Hierarchical Verification of Quantum Circuits

Sidi Mohamed Beillahi(✉), Mohamed Yousri Mahmoud, and Sofiène Tahar

Department of Electrical and Computer Engineering,
Concordia University, Montreal, Canada
{beillahi,mo_solim,tahar}@ece.concordia.ca

Abstract. In this paper, we introduce the idea of hierarchical verification for quantum circuits, where we use a powerful language, higher-order logic, to reason about quantum circuits formally. We propose a formal modeling and verification approach that captures quantum models built hierarchically from primitive optical quantum gates. The analysis and verification of composed circuits is done seamlessly based on dedicated mathematical foundations formalized in the HOL Light theorem prover. In order to demonstrate the effectiveness of the proposed infrastructure, we present the formal analysis of the controlled-phase gate and Shor's factoring quantum circuits.

1 Introduction

Since it has been proved that classical machines cannot simulate quantum physics in polynomial times [11], scientists were working to develop new computers which employ quantum physics. Throughout their research, quantum technologies showed a good potential to provide solutions to several challenges such as secure communication and faster computation. Quantum optics is considered as one of the promising approaches for realizing "universal" quantum machines [6].

Despite the fact that quantum computers are not yet commercially available, their realisation requires the development of comprehensive tools for the modeling and verification of quantum devices. Due to the inherent complexity of quantum circuits, numerical simulations are incomplete: the computation space increases exponentially with the size of the circuit. Nevertheless, a number of tools have been proposed for simulation of quantum circuits. For instance, in [4] quantum gates are described as matrices and applied to quantum states using matrix-vector multiplication, however, a time-out is reached when simulating 15 qubits (quantum bits) circuits. Hence, we believe that there is a dire need of comprehensive and expressive computer-aided design and verification tools for quantum systems that cover both the mathematics and the principles of quantum physics.

Higher-order-logic (HOL) theorem proving is an effective approach to analyze engineering systems, thanks to its solid mathematics. Therefore, we believe that HOL can assist in the modeling and verification of quantum computers. In this paper, we propose to use the HOL Light theorem prover [5] to handle the hierarchical verification of quantum circuits thanks to its rich support for

© Springer International Publishing Switzerland 2016
S. Rayadurgam and O. Tkachuk (Eds.): NFM 2016, LNCS 9690, pp. 344–352, 2016.
DOI: 10.1007/978-3-319-40648-0_26

multivariate calculus and Hilbert spaces theories [9] which are essential to reason about quantum optics.

Our ultimate goal is to build the necessary tools to formally model and verify quantum circuits composed of primitive quantum gates, that are built using only optical components, in a hierarchical fashion. The first step towards this goal is to formally define in HOL the required mathematics, including the notions of projection, tensor product, and tensor product projection. We then apply these definitions to formally model and verify quantum primitive gates and circuits. We use this approach to formally model and verify the controlled-phase (CZ) gate circuit [6] and the Shor's factorization circuit of number 15 [1]. The source code of our formalization is available for download at [2].

In [8], the authors formalized the controlled-not (CNOT) gate. However, they did not provide the bi-linearity of tensor product and other important properties which are required to model and verify composed quantum circuits. In [12], a quantum process calculus is used to model linear optical quantum systems. It was applied to model the CNOT gate. The main limitation of this work is that the beam splitters parameters are considered as real numbers, however, they often need to be complex numbers as in the case of quantum interferometer [10].

2 Formalization of Tensor Product and Projection

For quantum optics the state of a quantum system is a probability density function which provides the probability of the number of photons inside the optical beam, typically written as $|\psi\rangle$. The set of quantum pure states (i.e., states which form the basis for a quantum states space) are called fock states. An optical beam in a fock state $|n\rangle$, where $n = 0, 1, 2, \ldots$, means that the light stream exactly contains n photons. Given an n-beam quantum state where each $|\psi\rangle_{\mathtt{k}}$, $k \in [1; n]$, describes the quantum state of single mode beam k, then the joint state of the n optical beams is $|\psi\rangle_{\mathtt{1}} \otimes |\psi\rangle_{\mathtt{2}} \otimes ... \otimes |\psi\rangle_{\mathtt{n}}$ (sometimes we use $|\psi_{\mathtt{1}}, \psi_{\mathtt{2}}, ..., \psi_{\mathtt{n}}\rangle$), where $\otimes$ operation is the tensor product.

