bmatrix} I^{(n)} \\ 0^{(n)} \end{bmatrix} = \begin{bmatrix} I^{(n)} \\ 0^{(n)} \end{bmatrix} \begin{bmatrix} 0^{(n)} \\ I^{(n-1)} \otimes \frac{U^\dagger}{U} \end{bmatrix} = \begin{bmatrix} I^{(n)} \\ 0^{(n)} \end{bmatrix} \begin{bmatrix} 0^{(n)} \\ I^{(n-1)} \otimes \frac{U^\dagger}{U} \end{bmatrix} = A^{(n+1)}_{C\hat{U}}$$

Hence, our claim. □
Corollary 1

\[ P(\sqrt{\text{Swap}}_{[k,m,n+l]}, \rho) = Tr\left[ \left( P_{1}^{(m)} \otimes P_{1}^{(n)} \otimes I_{k-m-n} \right) \rho \right]. \]

**Proof** By a straightforward calculation, it can be seen that, for any arbitrary \( \rho \in D(\otimes^{k} \mathbb{C}^{2}) \), is:

\[ A_{\text{Swap}}^{(n+1)} = \left[ \begin{pmatrix} \mathbb{I} & 0^{(n)} \\ 0^{(n)} & P_{1}^{(n)} \end{pmatrix} \right] = P_{1} \otimes P_{1}^{(n)}. \]

Hence, \( A_{\text{Swap}}^{(m+n)} = P_{1}^{(m)} \otimes A_{\text{Swap}}^{(n+1)} = P_{1}^{(m)} \otimes P_{1}^{(n)}. \)

By a direct application of Theorem 6, easily follows our claim.

Following a similar reasoning, it is easy to verify that

\[ A_{\text{Swap}}^{(n+1)} = A_{\text{Swap}}^{(n+1)}, \]

hence \( P(\sqrt{\text{Swap}}_{[k,m,n+l]}, \rho) = P(\sqrt{\text{Swap}}_{[k,m,n+l]}, \rho) \) for any \( \rho \in \otimes^{k} D(\mathbb{C}^{2}). \)

It is remarkable to notice that in the evaluation of these probability values, the gates \( \text{Swap} \) and \( \sqrt{\text{Swap}}_{[k,m,n+l]} \) plays the same role of the binary identity gate.

It is also easy to verify that the probability value of the \( \text{CNot}_{[k,m,n+l]} \) gate in the context of the MT-QCL is in accord with the standard QCL.

As a final example concerning a ternary gate, we also calculate the probability of the generalized Fredkin gate.

**Theorem 8** Let \( \rho \in D(\otimes^{k} \mathbb{C}^{2}) \) and let \( F_{[k,m,n,m+n+l]} \) the \( k \)-dimensional Fredkin gate. We have that

\[ P(F_{[k,m,n,m+n+l]}, \rho) = Tr\left[ \left( P_{1}^{(m)} \otimes A_{F}^{(n+1)} \otimes I_{k-m-n-l} \right) \rho \right], \]

where \( A_{F}^{(n+1)} = P_{0} \otimes P_{1}^{(n)} \otimes P_{1}^{(l)} + P_{1} \otimes P_{1}^{(l)} \otimes P_{1}^{(n)} \).

**Proof** The proof follows by considering the Fredkin gate \( F_{[k,m,n,m+n+l]} \) (let’s simply say \( F \)) as a Control-Swap gate, where the \( m \)th qubit plays the role of control and the \( (m+n) \)th and the \( (m+n+l) \)th qubits play the role of target. For this reason, in accord with Definition 1, the probability will be given by:

\[ P(F, \rho) = Tr\left[ \left( P_{1}^{(m)} \otimes P_{1}^{(n)} \right) \otimes P_{1}^{(l)} \otimes I_{k-m-n-l}F \rho F \right]. \]

Further, by considering the representation of the Fredkin gate as (Sergioli and Freytes 2017):

\[ F_{[n,n,l]} = P_{0} \otimes F_{[n+l]} + P_{1} \otimes \text{Swap}_{[n+l, n+l]} \]