2.1 Formalization of Tensor Product

Given the quantum state $|\psi\rangle_1 \ldots |\psi\rangle_n$ of n optical beams, the function that describes the joint probability of the n beams is then the point-wise multiplication of all the states, which refers to the optical states tensor product. Hence, we define the tensor product for an n-beam quantum state as follows: $\lambda\ y_1 \ldots y_n.\ (|\psi\rangle_1 \otimes \ldots \otimes |\psi\rangle_n)(y_1 \ldots y_n) = |\psi\rangle_1 y_1 * \ldots * |\psi\rangle_n y_n$. We therefore define the tensor product for n beams in HOL, recursively, as:

Definition 1 (Tensor Product)

$\vdash$ $\mathtt{tensor\ 0\ mode} = (\lambda \mathtt{y.\ 1})\ \wedge$
$\mathtt{tensor\ n+1\ mode} = (\lambda \mathtt{y.\ ((tensor\ n\ mode)\ y)} * (\mathtt{mode\$(n+1)\ y\$(n+1)}))$

where the symbol \$ denotes the vector indexing operator ($a\$i \Leftrightarrow a(i)$). mode is a vector of size n that contains n modes. The basic case of zero mode $\texttt{n} = 0$ is a trivial case; it is a constant function (i.e., $y \rightarrow 1$) and it guarantees a terminating definition. Next, we prove that this tensor satisfies the bi-linearity property:

Theorem 1 (Tensor: Bi-Linearity)

$\vdash\ \texttt{0} < \texttt{k} \leq \texttt{n} + \texttt{1} \wedge \texttt{mode\$k} = \texttt{a1}\ \%\ \texttt{x1} + \texttt{a2}\ \%\ \texttt{x2}\ \Rightarrow$
$\texttt{tensor n} + \texttt{1 mode} = \texttt{a1}\ \%\ \texttt{tensor n} + \texttt{1}\ (\lambda\texttt{i. if i} = \texttt{k then x1 else mode\$i})$
$\quad + \texttt{ a2}\ \%\ \texttt{tensor n} + \texttt{1}\ (\lambda\texttt{i. if i} = \texttt{k then x2 else mode\$i})$

where the symbol % denotes the scalar multiplication. Note that the number of modes is $n + 1$ as this property does not hold for 0 where tensor is the constant function. The two assumptions $\texttt{0} < \texttt{k} \leq \texttt{n} + \texttt{1}$ and $\texttt{mode\$k} = \texttt{a1}\ \%\ \texttt{x1} + \texttt{a2}\ \%\ \texttt{x2}$ ensure that the element k is part of the tensor and is a combination of two vectors. The proof is based on using induction where the base case is trivial and in the inductive step we use the lemma $\texttt{k} \leq \texttt{n} + \texttt{2} \Leftrightarrow (\texttt{k} \leq \texttt{n} + \texttt{1} \vee \texttt{k} = \texttt{n} + \texttt{2})$ then using the induction hypothesis for the first and the definition of tensor for the second.

An important property for the manipulation of the tensor product is when we have a tensor constructed out of two elementary tensors. In this case, this property states that a tensor $v_1 \otimes ... \otimes v_m \otimes u_1 \otimes ... \otimes u_n = (v_1 \otimes ... \otimes v_m) \otimes (u_1 \otimes ... \otimes u_n)$.

Theorem 2 (Tensor: Multiplication)

$\vdash\ \texttt{tensor m} + \texttt{n mode}\ =$
$\quad (\lambda\texttt{y.}\,((\texttt{tensor m mode})\,\texttt{y}) * (\texttt{tensor n}\,(\lambda\texttt{i. mode\$(i} + \texttt{m)}))\,(\lambda\texttt{i. y\$(i} + \texttt{m)}))$

A typical usage of this theorem is to separate elementary tensors for the sake of conducting quantum transformations independently from each other. Then using the same theorem, we can return back to the initial tensor.

2.2 Formalization of Linear Projection

In linear algebra, a projection is a linear transformation p from a vector space to itself that maintains the idempotent property; $p^2 = p$. In the quantum context, for a pure state $|\psi\rangle$, the projection is defined as $p = |\psi\rangle\langle\psi|$ which is a self-adjoint and linear transformation. In particular, for a quantum circuit design, the expected circuit output is the projection of all possible outputs over the appropriate fock states. For example, let us consider the state $|\phi\rangle = \frac{1}{3}|n\rangle + \frac{1}{3}|n-1\rangle + \frac{1}{3}|n+1\rangle$ and the projection $p_n = |n\rangle\langle n|$. The result of the projection of $|\phi\rangle$ is $p_n(|\phi\rangle) = |n\rangle\langle n|(\frac{1}{3}|n\rangle + \frac{1}{3}|n-1\rangle + \frac{1}{3}|n+1\rangle) = \frac{1}{3}|n\rangle$, because the fock states form an orthonormal basis. Therefore, we define the projection on fock states as follows:

Definition 2 (Linear Projection)

$$\vdash \forall\ \texttt{x.}\ (\texttt{proj}\ \ |\texttt{n}\rangle_{\texttt{sm}})\ \texttt{x} = \langle \texttt{n}_{\texttt{sm}}|\texttt{x}\rangle\ \ \%\ \ |\texttt{n}\rangle_{\texttt{sm}}$$

where $\texttt{proj}\ |\texttt{n}\rangle_{\texttt{sm}}$ is the projection over the fock state and accepts as parameter x. We have proven the three requirements for this projection which are linearity, idempotent and self-adjoint properties. Next we show the first two properties:

Theorem 3 (Projection: Linearity)

$$\begin{array}{l}\vdash \texttt{is\_sm sm} \Rightarrow \forall\ \texttt{x y a.}\\ \quad (\texttt{proj}\ \ |\texttt{n}\rangle_{\texttt{sm}})\ (\texttt{a1}\,\%\texttt{x} + \texttt{a2}\,\%\texttt{y}) = \texttt{a1}\,\%(\texttt{proj}\ \ |\texttt{n}\rangle_{\texttt{sm}})\ \texttt{x} + \texttt{a2}\,\%(\texttt{proj}\ \ |\texttt{n}\rangle_{\texttt{sm}})\ \texttt{y}\end{array}$$

where the assumption `is_sm sm` is used to maintain the requirement that the optical mode `sm` is indeed the single mode used.

Theorem 4 (Projection: Idempotent)

$$\vdash \texttt{is\_sm sm} \Rightarrow \forall \texttt{x.}\ (\texttt{proj}\ \ |\texttt{n}\rangle_{\texttt{sm}})\ ((\texttt{proj}\ \ |\texttt{n}\rangle_{\texttt{sm}})\ \texttt{x}) = (\texttt{proj}\ \ |\texttt{n}\rangle_{\texttt{sm}})\ \texttt{x}$$

2.3 Formalization of Tensor Product Projection

In some realization of quantum optics, the gates are implemented using *ancillas* which are extra qubits that are used for detecting the correct output [7]. During the design process of a quantum circuit, the ancilla is measured after it leaves the circuit. The correct output is known whenever the detector registers the expected ancilla. In our formalization, we implement the process of detecting the expected ancillas in the outputs of a quantum circuit as the tensor product projection of the outputs. We combine the tensor product and linear projection together to obtain the tensor product projection. By doing this, we will eliminate the undesirable outputs and keep only the "correct" output. In addition, we will have the projected state multiplied by a scalar value which is the success probability of the circuit. We define the projection of multi-mode states as follows:

Definition 3 (Tensor Projection)

$$\begin{array}{l}\vdash \texttt{is\_tensor\_proj m\_proj}\ \ \Leftrightarrow \forall\ \texttt{mode1 mode2 n.}\\ \quad \texttt{is\_linear\_cop}\ (\texttt{m\_proj}\ (\texttt{tensor n mode1}))\ \wedge\\ \quad \texttt{m\_proj}\,(\texttt{tensor n mode1})\,(\texttt{tensor n mode2}) =\\ \qquad\qquad \texttt{tensor n}\ (\lambda\texttt{i.}((\texttt{proj mode1\$i})\ \texttt{mode2\$i}))\end{array}$$

where `is_linear_cop op` ensures that the operator `op` is indeed a linear operator. Using this definition, we prove a crucial property in the analysis of quantum circuits, which states that $(p_1 \otimes ... \otimes p_n)(u_1 \otimes ... \otimes u_n) = p_1(u_1) \otimes ... \otimes p_n(u_n)$:

Theorem 5 (Tensor Projection: Multiplication)

$$\begin{array}{l}\vdash \texttt{is\_tensor\_proj m\_proj} \wedge 1 \leq \texttt{n} \Rightarrow\\ (\texttt{m\_proj tensor m} + \texttt{n mode1})\ \texttt{tensor m} + \texttt{n mode2} =\\ (\lambda\texttt{y.}\ ((\texttt{m\_proj tensor m mode1})\ \texttt{tensor m mode2})\ \texttt{y} *\\ (\texttt{m\_proj tensor n}\ (\lambda\texttt{i.mode1\$(i} + \texttt{m}))\ \texttt{tensor n}\ (\lambda\texttt{i.mode2\$(i} + \texttt{m})))\ (\lambda\texttt{i.y\$(i} + \texttt{m})))\end{array}$$