and by a direct calculation, we obtain that \( P(F_{[k,m,n,m+n+l]}, \rho) = Tr\left[ (P_{1}^{(m)} \otimes A_{F}^{(n+1)} \otimes I_{k-m-n-l}) \rho \right], \) where
\[ A_{F}^{(n+l+1)} = [P_0 \otimes f^{(n+l)} + P_1 \otimes \text{Swap}_{[n+l,n,n+l]}] \cdot \left[ I \otimes \left( P_1^{(n)} \otimes P_1^{(l)} \right) \right] \]
\[ \cdot [P_0 \otimes f^{(n+l)} + P_1 \otimes \text{Swap}_{[n+l,n,n+l]}] \]
\[ = [P_0 \otimes P_1^{(n)} \otimes P_1^{(l)} + P_1 \otimes \left( \text{Swap}_{[n+l,n,n+l]} \cdot \left( P_1^{(n)} \otimes P_1^{(l)} \right) \right) ] \]
\[ \cdot [P_0 \otimes f^{(n+l)} + P_1 \otimes \text{Swap}_{[n+l,n,n+l]}] \]
\[ = (P_0 \cdot P_0) \otimes \left( \left( P_1^{(n)} \otimes P_1^{(l)} \right) \cdot f^{(n+l)} \right) \]
\[ + (P_0 \cdot P_1) \otimes \left( \left( P_1^{(n)} \otimes P_1^{(l)} \right) \cdot \text{Swap}_{[n+l,n,n+l]} \right) \]
\[ + (P_1 \cdot P_0) \otimes \left( \text{Swap}_{[n+l,n,n+l]} \cdot \left( P_1^{(n)} \otimes P_1^{(l)} \right) \cdot f^{(n+l)} \right) \]
\[ + (P_1 \cdot P_1) \otimes \left( \text{Swap}_{[n+l,n,n+l]} \cdot \left( P_1^{(n)} \otimes P_1^{(l)} \right) \cdot \text{Swap}_{[n+l,n,n+l]} \right). \]

Hence our claim. \( \square \)

The results provided in this section permit to realize that the QCL can be seen as a particular (one-target) generalization of the MT-QCL. Indeed, if we confine to the set of the one-target qubits, the MT-QCL replace the same probabilistic results of the standard QCL; on the other hand, the MT-QCL allows to consider also multi-target gates that are not possible to take into account in the context of the standard QCL.

7 Conclusion and Further Developments

In this paper we have introduced the formal framework of a generalization of the standard Quantum Computational Logic, that takes into account the different roles of the qubits in a quantum circuit. The most relevant utility of the Quantum Computational Logic is based on the fact that it turns out to be strongly holistic (non-compositional). As a natural consequence, it turns out to be extremely useful to represent all these kinds of situations where the meaning of a compound system (logically represented by a sentence) is not simply dependent on the meaning of its subsystems (logically represented by the atomic sentences), but it has to be considered as a whole. Further, the standard Quantum Computational Logic has also the extreme privilege to be context dependent, i.e. the same sentence can assume different meanings in different contexts. Non-compositionality and context-dependences are two features that make the Quantum Computational Logic extremely useful to describe situations related to very different aspects of the real rational activity in many different contexts such as common language, human psychology, machine learning and even the usual way to perceive the music (Beltrametti et al. 2012; Dalla Chiara et al. 2015, 2016; Holik et al. 2018; Sergioli et al. 2017).

Let us notice how the QCL can reach a so strong expressive power by considering in the language only a particular class of gates (one-target gates). The Multi Target Quantum Computational Logic hereby introduced, allows to consider infinite many more gates (i.e. logical connectives) inspired by the standard language of Quantum Computational Logic and, at the same time, it is more realistic in accord with what happens in a real quantum computational process. On this basis, this expansion of the language can be considered as a promising tool to boost the expressive power of the QCL. As a natural pursuance, the investigation on the theoretical and semantic benefits of this more expressive representation seems to be an interesting argument for further developments. In particular, future investigations will be devoted—from
a theoretical point of view—to a complete study on the algebraic properties of this new logical structures and—from a more applicative viewpoint—to continue to investigate on the benefit obtained by applying these logical structures inspired by quantum theory to non-standard contexts such as human behavior, machine learning and so on.

Acknowledgements This work is partially supported by Regione Autonoma della Sardegna within the project “Time-logical evolution of correlated microscopic systems” CRP 55, L.R. 7/2007 (2015) and by Fondazione Sardegna within the project “Strategies and Technologies for Scientific Education and Dissemination” (2017).