This property is very useful when projecting a multi-mode state which is applied to parallel quantum gates as the case for the controlled-phase gate. Using the tensor product lemma $v_1 \otimes ... \otimes 0 \otimes ... \otimes v_n = 0$, we prove the following property:

Theorem 6 (Tensor Projection: Fock States)

$\vdash \mathtt{is\_tensor\_proj\ m\_proj} \wedge 0 < \mathtt{k} \wedge \mathtt{mode1\$k} = |\mathtt{m1}\rangle_{\mathtt{sm}} \wedge \mathtt{mode2\$k} = |\mathtt{m2}\rangle_{\mathtt{sm}} \wedge$
$\mathtt{m1} \neq \mathtt{m2} \wedge \mathtt{is\_sm\ sm} \wedge \mathtt{k} \leq \mathtt{n} + 1 \Rightarrow$
$(\mathtt{m\_proj\ tensor\ n} + 1\ \mathtt{mode1})\ \mathtt{tensor\ n} + 1\ \mathtt{mode2} = 0$

This theorem is very important for the measurement of photons as it indicates that for two multi-mode states, where in the first state, the single mode k contains the fock state $|m1\rangle$ and in the second state, the single mode k contains the fock state $|m2\rangle$. If $m1$ and $m2$ are different, then the projection of the first multi-mode state over the other is zero. By this, we have covered the required mathematics for dealing with the verification and analysis of quantum circuits.

3 Hierarchical Verification: Applications

In this section, we will demonstrate the idea of hierarchical verification of quantum circuits based on the formalization of primitive gates, reported in [3], by formally verifying the controlled-phase (CZ) gate and Shor's factoring of number 15 circuits.

3.1 Verification of CZ Gate

A CZ gate is constructed using two non-linear sign (NS) gates [3] and two beam splitters, as shown in Fig. 1. The CZ gate transforms the input $|x, y\rangle$ to the output $e^{i\pi x.y}|x, y\rangle$, $x, y \in \{0, 1\}$. The success probability of measuring the ancilla state $|1, 0\rangle$ in both NS gates is $\frac{1}{16}$ [3]. We define the CZ gate as follows:

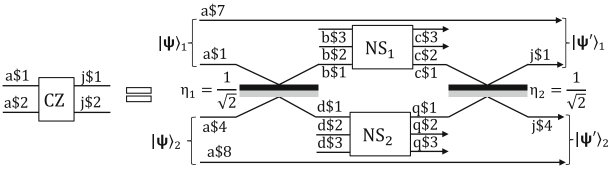

Fig. 1. Controlled-phase gate circuit

Definition 4 (CZ Gate)

$\vdash \mathtt{is\_cz}\ (\mathtt{a}, \mathtt{j}, \mathtt{ten})\ \Leftrightarrow (\forall\ \mathtt{b\ c\ d\ q\ k\ l\ m\ p}.$
$\mathtt{ns\_gate}(\mathtt{d}, \mathtt{m}, \mathtt{p}, \mathtt{q}, \mathtt{ten}) \wedge \mathtt{ns\_gate}(\mathtt{b}, \mathtt{l}, \mathtt{k}, \mathtt{c}, \mathtt{ten})\ \wedge$
$\mathtt{beam\_splitter}(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, \mathtt{ten}, \mathtt{a\$1}, 1, \mathtt{a\$4}, 4, \mathtt{b\$1}, 1, \mathtt{b\$4}, 4)\ \wedge$
$\mathtt{beam\_splitter}(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, \mathtt{ten}, \mathtt{c\$1}, 1, \mathtt{c\$4}, 4, \mathtt{j\$1}, 1, \mathtt{j\$4}, 4)\ \wedge$
$(\mathtt{q\$1} = \mathtt{c\$4} \wedge \mathtt{q\$2} = \mathtt{c\$5} \wedge \mathtt{q\$3} = \mathtt{c\$6}) \wedge (\mathtt{b\$4} = \mathtt{d\$1} \wedge \mathtt{b\$5} = \mathtt{d\$2} \wedge \mathtt{b\$6} = \mathtt{d\$3})$

Note that we rename the input and output ports for the second NS gate in order to match the order of the modes in the definition of NS, instead of $|b\$4, b\$5, b\$6\rangle$ and $|c\$4, c\$5, c\$6\rangle$ we have $|d\$1, d\$2, d\$3\rangle$ and $|q\$1, q\$2, q\$3\rangle$, respectively. We formally verified the CZ operations and its success probability for the four possible combinations of inputs, among which we provide here one of them.