References

Beltrametti, E., Dalla Chiara, M. L., Giuntini, R., Leporini, R., & Sergioli, G. (2012). Epistemic quantum computational structures in a Hilbert-space environment. Fundamenta Informaticae, 115, 1–14.
Dalla Chiara, M. L., Giuntini, R., & Greechie, R. (2004). Reasoning in quantum theory: Sharp and unsharp quantum logic. Trends in logic. Berlin: Springer.
Dalla Chiara, M. L., Giuntini, R., Leporini, R., Negri, E., & Sergioli, G. (2015). Quantum information, cognition and music. Frontiers in Psychology, 6, 1583.
Dalla Chiara, M. L., Giuntini, R., Leporini, R., & Sergioli, G. (2016). Holistic logical arguments in quantum computation. Mathematica Slovaca, 66(2), 313–334.
Dalla Chiara, M. L., Giuntini, R., & Sergioli, G. (2013). Probability in quantum computation and in quantum computational logic. Mathematical structures in computer science (Vol. 14). Cambridge: Cambridge University Press.
Devitt, S. J. (2016). Performing quantum computing experiments in the cloud. Physical Review A. https://doi.org/10.1103/PhysRevA.94.032329.
Fowler, A. G., Hill, C. D., & Hollenberg, L. C. L. (2004). Quantum error correction on linear nearest neighbor qubit arrays. Physical Review A, 69, 042314.1–042314.4.
Fuchs, J. (1992). Affine Lie algebras and quantum groups. Cambridge: Cambridge University Press.
Gerdt, V. P., Kragler, R., & Prokopenya, A. N. (2009). A Mathematica program for constructing quantum circuits and computing their unitary matrices. Physics of Particles and Nuclei Letters, 6, 526.
Gerdt, V. P., & Prokopenya, A. N. (2011). The circuit model of quantum computation and its simulation with Mathematica. In Mathematical modelling and computer science, LNCS (pp. 43–55). Springer.
Häffner, H., Hänsel, W., Roos, C. F., Benhelm, J., Chek al kar, D., Chwalla, M., et al. (2005). Scalable multiparticle entanglement of trapped ions. Nature, 438, 643–646.
Hirvensalo, M. (2001). Quantum computing. Natural computing series. Berlin: Springer.
Holik, F, Sergioli, G., Freytes, H., & Plastino, A. (2018). Pattern recognition in non-Kolmogorovian structures. Foundations of Science, 23(1), 119–132.
Jozsa, R., & Miyake, A. (2008). Matchgates and classical simulation of quantum circuits. Proceedings of the Royal Society A, 464, 3089–3106.
Kane, B. (1998). A silicon-based nuclear spin quantum computer. Nature, 393, 133–137.
Kitaev, A. Y., Shen, A., & Vyalyi, M. N. (2002). Classical and quantum computation. Graduate studies in mathematics (Vol. 47). Rhode Island: AMS.
Kumar, P. (2013). Efficient quantum computing between remote qubits in linear nearest neighbor architectures. Quantum Information Processing, 12–4, 1737–1757.
Laforest, M., Simon, D., Boileau, J. C., Baugh, J., Dity, M., & Laflamme, R. (2007). Using error correction to determine the noise model. Physical Review A, 75, 133–137.
Ledda, A., & Sergioli, G. (2010). Towards quantum computational logics. International Journal of Theoretical Physics, 49, 3158–3165.
Linke, N. M., Maslov, D., Roetteler, M., Debnath, S., Figgatt, C., Landsman, K. A., et al. (2017). Experimental comparison of two quantum computing architectures. Proceedings of the National Academy of Sciences of the United States of America, 114, 3305–3310.
Nielsen, M. A., & Chuang, I. (2000). Quantum computation and quantum information. Cambridge: Cambridge University Press.
Sergioli, G. (2017). Quantum circuit optimization for unitary operators over non-adjacent qudits. arXiv :1711.09765 [quant-ph]
Sergioli, G., & Freytes, H. (2017). Fuzzy approach to quantum Fredkin gate. Journal of Logic & Computation, 28(1), 245–263.
Sergioli, G., Giuntini, R., & Paoli, F. (2011). Irreversibility in quantum computational logics. *Applied Mathematics and Information Sciences, 5*(2), 171–191.

Sergioli, G., Santucci, E., Didaci, L., Miszczak, J. A., & Giuntini, R. (2017). A quantum inspired version of the NMC classifier. *Soft Computing, 22*(3), 691–705.

Takahashi, Y., Kunihiro, N., & Ohta, K. (2007). The quantum Fourier transform on a linear nearest neighbor architecture. *Quantum Information and Computation, 7*, 383–391.

Toffoli, T. (1980). Reversible computing. In *Proceedings of the 7th colloquium on automata, languages and programming* (pp. 632–644). London: Springer.

**Giuseppe Sergioli** is a Theoretical Physicist (graduated at the University of Catania). He got his Ph.D. in Logic and Philosophy of Science at the University of Cagliari. He has been Visiting Scholar at the Indiana University-Bloomington and he is currently Researcher at the University of Cagliari and he is coordinator of the local project “Logical-time evolution of correlated microscopic systems”. His main research interests are: Quantum Information and Computation, Foundations of Quantum Mechanics and Quantum Computational Logic. He has published several papers in prominent journals in the fields on Computer Science, Theoretical Physics and Foundations of Sciences.