Theorem 7 (CZ Gate: Input: $|1, 1\rangle$)

$\vdash$ $\texttt{let constraints} = \texttt{is\_tensor\_proj m\_proj} \wedge \texttt{is\_tensor ten} \wedge$
$\texttt{is\_cz (a, j, ten) in}$
$\texttt{let } |2,1,0,0,1,0,0,0\rangle_{\texttt{cq}} = \texttt{tensor}\,8\,(\lambda\texttt{i. if i} = 1\,\texttt{then}\,|2\rangle_{\texttt{c\$1}}\,\texttt{elseif i} = 2\,\texttt{then}$
$|1\rangle_{\texttt{c\$2}}\,\texttt{elseif i} = 5\,\texttt{then}\,|1\rangle_{\texttt{q\$2}}\,\texttt{else}\,|0\rangle_{\texttt{c\$3}})\texttt{ in}$
$\texttt{let } |0,1,0,2,1,0,0,0\rangle_{\texttt{cq}} = \texttt{tensor}\,8\,(\lambda\texttt{i. if i} = 2\,\texttt{then}\,|1\rangle_{\texttt{c\$2}}\,\texttt{elseif i} = 4\,\texttt{then}$
$|2\rangle_{\texttt{q\$1}}\,\texttt{elseif i} = 5\,\texttt{then}\,|1\rangle_{\texttt{q\$2}}\,\texttt{else}\,|0\rangle_{\texttt{c\$3}})\texttt{ in}$
$\texttt{let } |1,1,0,1,1,0,0,0\rangle_{\texttt{cq}} = \texttt{tensor}\,8\,(\lambda\texttt{i. if i} = 1\,\texttt{then}\,|1\rangle_{\texttt{c\$1}}\,\texttt{elseif i} = 2\,\texttt{then}$
$|1\rangle_{\texttt{c\$2}}\,\texttt{elseif i} = 4\,\texttt{then}\,|1\rangle_{\texttt{q\$1}}\,\texttt{elseif i} = 5\,\texttt{then}\,|1\rangle_{\texttt{q\$2}}\,\texttt{else}\,|0\rangle_{\texttt{c\$3}})\texttt{ in}$
$\texttt{let } |1,1,0,1,1,0,0,0\rangle_{\texttt{ab}} = \texttt{tensor}\,8\,(\lambda\texttt{i. if i} = 1\,\texttt{then}\,|1\rangle_{\texttt{a\$1}}\,\texttt{elseif i} = 2\,\texttt{then}$
$|1\rangle_{\texttt{b\$2}}\,\texttt{elseif i} = 4\,\texttt{then}\,|1\rangle_{\texttt{a\$4}}\,\texttt{elseif i} = 5\,\texttt{then}\,|1\rangle_{\texttt{b\$5}}\,\texttt{else}\,|0\rangle_{\texttt{b\$3}})\texttt{ in}$
$\texttt{let } |1,1,0,1,1,0,0,0\rangle_{\texttt{cj}} = \texttt{tensor}\,8\,(\lambda\texttt{i. if i} = 1\,\texttt{then}\,|1\rangle_{\texttt{j\$1}}\,\texttt{elseif i} = 2\,\texttt{then}$
$|1\rangle_{\texttt{c\$2}}\,\texttt{elseif i} = 4\,\texttt{then}\,|1\rangle_{\texttt{j\$4}}\,\texttt{elseif i} = 5\,\texttt{then}\,|1\rangle_{\texttt{c\$5}}\,\texttt{else}\,|0\rangle_{\texttt{c\$3}})\texttt{ in}$
$\texttt{constraints} \Rightarrow (\texttt{m\_proj}\ |2,1,0,1,0,0,0,0\rangle_{\texttt{cq}} + \texttt{m\_proj}\ |0,1,0,1,2,0,0,0\rangle_{\texttt{cq}} +$
$\texttt{m\_proj}\ |1,1,0,1,1,0,0,0\rangle_{\texttt{cq}})\ (|1,1,0,1,1,0,0,0\rangle_{\texttt{ab}}) = -\frac{1}{4}\ \%\ |1,1,0,1,1,0,0,0\rangle_{\texttt{cj}}$

Note that the output of the CZ gate has been projected over three different states. This is because that we have two photons at the input ($|1, 1\rangle$) which results in three possibilities at the input of the two parallel NS gates: (1) two photons go through the first NS gate; (2) two photons go through the second NS gate; and (3) one photon goes through the first NS gate and the other goes through the second NS gate. The verification of the CZ gate has been done using Theorem 6 in order to subdivide the main tensor product projection to two tensor product projections, where each is fed to an NS gate. This completes the analysis of the CZ for the input "11". The analysis for the inputs "01", "00", and "10" follows the same pattern. The actual physical implementations of the CZ gate have 8 input modes. However, the CZ is a 2-qubits gate, where each logical qubit is represented by two optical modes and the rest of the modes are ancillas. Therefore in order to facilitate the use of this gate in complex quantum circuits, we developed an input/output behavioral description:

Definition 5 (CZ Behavioral Description)

$\texttt{Input}: |1,1\rangle_{\texttt{L}} \equiv (\texttt{m\_proj}\ |2,1,0,1,0,0,0,0\rangle_{\texttt{cq}} + \texttt{m\_proj}\ |0,1,0,1,2,0,0,0\rangle_{\texttt{cq}} +$
$\texttt{m\_proj}\ |1,1,0,1,1,0,0,0\rangle_{\texttt{cq}})\ |1,1,0,1,1,0,0,0\rangle_{\texttt{ab}}$
$\texttt{Output}: |1,1,0,1,1,0,0,0\rangle_{\texttt{cj}} \equiv |1,1\rangle_{\texttt{L}}$

3.2 Verification of Shor's Factorization

Shor's integer factorization is a quantum algorithm to compute the two primes factor of a given integer much faster than classical algorithms. Our objective here is to show the formal modeling and verification of a compiled version of Shor's factoring of number 15 [1] using the previously presented formalization. The task of the underlying circuit is to find the minimum integer r that satisfies a^r *mode* $N = 1$, where $N = 15$ and a is a randomly chosen co-prime integer to N, in our case $a = 2$. r is called the order of a modulo N, from which we compute the desired prime factors; $(a^{\frac{r}{2}} - 1)$ and $(a^{\frac{r}{2}} + 1)$. The circuit is composed of six Hadamard [3] and two CZ gates, as shown in Fig. 2, and has 4 inputs/outputs. Inputs are initialized to the state; $|\psi\rangle_{in} = |0,0,1,0\rangle_{x1f1f2x2}$. From the computed output, $|\psi\rangle_{out} = |.,.,.,.\rangle_{\dot{x}1\dot{f}1\dot{f}2\dot{x}2}$, we extract the variable $z = |.,.,0\rangle_{\dot{x}1\dot{x}2}$, then we obtain $r = a^z$ *mod* 15. Accordingly, we formally define the structure of the circuit and verify its operation as follows:

Definition 6 (Shor Circuit)
$\vdash \texttt{shor}\ (\texttt{x1},\texttt{x2},\texttt{f1},\texttt{f2},\ddot{\texttt{f}}\texttt{1},\ddot{\texttt{f}}\texttt{2},\texttt{j1},\texttt{j2},\texttt{ten}) \Leftrightarrow (\forall\ \texttt{b}\ \texttt{d}.\ \texttt{j2\$2} = \ddot{\texttt{x}}\texttt{2}\ \wedge$
$\texttt{is\_hadamard}(\texttt{x1},\texttt{b\$2},\texttt{ten}) \wedge \texttt{is\_hadamard}(\texttt{f1},\texttt{b\$1},\texttt{ten}) \wedge \texttt{is\_cz}(\texttt{d},\texttt{j2},\texttt{ten})\ \wedge$
$\texttt{is\_hadamard}(\texttt{x2},\texttt{d\$2},\texttt{ten}) \wedge \texttt{is\_hadamard}(\texttt{f2},\texttt{d\$2},\texttt{ten}) \wedge \texttt{is\_cz}(\texttt{b},\texttt{j1},\texttt{ten})\ \wedge$
$\texttt{j1\$2} = \ddot{\texttt{x}}\texttt{1} \wedge \texttt{is\_hadamard}(\texttt{j1\$1},\ddot{\texttt{f}}\texttt{1},\texttt{ten}) \wedge \texttt{is\_hadamard}(\texttt{j2\$1},\ddot{\texttt{f}}\texttt{2},\texttt{ten}))$

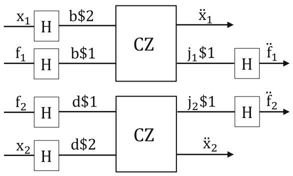

Fig. 2. Shor's factoring of 15 circuit

Theorem 8 (Shor' Factoring of 15)
$\vdash$ $\texttt{let constraints} = \texttt{is\_tensor\_proj m\_proj} \wedge \texttt{is\_tensor ten}\ \wedge$
$\texttt{shor}\ (\texttt{x1},\texttt{x2},\texttt{f1},\texttt{f2},\ddot{\texttt{f}}\texttt{1},\ddot{\texttt{f}}\texttt{2},\texttt{j1},\texttt{j2},\texttt{ten})\ \texttt{in}$
$\texttt{let}\ |0,0,1,0\rangle_{\texttt{f1x1f2x2}} = \texttt{tensor}\,4\,(\lambda\texttt{i}.\ \texttt{if i} = 1\ \texttt{then}\ |0\rangle_{\texttt{f1}}\ \texttt{elseif}\ \ \texttt{i} = 2\ \texttt{then}\ |0\rangle_{\texttt{x1}}$
$\texttt{elseif i} = 3\ \texttt{then}\ |1\rangle_{\texttt{f2}}\ \texttt{else}\ |0\rangle_{\texttt{x2}})\ \ \texttt{in}$
$\texttt{let}\ |0,0,0,1\rangle_{\ddot{\texttt{f}}1\ddot{\texttt{x}}1\ddot{\texttt{f}}2\ddot{\texttt{x}}2} = \texttt{tensor}\,4\,(\lambda\texttt{i}.\ \texttt{if i} = 1\ \texttt{then}\ |0\rangle_{\ddot{\texttt{f}}1}\ \texttt{elseif i} = 2\ \texttt{then}\ |0\rangle_{\ddot{\texttt{x}}1}$
$\texttt{elseif i} = 3\ \texttt{then}\ |0\rangle_{\ddot{\texttt{f}}2}\ \texttt{else}\ |1\rangle_{\ddot{\texttt{x}}2})\ \ \texttt{in}$
$\texttt{let}\ |0,0,1,0\rangle_{\ddot{\texttt{f}}1\ddot{\texttt{x}}1\ddot{\texttt{f}}2\ddot{\texttt{x}}2} = \texttt{tensor}\,4\,(\lambda\texttt{i}.\ \texttt{if i} = 1\ \texttt{then}\ |0\rangle_{\ddot{\texttt{f}}1}\ \texttt{elseif i} = 2\ \texttt{then}\ |0\rangle_{\ddot{\texttt{x}}1}$
$\texttt{elseif i} = 3\ \texttt{then}\ |1\rangle_{\ddot{\texttt{f}}2}\ \texttt{else}\ |0\rangle_{\ddot{\texttt{x}}2})\ \ \texttt{in}$
$\texttt{let}\ |1,1,0,1\rangle_{\ddot{\texttt{f}}1\ddot{\texttt{x}}1\ddot{\texttt{f}}2\ddot{\texttt{x}}2} = \texttt{tensor}\,4\,(\lambda\texttt{i}.\ \texttt{if i} = 1\ \texttt{then}\ |1\rangle_{\ddot{\texttt{f}}1}\ \texttt{elseif i} = 2\ \texttt{then}\ |1\rangle_{\ddot{\texttt{x}}1}$
$\texttt{elseif i} = 3\ \texttt{then}|0\rangle_{\ddot{\texttt{f}}2}\ \texttt{else}\ |1\rangle_{\ddot{\texttt{x}}2})\ \texttt{in}$
$\texttt{let}\ |1,1,1,0\rangle_{\ddot{\texttt{f}}1\ddot{\texttt{x}}1\ddot{\texttt{f}}2\ddot{\texttt{x}}2} = \texttt{tensor}\,4\,(\lambda\texttt{i}.\ \texttt{if i} = 1\ \texttt{then}\ |1\rangle_{\ddot{\texttt{f}}1}\ \texttt{elseif i} = 2\ \texttt{then}\ |1\rangle_{\ddot{\texttt{x}}1}$
$\texttt{elseif i} = 3\ \texttt{then}\ |1\rangle_{\ddot{\texttt{f}}2}\ \texttt{else}\ |0\rangle_{\ddot{\texttt{x}}2})\ \ \texttt{in}$
$\texttt{constraints} \Rightarrow |0,0,1,0\rangle_{\texttt{f1x1f2x2}} = \frac{1}{32}\ \ \%\ \ (|1,1,1,0\rangle_{\ddot{\texttt{f}}1\ddot{\texttt{x}}1\ddot{\texttt{f}}2\ddot{\texttt{x}}2} + |1,1,0,1\rangle_{\ddot{\texttt{f}}1\ddot{\texttt{x}}1\ddot{\texttt{f}}2\ddot{\texttt{x}}2}$
$+\ \ |0,0,1,0\rangle_{\ddot{\texttt{f}}1\ddot{\texttt{x}}1\ddot{\texttt{f}}2\ddot{\texttt{x}}2} + |0,0,0,1\rangle_{\ddot{\texttt{f}}1\ddot{\texttt{x}}1\ddot{\texttt{f}}2\ddot{\texttt{x}}2})$

Here the circuit outputs two categories of solutions; (1) $|000\rangle$ or $|100\rangle$ which are expected failures of the algorithm [1]; (2) $|010\rangle$ or $|110\rangle \equiv z = 2$ or $z = 6$ which give $r = 4$ from which we obtain the 5 and 3 prime numbers. The verification of the compiled Shor's circuit has been done using Theorem 3 to subdivide the tensor to four tensors, and apply Hadamard transformation on each elementary tensor.

4 Conclusion and Discussion

In this paper, we reported a novel application of formal methods to enable the hierarchical modeling and verification of quantum circuits. We presented the higher-order logic formalization of mathematical foundations such as tensor product, linear projection, and tensor product projection. Then we showed how they can be applied for the hierarchical modeling and analysis of composed quantum circuits using the CZ gate and Shor's 15 factoring circuits.

One of the important outcomes of this work is the efficiency that the tensor projection brought to our formalization: if we tackled the NS gate without the projection (such as in [8,12]), we will have 10 possible outputs (with only one correct output) which dramatically affects the CZ analysis that contains two parallel NS gates which in turn produce $10*10 = 100$ possible outputs. Moreover, it gets worse when it comes to the Shor's circuit where we have two CZ gates and at the level of inputs of each gate we have four possible inputs, which means at the output of these gates we have $(4 * 100) * (4 * 100) = 16.10^4$ possible outputs. Thanks to the projection properties, such as projection linearity and projection of two orthogonal tensor products, we were able to reduce the possible outputs to consider only the correct ones. This is very important for scalability reasons, especially for larger circuits which contain many quantum gates. We believe this to be a significant feature of our formalization compared to before mentioned related works, e.g., [8,12]. The reported mathematical foundation can be used to reduce the complexity in the implementation of design verification tools for quantum optics circuits analysis.

In future work, we plan to apply the formalization developed in this paper to perform a formal synthesis of quantum circuits and to apply our methods on other quantum systems, such as Grover's algorithm.

References

1. Politi, A., Matthews, J.C.F., O'Brien, J.L.: Shor's quantum factoring algorithm on a photonic chip. Science **325**(5945), 1221 (2009)
2. Beillahi, S.M., Mahmoud, M.Y.: Hierarchical Verification of Quantum Circuits (2016). http://hvg.ece.concordia.ca/projects/optics/hvqc.html
3. Beillahi, S.M., Mahmoud, M.Y., Tahar, S.: Optical Quantum Gates Formalization in HOL Light. Technical report, ECE Department, Concordia University, Montreal, QC, Canada, February 2016
4. Viamontes, G.F., Rajagopalan, M., Markov, I.L., Hayes, J.P.: Gate level simulation of quantum circuits. In: ASP-DAC, pp. 295–301 (2003)
5. Harrison, J.: HOL light: a tutorial introduction. In: Srivas, M., Camilleri, A. (eds.) FMCAD 1996. LNCS, vol. 1166, pp. 265–269. Springer, Heidelberg (1996)

6. Knill, E., Laflamme, R., Milburn, G.J.: A scheme for efficient quantum computation with linear optics. Nature **409**, 46–52 (2001)
7. Kok, P., Munro, W.J., Nemoto, K., Ralph, T.C., Dowling, J.P., Milburn, G.J.: Linear optical quantum computing with photonic qubits. Rev. Mod. Phys. **79**, 135–174 (2007)
8. Mahmoud, M.Y., Panangaden, P., Tahar, S.: On the formal verification of optical quantum gates in HOL. In: Núñez, M., Güdemann, M. (eds.) FMICS 2015. LNCS, vol. 9128, pp. 198–211. Springer, Heidelberg (2015)
9. Mahmoud, M.Y., Aravantinos, V., Tahar, S.: Formalization of infinite dimension linear spaces with application to quantum theory. In: Brat, G., Rungta, N., Venet, A. (eds.) NFM 2013. LNCS, vol. 7871, pp. 413–427. Springer, Heidelberg (2013)
10. Mandel, L., Wolf, E.: Optical Coherence and Quantum Optics. Cambridge University Press, Cambridge, UK (1995)
11. Feynman, R.P.: Simulating physics with computers. Int. J. Theor. Phys. **21**(6–7), 467–488 (1982)
12. Franke-Arnold, S., Gay, S.J., Puthoor, I.V.: Quantum process calculus for linear optical quantum computing. In: Dueck, G.W., Miller, D.M. (eds.) RC 2013. LNCS, vol. 7948, pp. 234–246. Springer, Heidelberg (2013)